



InfoMover File Transfer
Version 3.7

USER'S GUIDE
P/N 300-999-126
REV C08

EMC Corporation
Corporate Headquarters:
Hopkinton, MA 01748-9103
1-508-435-1000
www.emc.com

Copyright © 2000 - 2004 EMC Corporation. All rights reserved.

Printed July, 2004

EMC believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

THE INFORMATION IN THIS PUBLICATION IS PROVIDED "AS IS." EMC CORPORATION MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WITH RESPECT TO THE INFORMATION IN THIS PUBLICATION, AND SPECIFICALLY DISCLAIMS IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Use, copying, and distribution of any EMC software described in this publication requires an applicable software license.

Trademark Information

EMC², EMC, Symmetrix, Celerra, CLARiiON, CLARAlert, Documentum, HighRoad, LEGATO, Navisphere, PowerPath, ResourcePak, SnapView/IP, SRDF, TimeFinder, VisualSAN, and where information lives are registered trademarks and EMC Automated Networked Storage, EMC ControlCenter, EMC Developers Program, EMC OnCourse, EMC Proven, EMC Snap, Access Logix, AutoAdvice, Automated Resource Manager, AutoSwap, AVALONidm, C-Clip, Celerra Replicator, Centera, CentraStar, CLARevent, Connectrix, CopyCross, CopyPoint, DatabaseXtender, Direct Matrix, Direct Matrix Architecture, EDM, E-Lab, Engenuity, FarPoint, FLARE, GeoSpan, InfoMover, MirrorView, NetWin, OnAlert, OpenScale, Powerlink, PowerVolume, RepliCare, SafeLine, SAN Architect, SAN Copy, SAN Manager, SDMS, SnapSure, SnapView, StorageScope, SupportMate, SymmAPI, SymmEnabler, Symmetrix DMX, Universal Data Tone, and VisualSRM are trademarks of EMC Corporation. All other trademarks used herein are the property of their respective owners.

Preface	xiii
----------------------	------

Chapter 1 Introduction

Overview of InfoMover File Transfer	1-2
Upgrading the Software	1-2
Features	1-2
Architecture	1-3
IFT Client.....	1-3
IFT Server	1-3
Transfer Devices.....	1-3
User Interfaces.....	1-6
The Graphic User Interface	1-6
The Command Line Interface	1-6
The IFT Management Utility	1-6
File Transfers.....	1-7
OS/390 and z/OS Flat Files	1-7
Transfer Devices.....	1-8
Concurrent Connections	1-9
Data Translation	1-10
EMC TimeFinder/Mirror Software	1-10
Security.....	1-11
More Information.....	1-11

Chapter 2 Using the Graphic User Interface

Overview of the IFT Graphic User Interface.....	2-2
Invoking the GUI	2-3
Using the Workspace Tab.....	2-4
Understanding Sessions and Transfers.....	2-5

	Controlling Sessions	2-6
	Transferring Files	2-13
	Mainframe Datasets	2-19
	Using the Transfer Table Tab	2-21
	Using the Generate Scripts Tab	2-23
	Exiting the IFT GUI	2-26
Chapter 3	Using the Command Line Interface	
	Overview	3-2
	Supported Hosts	3-2
	CLI Syntax	3-2
	Usage	3-2
	CLI Interactive Mode	3-3
	Starting a Client	3-4
	Using the Verbose Mode	3-10
	Establishing a Connection	3-10
	Checking and Setting Conditions	3-13
	Transferring Data	3-14
	Closing Connections	3-17
	Stopping a Client	3-17
	CLI Batch or File Input Mode	3-19
	Creating a Script Manually	3-19
	Generating a Script with the GUI	3-22
	Generating a Script with the IFT Management Utility	3-22
	Encrypting Script File Passwords	3-23
	Running a Script	3-25
	Command Summary	3-27
Chapter 4	CLI Command Reference	
	Introduction	4-3
	binary	4-4
	blksize (OS/390 and z/OS)	4-5
	cd	4-7
	close	4-11
	del	4-12
	dir	4-14
	get	4-23
	help [?]	4-30
	imaddlf (OS/390 and z/OS)	4-32
	imbufno (OS/390 and z/OS)	4-35
	imcrLf	4-37

imcylinder (OS/390 and z/OS).....	4-40
imdirblocks (OS/390 and z/OS).....	4-44
imfilenumber (OS/390 and z/OS).....	4-46
imlog (Open Systems)	4-48
imrdw (OS/390 and z/OS).....	4-50
imrecvx	4-52
imrelease (OS/390 and z/OS)	4-54
imsendx.....	4-55
imtrail (OS/390 and z/OS).....	4-57
imwrap (Open Systems).....	4-59
imxlate.....	4-61
lcd	4-64
ldir	4-68
likedsn (OS/390 and z/OS).....	4-75
lpwd	4-78
lrecl (OS/390 and z/OS)	4-79
lsp (OS/390 and z/OS).....	4-80
mget.....	4-81
mput.....	4-84
open.....	4-87
prompt	4-89
put.....	4-90
pwd	4-97
quit.....	4-98
recfm (OS/390 and z/OS).....	4-99
replace (OS/390 and z/OS)	4-101
status	4-103
unit (OS/390 and z/OS).....	4-107
unitcount (OS/390 and z/OS).....	4-108
upper	4-110
verbose	4-112
volcount (OS/390 and z/OS)	4-113
volume (OS/390 and z/OS)	4-115

Appendix A Error and Event Messages

Overview of Event and Error Messages	A-2
Event Message Format	A-2
Looking Up an Error Code	A-3
Verbose Messages.....	A-3
GUI-Based Messages	A-5
Event Files on AIX, HP-UX, Solaris, and Windows Systems ..	A-6
AIX, HP-UX, and Solaris Events.....	A-6

Windows Events and the Windows Event Viewer	A-6
IFT Event Message Summary	A-7
Event Messages in Cancelled Mainframe Jobs	A-27
CANCEL Command and IFT Servers.....	A-27
Abnormal Disposition OUTDISP Subparameter	A-27

Appendix B Using IFT on Mainframe Hosts

Creating Batch CLI Client Jobs	B-2
Overview	B-2
Serializing Datasets	B-2
Continuation in SYSIN Control Statements	B-4
Using Generation Data Groups	B-7
Overview	B-7
GDG Names	B-7
GDG Support	B-9
Specifying Allocation Attributes	B-13
Overview	B-13
Default Values	B-13
Configuration Parameters	B-13
The likedsn Command	B-14
Other CLI Commands.....	B-16
Physical Sequential and Partitioned Data Sets.....	B-17
SMS Parameters: GDG Datasets.....	B-19
Magnetic Tape Support	B-24
Overview	B-24
Target Datasets on Tape.....	B-24
GDGs and Magnetic Tape	B-24
Configuration Parameters for Tape Operations.....	B-24
Tape Mount Operations.....	B-25
Dataset Allocations.....	B-25
Configuration Parameters and CLI Commands	B-26
Managing Files with ISO Control Characters	B-27
Fixed Format Files	B-27
Variable Format Files	B-27
Open Systems Files.....	B-27
Miscellaneous Issues	B-28
Performing Translation on the Mainframe	B-29

Appendix C Customer Support

Overview of Detecting and Resolving Problems	C-2
Troubleshooting the Problem	C-3
Before Calling the Customer Support Center	C-4

Documenting the Problem	C-4
Reporting a New Problem	C-5
Sending Problem Documentation	C-6
Glossary	g-1
Index	i-1

1-1	Host to Host Transfer Paths	1-4
2-1	The IFT GUI	2-3
2-2	Workspace Tab	2-4
2-3	Session Configuration Window	2-7
2-4	Session Pull Down Menus	2-9
2-5	Mainframe Session Detail	2-10
2-6	Workspace Panel with Enabled Translation	2-16
2-7	Transfer Status Settings Screen	2-17
2-8	Transfer Table Tab	2-21
2-9	The Generate Scripts Panel	2-23
2-10	Save Dialog Box	2-24
3-1	Performing Data Operations	3-3
A-1	Message Format	A-2
A-2	InfoMover File Transfer Error Window	A-5
C-1	Problem Detection and Resolution Process	C-2

1-1	InfoMover File Transfer File Types	1-7
2-1	Session Details	2-7
2-2	Extra Server Detail Configuration Details	2-8
2-3	GUI Status Settings	2-18
3-1	Valid Client Locations	3-11
3-2	Client Command Relationships	3-27
4-1	Mainframe dir Wildcard Characters	4-16
4-2	Open Systems dir Wildcard Characters	4-18
4-3	Mainframe ldir Wildcard Characters	4-70
4-4	Open Systems ldir Wildcard Characters	4-71
4-5	Valid mget Wildcard Characters	4-82
4-6	Valid mput Wildcard Characters	4-85
A-1	Message Severity Codes and Severity Levels	A-3
A-2	InfoMover File Transfer Event Messages	A-7
B-1	Disposition Statuses for put and get Commands	B-3
B-2	Absolute and Relative Dataset Names	B-9
B-3	Configuration Parameters for New Dataset Allocation	B-14
B-4	New PS and PDS Allocation Rules	B-17
B-5	New GDG Datasets Allocation Rules	B-19
B-6	Magnetic Tape Commands and Configuration Parameters	B-26

As part of an effort to improve and enhance the performance and capabilities of its product line, EMC from time to time releases revisions of its hardware and software. Therefore, some features described in this guide may not be supported by all revisions of the software or hardware currently in use. For the most up-to-date information about product features, refer to your product release notes.

If a product does not function properly or does not function as described in this guide, please contact your EMC representative.

Audience

This guide is intended for readers who will use the InfoMover File Transfer (IFT) component of InfoMover to transfer data. The guide expects that readers are familiar with:

- ◆ The host operating systems used at their site
- ◆ Symmetrix storage subsystems
- ◆ TCP/IP and network environments

Organization Here is a table listing where information is located in this guide:

Chapter or Appendix	Contents
Chapter 1, <i>Introduction</i>	An overview of IFT's components and how IFT operates.
Chapter 2, <i>Using the Graphic User Interface</i>	A discussion of how to use the IFT client GUI interface.
Chapter 3, <i>Using the Command Line Interface</i>	A discussion of how to use the IFT client CLI interface.
Chapter 4, <i>CLI Command Reference</i>	Reference pages about each CLI client command.
Appendix A, <i>Error and Event Messages</i>	A list of error codes and messages.
Appendix B, <i>Using IFT on Mainframe Hosts</i>	A discussion of various topics of interest to mainframe users.
Appendix C, <i>Customer Support</i>	Essential questions that you should be prepared to answer when contacting the EMC Customer Support Center.
<i>Glossary</i>	Definitions of terms used in this publication.

Related Documentation

The following EMC publications provide additional information:

Publication	EMC Part Number
<i>Getting Started with EMC InfoMover</i>	300-999-123
<i>InfoMover File Transfer Installation Guide</i>	300-000-451
<i>InfoMover File Transfer System Administrator's Guide</i>	300-999-125
<i>InfoMover File Transfer CLI Commands Quick Reference</i>	300-000-112

Related Vendor Information

For reference material about the operating systems supported by InfoMover File Transfer, refer to the following sites:

For information about	Go to
HP-UX (and Tru64 UNIX)	http://www.docs.hp.com/
NCR	http://www.ncr.com/
OS/390 and z/OS, AIX, or DYNIX/ptx	http://publibz.boulder.ibm.com/
Red Hat Linux	http://www.redhat.com/
Sun Solaris	http://docs.sun.com/
Tru64 UNIX	http://www.tru64unix.compaq.com/docs
Windows	http://www.microsoft.com/

Note: These sites were correct at the time of publication. However, because such information can change quickly, you should contact your system vendor to determine the correct URL.

Conventions Used in this Guide

EMC uses the following conventions for note and caution notices.

Text marked **Note:**, **Important:**, **Tip:**, **Background:**, or **Exception:** calls attention to any item of information that may be of special importance.



CAUTION

A caution contains information essential to avoid damage to the system or equipment. The caution may apply to hardware or software.

Typographical Conventions

EMC uses the following type style conventions in this guide:

Typeface	Uses
AVANTGARDE SMALLCAPS	Keystrokes
Sans Serif Bold	Selections you can make from windows, including menu items and field entries
Bold	Names of CLI commands or utilities used in text
<i>Italic</i>	<ul style="list-style-type: none"> • New terms or unique word usage in text • Emphasis in text • Variable values that are displayed only and do not have to be entered • Book titles in text • Heading titles used as references in text
<Courier>	Variables in syntax examples
Courier	<ul style="list-style-type: none"> • System output • Filenames • Complete paths
Courier bold	<ul style="list-style-type: none"> • User entry in examples • CLI commands in syntax examples • Options in\ syntax examples
<i>Courier italic</i>	<ul style="list-style-type: none"> • Arguments used in examples of command line syntax • Variables in examples of screen or file output • Variables in pathnames

The syntax conventions used in this guide are:

- ◆ [] = Encloses an optional entry in syntax statements
- ◆ | = Separates alternative argument values in statements

Terminology

This document uses “IBM mainframe host” or “mainframe host” to refer to an IBM mainframe host, and uses “OS/390 and z/OS” to refer to the IBM family of operating systems.

Where to Get Help

For questions about technical support call your local sales office or service provider.

If you have a valid EMC service contract, contact EMC Customer Service at:

United States: (800) 782-4362 (SVC-4EMC)

Canada: (800) 543-4782 (543-4SVC)

Worldwide: (508) 497-7901

Follow the voice menu prompts to open a service call and select the applicable product support.

Sales and Customer Service Contacts

For the list of EMC sales locations, please access the EMC home page at:

<http://emc.com/contact/>

For additional information on the EMC products and services available to customers and partners, refer to the EMC Powerlink Web site at:

<http://powerlink.emc.com>

Your Comments

Your suggestions will help us continue to improve the accuracy, organization, and overall quality of the user publications. Please send a message to techpub_comments@emc.com with your opinions of this guide.

Your technical enhancement suggestions for future development consideration are welcome. To send a suggestion, log on to <http://powerlink.emc.com>, follow the path **Support, Contact Support**, and choose **Software Product Enhancement Request** from the Subject menu.

This chapter introduces InfoMover File Transfer (IFT). Topics include:

- ◆ Overview of InfoMover File Transfer 1-2
- ◆ Architecture 1-3
- ◆ User Interfaces 1-6
- ◆ File Transfers 1-7
- ◆ Data Translation 1-10
- ◆ EMC TimeFinder/Mirror Software 1-10
- ◆ Security 1-11
- ◆ More Information 1-11

Note: This document uses “mainframe host” to refer to an IBM mainframe host, and uses “OS/390 and z/OS” to refer to the IBM family of operating systems.

Overview of InfoMover File Transfer

InfoMover™ File Transfer (IFT) 3.7 is the component of EMC InfoMover 4.4 that allows you to transfer data among:

- ◆ OS/390 and z/OS systems
- ◆ UNIX systems
- ◆ Windows systems

Upgrading the Software

If you are moving from IFT 3.6 to IFT 3.7, you can use IFT 3.7 without having to modify your existing scripts and batch jobs.

Features

IFT allows you to:

- ◆ Access data in OS/390 and z/OS datasets, UNIX files, UNIX named pipes, and Windows files through Symmetrix® storage subsystems
- ◆ Read or copy data from a source file on one host to a target file on another host
- ◆ Translate data from one format to another with single-byte translation (for example, ASCII ¹ to EBCDIC ² and EBCDIC to ASCII)
- ◆ Operate under the control of OS/390, z/OS, UNIX, and Windows security systems
- ◆ Use an FTP-like command line interface (CLI) or a graphic user interface (GUI)
- ◆ Save network resources and achieve higher performance by using Symmetrix channels rather than the network to read or copy data from one host to another

-
1. American Standard Code for Information Interchange.
 2. Extended Binary Coded Decimal Interchange Code.

Architecture

IFT architecture is a variation of client-server model. In fact, IFT architecture can be called *client-server-server* because IFT must have at least one client and two servers in the file transfer environment.

IFT Client

The user makes the file transfer request through the IFT client. The IFT client specifies the file or dataset to be accessed, the *source file*, and the location to which the data is to be copied, the *target file*. On UNIX systems, the source file and the target file can be actual files or named pipes.

The IFT client can reside on:

- ◆ An OS/390 or z/OS system that also has a resident IFT server
- ◆ A UNIX system with either a resident IFT server or TCP/IP (transmission control protocol/internet protocol) access to an IFT server on another UNIX or Windows system
- ◆ A Windows system with either a resident IFT server or TCP/IP access to an IFT server on another UNIX or Windows system

IFT Server

An IFT server must reside on every host from which and to which files are to be transferred. The IFT server processes the file transfer request from the IFT client and performs the actual file transfer.

The components of the IFT server are:

- ◆ The *Initiator* that runs on the local host and responds to IFT client requests
- ◆ The *Connector* that runs on the remote host and responds to initiator requests

Transfer Devices

IFT transmits data from one host to another through transfer devices on a common Symmetrix storage subsystem. The transfer devices must be fixed-block architecture (FBA) devices.

Your technical support or system administrative personnel, in conjunction with EMC Customer Support personnel, have configured these transfer devices to be accessible by all hosts that are to participate in data transfer.

The transfer devices provide disk space for buffering and transporting the data during a file transfer. This disk space can be:

- ◆ Spread across multiple logical volumes (as perceived by the host)
- ◆ Located on a single volume

You should not have more than eight clients and three active hosts sharing the same set of transfer devices concurrently.

More Information: Refer to the *InfoMover File Transfer System Administrator's Guide* for more information about transfer devices.

Transfer Characteristics

Figure 1-1 shows the data transfer paths between a Symmetrix storage subsystem and several hosts:

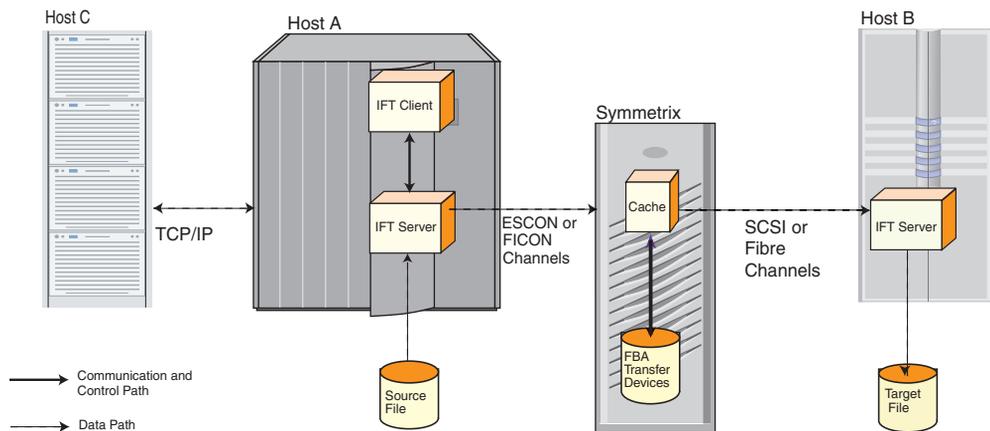


Figure 1-1 Host to Host Transfer Paths

In Figure 1-1, host A, host B, and host C can be:

- ◆ IBM mainframe hosts
- ◆ UNIX hosts
- ◆ Windows hosts

A TCP/IP network connection between hosts can be required, depending upon where the client is running. The IFT graphic user interface requires a TCP/IP connection. Source and target files reside on host disks. On IBM mainframe hosts, source and target datasets can also reside on magnetic tape devices.

The mainframe host and the common Symmetrix system can use any of the following types of Symmetrix channel connections:

- ◆ ESCON
- ◆ FICON

The UNIX and Windows hosts and the common Symmetrix system can use any of the following:

- ◆ Fast Wide SCSI
- ◆ Ultra SCSI
- ◆ Fibre Channel

Transfer Initiation

The following steps take place in the file transfer, as shown in Figure 1-1 on page 1-4:

1. The user verifies that the IFT servers are started on hosts A and B.
2. The user starts the IFT client on host A.
3. The IFT client on host A tells the IFT server on host A to connect to the IFT server on host B.
4. The user enters IFT commands through the IFT client to initiate data transfer. IFT sends these commands to host A and then to the Symmetrix storage subsystem.
5. The IFT servers on host B and host A accept and process the requests from the client.

User Interfaces

IFT has both a graphic user interface and a command line interface.

The Graphic User Interface

The GUI allows you to transfer files and generate batch scripts. You can use the GUI on any of the following systems:

- ◆ Windows 2000
- ◆ Windows NT
- ◆ Windows Server 2003

More Information: Refer to Chapter 2, *Using the Graphic User Interface*.

The Command Line Interface

The CLI uses FTP-like commands. These commands allow you to display lists of files and manage file transfers. The CLI is available on all supported platforms.

More Information: Refer to Chapter 3, *Using the Command Line Interface*.

The IFT Management Utility

The IFT Management Utility allows you to:

- ◆ Initialize the Symmetrix devices for transfer
- ◆ Create or modify script files with encrypted passwords
- ◆ Create or modify IFT configuration files
- ◆ Create or update your Symmetrix device database
- ◆ Create translation files for use by IFT
- ◆ View diagnostic information
- ◆ Gather (“bundle”) log files

More Information: Refer to the *InfoMover File Transfer System Administrator’s Guide* for more information about the Management Utility.

File Transfers

IFT supports the transfer of a dataset or a file that is structured as a flat file. Flat files consist of records of one type with no embedded structure information.

IFT also supports the transfer of data to and from a named pipe under certain conditions.

Table 1-1 shows the IFT file types on various platforms:

Table 1-1 InfoMover File Transfer File Types

File Type	OS/390 and z/OS Platforms		UNIX Platforms		Windows Platforms	
	Source	Target	Source	Target	Source	Target
Flat files on disk	Yes	Yes	Yes	Yes	Yes	Yes
Flat files on magnetic tape	Yes	Yes	No	No	No	No
Named pipes	No	No	Yes ^a	Yes ^a	No	No

a. On UNIX hosts, named pipes can be sources or targets. The participating partners in the transfer can be on any host. Refer to *get* on page 4-23 and *put* on page 4-90 for more information about how to use named pipes as sources and targets.

IFT cannot directly read or write to raw devices.

OS/390 and z/OS Flat Files

IFT provides access to the following types of datasets:

- ◆ Physical sequential (PS) datasets with the following record formats:

F	FBSA	UA	VBSA
FA	FBSM	UM	VBSM
FB	FM	V	VBM
FBA	FS	VA	VM
FBM	FSA	VB	VS
FBS	FSM	VBA	VSA
	U	VBS	VSM

Important: IFT transfers undefined (U) length records as streams of data. It does not preserve the internal blocking structure of the dataset.

- ◆ Partitioned dataset (PDS) members and directories

Important: Although you can transfer a member of a PDS defined with RECFM=U to another dataset, the internal structure of the PDS member is not preserved. Thus, although you can transfer load modules, you cannot use the transferred data as a load module.

- ◆ Generation datasets (GDS)
- ◆ Files on magnetic tape

IFT supports reading datasets with extents on multiple volumes.

Unsupported Datasets

IFT cannot use the following types of OS/390 and z/OS datasets as sources and targets on these platforms:

- ◆ UNIX System Services (USS) files under the Hierarchical File System (HFS)
- ◆ Files with embedded bitmaps or packed decimal data in text mode
- ◆ Striped datasets for IFT created target datasets
- ◆ VSAM (Virtual Storage Access Method) datasets

Transfer Devices

IFT uses devices on a Symmetrix storage subsystem, called *transfer devices* to perform transfers.

More Information: Refer to Chapter 3 in the *InfoMover File Transfer Installation Guide* for an explanation of transfer devices.

Concurrent Connections

EMC recommends that you perform concurrent operations with IFT. With concurrent operations, you start multiple clients on multiple hosts, and perform parallel data operations at the same time.

However, there is a performance trade-off with concurrent operations. EMC recommends that you run no more than eight concurrent operations in a particular *instance* of IFT. If you are running more than eight concurrent connections, you should create another instance of IFT.

Explanation: An instance is a single execution or configuration of IFT. Each instance has its own servers and configuration file. You can create multiple instances on mainframe and UNIX hosts, but not on Windows hosts. Refer to the *InfoMover File Transfer System Administrator's Guide* for information about creating additional IFT instances.

Data Translation

Whenever you transfer a text file or dataset, you can specify that IFT perform data translation; that is, convert ASCII data to EBCDIC or convert EBCDIC data to ASCII. By default, IFT performs translation for text mode transfers on the local host that initiates the transfer, unless that transfer is between an open systems and a mainframe host. In that case, IFT, by default, performs translation on the open systems host.

However, your site can configure IFT to perform the translation on either the local host or the remote host, even if that host is a mainframe system. Consult with your System Administrator to determine how your site performs data translation.

More Information: Refer to the *InfoMover File Transfer System Administrator's Guide* for an explanation of data translation.

EMC TimeFinder/Mirror Software

If the Symmetrix storage subsystem is part of an EMC TimeFinder/Mirror™ software configuration, IFT can read data from a mirrored volume that has been recataloged. However, before you use IFT to read the data, you must use a TimeFinder/Mirror utility to change the volser (volume serial number) of the mirrored volume.

Important: BCVs must not be online or imported to open systems hosts during mainframe TimeFinder/Mirror operations.

Security

IFT uses the standard security features of OS/390, z/OS, UNIX, and Windows systems, and invokes user authentication whenever you connect to a new host.

Whenever you invoke an IFT client (either through the GUI or the CLI), you have to log in with:

- ◆ A username
- ◆ A password

Whenever you use the client to connect to a server, IFT also requires you to enter a username and password for a valid account on the server host.

Important: The maximum length of an IFT password on mainframe hosts is eight characters. The maximum length of an IFT password on open systems hosts is 16 characters.

IFT does not support the use of user IDs such as OS/390 and z/OS “protected” user IDs that do not require a password.

More Information

To know more about IFT than is provided in this guide, refer to the *InfoMover File Transfer System Administrator's Guide*. It describes how to configure and manage an IFT site.

In addition, the *InfoMover File Transfer System Administrator's Guide* provides detailed information about:

- ◆ System architecture
- ◆ File formats suitable for transfer
- ◆ File translation
- ◆ Managing event and error messages

Using the Graphic User Interface

This chapter describes how to use the updated IFT graphic user interface. Topics include:

- ◆ Overview of the IFT Graphic User Interface2-2
- ◆ Invoking the GUI2-3
- ◆ Using the Workspace Tab.....2-4
- ◆ Using the Transfer Table Tab2-21
- ◆ Using the Generate Scripts Tab2-23
- ◆ Exiting the IFT GUI2-26

Overview of the IFT Graphic User Interface

You can invoke the IFT graphic user interface on the following operating systems:

- ◆ Windows 2000
- ◆ Windows NT
- ◆ Windows Server 2003

Although you can run the GUI only on Windows platforms, you can use the GUI to transfer files to or from any IFT server on any platform.

Before you can use the GUI, system management personnel at your site must:

- ◆ Download and install the Java Runtime Environment (part of the Java 2 Standard Edition 1.4 or higher) on your Windows system
- ◆ Install and configure the IFT client software

More Information: Refer to the *InfoMover File Transfer Installation Guide* for an explanation of how to download and install the Java Runtime Environment and configure IFT for a Windows host.

Invoking the GUI

After the IFT servers are started on your host and on the connected hosts you want to use, you invoke the GUI on Windows systems by taking *either* of the following steps:

- ◆ Double-click the GUI icon on your desktop.
- ◆ From the Windows Task Bar, select **Start, Programs, Emc, InfoMover File Transfer, IFT GUI**.

Result: The window shown in Figure 2-1 appears:

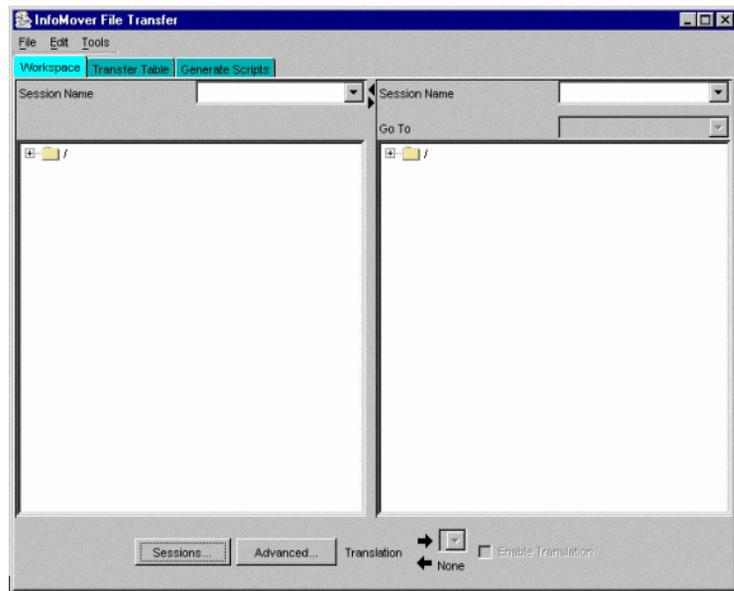


Figure 2-1 The IFT GUI

As Figure 2-1 shows, the IFT GUI has three tabs:

- ◆ Workspace
- ◆ Transfer Table
- ◆ Generate Scripts

The following sections describe how to use these tabs.

Using the Workspace Tab

When you enter the GUI, you see the Workspace tab, as shown in Figure 2-2:

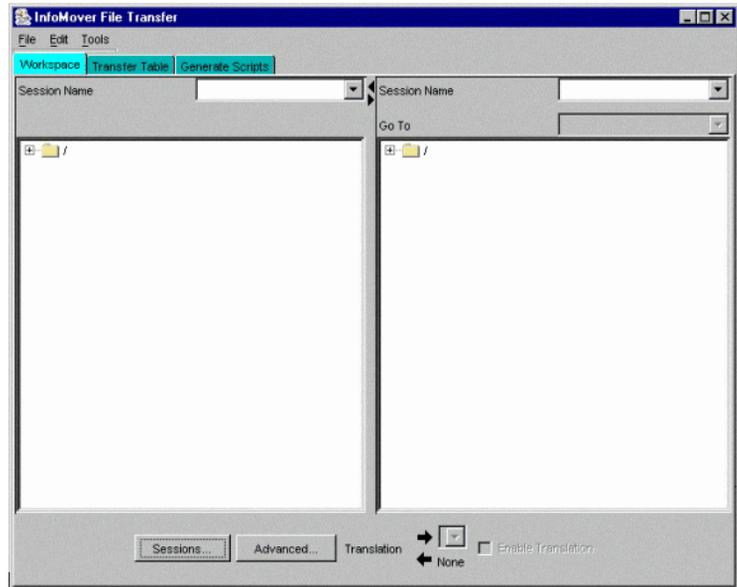


Figure 2-2 Workspace Tab

The Workspace tab allows you to:

- ◆ View and browse the contents of host system directories in a Windows-like directory structure
- ◆ Transfer data directly from source systems to target systems

Background: For a full list of supported systems, refer to the *InfoMover File Transfer System Administrator's Guide* or the *InfoMover File Transfer Installation Guide*.

Understanding Sessions and Transfers

The basic units in the GUI are:

- ◆ The *session*
- ◆ The *transfer*

Each session specifies the source or the target of a transfer. To create a session, you must provide:

- ◆ A name for the session

Note: The session name is an arbitrary alphanumeric string that can be from one to 32 characters long and can include symbolic characters such as underscores (_).

- ◆ Login information (user ID and password)
- ◆ Server name
- ◆ Whether the session is on an open systems or mainframe host
- ◆ Host IP address (open systems only)
- ◆ Primary server port (open systems only)

After you specify at least two sessions you can create a transfer. Each transfer is made up of two sessions, one the source session and the other the target session.

Keep the following points in mind about sessions and transfers:

- ◆ If you are making an open systems to open systems transfer, you can select the sessions in either panel on the Workspace tab. However, the first session you select must be in the left Workspace panel. Otherwise, you receive an error message.
- ◆ If you are making an open systems to mainframe transfer, you must first select the open systems session in the left Workspace panel and then select the mainframe session in the right Workspace panel. You cannot select a mainframe session if you have not already selected an open systems session.
- ◆ You cannot use the IFT GUI to perform a mainframe to mainframe transfer. Use the IFT command line interface (CLI) to perform such transfers. (Refer to Chapter 3 for a description of the CLI.)
- ◆ After you have two sessions open, you can transfer files or datasets in either direction. That is, either displayed session can be a source or a target.

Controlling Sessions

Through the Workspace tab, you can:

- ◆ Add new sessions
- ◆ Browse current sessions
- ◆ Search current sessions
- ◆ Switch between current sessions
- ◆ Delete current sessions
- ◆ Save and reuse sessions
- ◆ Deleting saved sessions

Adding New Sessions

To add sessions in the GUI:

1. Start IFT servers running or make sure that servers are already running for those sessions.

More Information: Refer to the *InfoMover File Transfer System Administrator's Guide* for information about starting IFT servers.

2. Invoke the IFT GUI.

Result: The Workspace tab appears.

3. Take one of the following steps:

- From the GUI menu bar, select **Tools, Sessions**.
- From the Workspace tab, click **Sessions**.

Result: The Session Configuration window appears, as shown in Figure 2-3 on page 2-7:

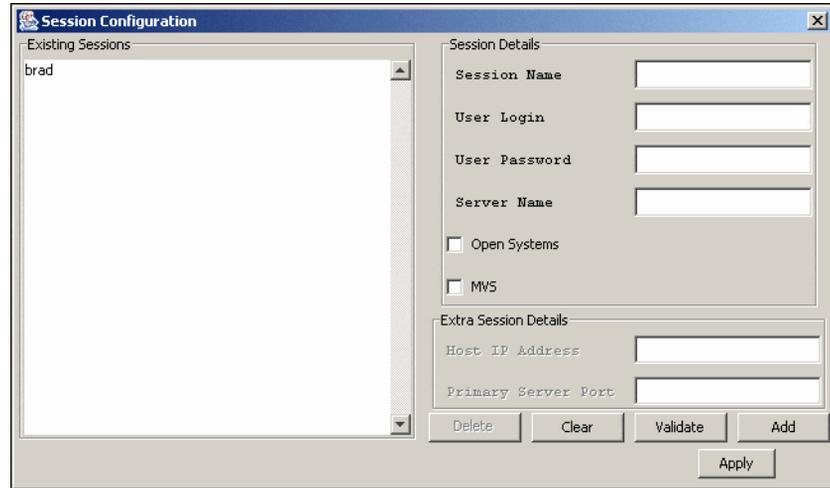


Figure 2-3 Session Configuration Window

4. Under Session Details, enter the proper values, as listed in Table 2-1:

Table 2-1 Session Details

Session Detail	Meaning
Session Name	Session names are a means of identifying a session. Naming is arbitrary and can be up to 32 characters.
User Login	The user ID of the host account you are using for the session.
User Password	The password for the user ID.
Server Name	Enter the name of the IFT server to which you want to connect. The IFT server name is specified during installation by your system administrator. Contact your local technical support for server names.
Host Type	The host type where the session is running. Check boxes for 'Open systems' for any UNIX or Windows platform, or 'MVS' for OS/390 or z/OS platforms.

5. If the session is on an open systems host, enter the extra server details, as listed in Table 2-2:

Table 2-2 Extra Server Detail Configuration Details

Details	Meaning	Applicable Sessions
Host IP Address	The IP address of the session you are adding.	UNIX and Windows
Primary Server Port	The port number on which the IFT server software listens for connections. IFT uses this port for access transfers.	UNIX and Windows

6. Take one of the following steps:
 - If you are satisfied with the information you entered, click **Add**.
 - If you decide you don't want to create the session, click **Clear**.
 - If your session is on an open systems host, you can optionally click **Validate** to verify the login information you entered.

Important: You must have the IFT server running before you click **Validate**.

When the validation completes successfully, click **Add**.

Result: The new session appears in the Existing Sessions list.

7. Click **Apply** to apply the session configuration information.

Result: The new session is added to the current GUI. You can browse it from the Workspace tab.

Browsing Current Sessions

The Workspace tab allows you to browse sessions. To browse a session, take the following steps:

1. Start IFT servers running or make sure that servers are already running for those sessions.

More Information: Refer to the *InfoMover File Transfer System Administrator's Guide* for information about starting IFT servers.

2. Invoke the IFT GUI.

Result: The Workspace tab appears.

Important: As discussed in *Understanding Sessions and Transfers* on page 2-5, you must add the first open systems session in the left panel. For a transfer between an open systems host and a mainframe host, add the open systems session first in the left panel and add the mainframe session afterwards in the right panel.

3. Select the existing session you want to browse from the Session Name pull down menus at the top of the Workspace tab, shown in Figure 2-4:

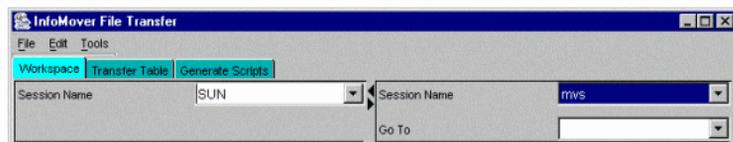


Figure 2-4 Session Pull Down Menus

When the session is displayed, you can browse the sessions' directory contents using the plus (+) and minus (-) icons to expand and minimize directory structures.

Browsing HLQs in Mainframe Sessions

To browse HLQs (high level qualifiers) in mainframe sessions, use the Go To text box on the Workspace tab. The Go To text box applies only to mainframe sessions. The Workspace tab allows you to search any mainframe session (always in the right panel). To search for a directory or file on a current session, take the following steps:

1. Start IFT servers running or make sure that servers are already running for those sessions.

More Information: Refer to the *InfoMover File Transfer System Administrator's Guide* for information about starting IFT servers.

2. Invoke the IFT GUI.

Result: The Workspace tab appears.

3. On the Workspace tab, add new sessions or select existing sessions to make a transfer.
4. Select the session you want to search from the right Session Name pull-down menu.

5. Enter the high level qualifier (HLQ) you want to view in the Go To text box and press ENTER.

Note: The Go To text box also allows you to add HLQs to any source or target mainframe session in which you are browsing.

Result: IFT searches the session for the specified HLQ. If the HLQ is found, the GUI displays it in the session directory tree.

6. Browse the HLQ using the plus (+) and minus (-) icons to expand, minimize, and refresh the HLQ structures until you locate the dataset.

Figure 2-5 shows the detail in the mainframe session:

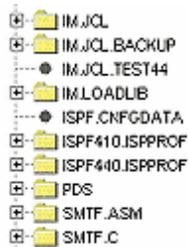


Figure 2-5 Mainframe Session Detail

Note: The GUI only supports HLQs with less than 410 datasets.

Note: If you enter an HLQ that does not exist, you see the following error message: `The connector flagged an error.`

Switching Between Current Sessions

After you add several sessions, you can switch between them so that you can display various combinations of sessions in the Workspace tab. This allows you to make multiple transfers between different sessions.

Take either of the following steps:

- ◆ To change the open systems session displayed in the left panel on the Workspace tab, pull down the left Session Name menu and select the session you want to use.
- ◆ To change the open systems or mainframe session displayed in the right panel on the Workspace tab, pull down the right Session Name menu and select the other session you want to use.

You can now make transfers between the two sessions. After you finish, you may change one or both of the displayed sessions, as described in the previous steps.

More Information: Refer to *Transferring Files* on page 2-13.

Deleting Current Sessions

You can delete current sessions by using the Session Configuration window:

1. Start IFT servers or make sure that servers are already running for those sessions.

More Information: Refer to the *InfoMover File Transfer System Administrator's Guide* for information about starting IFT servers.

2. Invoke the IFT GUI.

Result: The Workspace tab appears. Create your sessions and make your transfers.

3. To delete any of the sessions, take either of the following steps:

- From the GUI menu bar, select **Tools, Sessions**.
- From the Workspace tab, click **Sessions**.

Result: The Session Configuration Window appears, as shown in Figure 2-3 on page 2-7.

4. In the Existing Sessions text block, highlight the session you want to delete.

5. Click **Delete**.

6. When you are finished deleting sessions, click **Apply**.

Result: You return to the Workspace tab.

Saving and Reusing Session Information

Whenever you click **Apply** on the Session Configuration window, IFT saves your current session setting in the Session Settings file. The Session Settings file is stored in your home folder and has the name:

```
iftrc.<username>.<version_number>a.txt
```

where:

`<username>`

Your user name

`<version_number>`

The version number of the software

Important: Do not edit this file.

Any sessions saved to the Session Settings file are available automatically whenever you reinvoke the GUI on the same host with the same user ID. Upon invocation, IFT opens your Session Settings file, and populates the saved session settings from the Session Settings file to the GUI.

Deleting Saved Sessions

To delete any sessions previously saved to the Session Settings file:

1. Start IFT servers or make sure that servers are already running for those sessions.

More Information: Refer to the *InfoMover File Transfer System Administrator's Guide* for information about starting IFT servers.

2. Invoke the IFT GUI.

Result: The Workspace tab appears. The sessions saved to the Session Settings file are available for selection.

3. Take either of the following steps:

- From the GUI menu bar, select **Tools, Sessions**.
- From the Workspace tab, click **Sessions**.

Result: The Session Configuration Window appears, as shown in Figure 2-3 on page 2-7. The Existing Sessions text block displays a list of all previously saved sessions.

4. Follow the steps outlined in *Deleting Current Sessions* on page 2-11 to delete any previously saved session.

Transferring Files

You use the Workspace tab to transfer files between two sessions using standard Windows “drag and drop” techniques. The sessions currently displayed in the two text boxes can be both source and target sessions. In other words, you can perform transfers in either direction.

To transfer a file, take the following steps:

1. Start IFT servers running or make sure that servers are already running for those sessions.

More Information: Refer to the *InfoMover File Transfer System Administrator's Guide* for information about starting IFT servers.

2. Invoke the IFT GUI.

Result: The Workspace tab appears.

3. On the Workspace tab, add new or select existing sessions to make a transfer.
4. Ensure that all necessary translations and transfer settings are set before you transfer files between sessions.

If you need to perform file translation during the transfer, refer to *Performing File Translation* on page 2-15.

If you need to set advanced transfer settings for the transfer, refer to *Specifying Transfer Settings* on page 2-17.

5. Select the file to transfer from either session.

Note: Transfers are limited by UNIX and Windows permissions on files and folders on the target session.

6. Drag the file and drop it on the required location in the opposite session.

Note: You can drop the file on a directory structure or on a named file within a directory.

Result: The Transfer Confirmation window appears to let you confirm your choices. The Transfer Confirmation window displays the following information:

- Source
The full path and name of the file you are transferring.
- Target
The full path and name of the destination file on the target session. You can modify the path and filename in the Transfer Confirmation window if necessary.
- Transfer Tag
The sequential number given to the transfer. You cannot change the transfer tag.
- Translation
The translation file to be applied to the file transfer.

More Information: Refer to the following section, *Performing File Translation* on page 2-15 for more information about file translation.

7. Review the transfer details in the Transfer Confirmation window.
8. Take the appropriate action:

If you want to	then
confirm the transfer	click Confirm .
modify any of the target details	modify these from within the Transfer Confirmation window.
modify any other details or settings	click Cancel to cancel the transfer, and modify your settings from the workspace window.

Result: If you confirm the transfer, IFT transfers the file. When file transfer is complete, you can transfer files again.

Important: IFT allows one file transfer at any given time within a single GUI session.

Performing File Translation

IFT provides the following file translation for data transfers between OS/390, z/OS, AIX, HP-UX, Solaris, and Windows systems:

- ◆ Mainframe EBCDIC to open systems ASCII and ISO Latin
- ◆ Open systems ASCII and ISO Latin to mainframe EBCDIC
- ◆ Binary translation

Translation is disabled for file transfers between UNIX and Windows sessions. Translation is available for file transfers between mainframe and open systems hosts. Although the CLI interface allows for more translation options, the GUI requires that you specify a translation file on the open systems host to perform the translation.

To specify a translation format to apply to InfoMover file transfers other than the default translation file, take the following steps:

1. Start IFT servers running or make sure that servers are already running for those sessions.

More Information: Refer to the *InfoMover File Transfer System Administrator's Guide* for information about starting IFT servers.

2. Invoke the IFT GUI.

Result: The Workspace tab appears.

3. On the Workspace tab, add new sessions or select existing sessions to make a transfer.
4. In the Workspace tab, click the Enable Translation check box, as shown in Figure 2-6 on page 2-16.

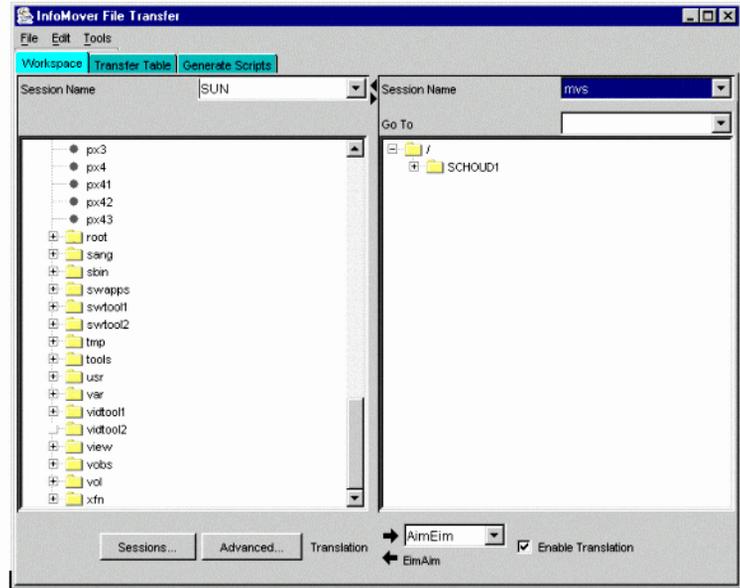


Figure 2-6 Workspace Panel with Enabled Translation

Result: The Workspace panel displays the default translation file for the type of transfer you are performing (AimEim.imxlate for open systems to mainframe transfers and EimAim.imxlate for mainframe to open systems transfers).

5. Take one of the following steps:
 - To use the default, take no action.
 - To use another translation file on the open systems host, select the file from the translation pull-down menu.

Important: When you are performing a transfer between a mainframe and an open systems host, you can only specify a translation file on the open systems server your client is using to initiate the transfer. This is unlike using the IFT CLI where your site can configure translation to be on either the remote or local host, even if that host is a mainframe host.

6. Drag and drop the file you want to transfer.

Result: IFT applies the selected translation file to the transfer.

More Information: Refer to the *InfoMover File Transfer System Administrator's Guide* for further details about translation files.

Specifying Transfer Settings

In addition to translation formats, IFT also allows you to specify other transfer settings. If you do not alter these settings, IFT uses default values. Take the following steps:

1. Start IFT servers running or make sure that servers are already running for those sessions.

More Information: Refer to the *InfoMover File Transfer System Administrator's Guide* for information about starting IFT servers.

2. Invoke the IFT GUI.

Result: The Workspace tab appears.

3. On the Workspace tab, add or select two sessions to make a transfer, as described in *Adding New Sessions* on page 2-6.

4. Take *either* of the following steps:

- Click **Advanced** in the Workspace tab.
- Select **Edit** from the IFT GUI menu bar.

Result: The Transfer Status Settings window appears, as shown in Figure 2-7 on page 2-17:

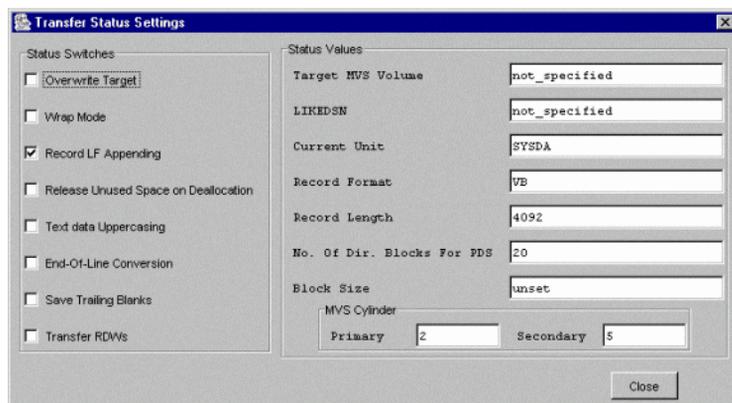


Figure 2-7 Transfer Status Settings Screen

Transfer Status Switches

Table 2-3 lists the status switches you can specify in the Transfer Status Settings window:

Table 2-3 GUI Status Settings

Status Setting	Function
Overwrite Target	Invokes the command to overwrite the target file rather than making a updated copy of it.
Wrap Mode	Enables or disables correct wrapping in a UNIX or Windows source file during transfer.
Rec LF Appending	<i>OS/390 and z/OS Systems Only:</i> Enables or disables the appending of line feeds in OS/390 and z/OS source and target datasets.
Release Unused Space on Deallocation	<i>OS/390 and z/OS Systems Only:</i> Enables or disables automatic release of unused space when deallocating a new dataset.
Text Data Uppercasing	Enables or disables the uppercase conversion of text for a transfer.
End-Of-Line Conversion	Enables or disables automatic conversion of EOL characters (record terminators) on text files being sent between servers on different platforms.
Save Trailing Blanks	<i>OS/390 and z/OS Systems Only:</i> Enables or disables retention of trailing blanks in a mainframe source file during transfer. On all host, trailing blanks retention is disabled by default.
Transfer RDW's	<i>OS/390 and z/OS Systems Only:</i> Enables or disables the retention of mainframe specific record /span descriptor words (RDWS and SDWs) found in variable blocked format datasets during data transfer.
Target MVS Volume	<i>OS/390 and z/OS Systems Only:</i> Specifies the serial number of the volume on which you want to dynamically allocate the target mainframe dataset.
LIKEDSN	<i>OS/390 and z/OS Systems Only:</i> Names a dataset as a model for the allocation attributes to use any subsequent mainframe datasets that are the targets of IFT transfer commands.
Current Unit	<i>OS/390 and z/OS Systems Only:</i> Overrides the value in the default_unit configuration parameter.

Table 2-3 GUI Status Settings (continued)

Status Setting	Function
Record Format	<i>OS/390 and z/OS Systems Only:</i> Specifies the format and characteristics for the records in a new dataset.
Record Length	<i>OS/390 and z/OS Systems Only:</i> Specifies a new length for fixed length records or maximum length for variable-length records. Without an argument, resets the record length to the default_lrecl value in the configuration file that is 4092 bytes.
No of Dir. Blocks for PDS	<i>OS/390 and z/OS Systems Only:</i> Specifies a new number of dir blocks to be allocated for the directory of a partitioned dataset (PDS).
BlockSize	<i>OS/390 and z/OS Systems Only:</i> Specifies the maximum length of a block in bytes for the new mainframe target datasets.
MVS Cylinder	<i>OS/390 and z/OS Systems Only:</i> Specifies a new default number of cylinders to reserve for a target file on a mainframe system. Primary: The number of cylinders to reserve on a mainframe system for a target file's primary allocation extent. Secondary: The number of cylinders to allocate with each new secondary extent.

Mainframe Datasets

As your transfers may involve mainframe datasets, you need to keep in mind how IFT handles such data. IFT can support transfers of non-VSAM datasets. These include:

- ◆ Sequential datasets
- ◆ PDS members
- ◆ GDG members

Each dataset exhibits different behavior when being created or modified.

Sequential Datasets

Sequential datasets act like normal flat files on UNIX (AIX, HP-UX, or Solaris) systems, and can be written to or created using the normal OS/390 and z/OS dataset naming conventions.

PDS Members and Directories

Partitioned dataset (PDS) directories are datasets that contain members. A base member of a partitioned dataset can have the name 'ABC.DEF.GHI'.

This base member can contain member datasets, each of which need a naming convention of:

```
'ABC.DEF.GHI(mem1) . . . ABC.DEF.GHI(memN) ' . . .
```

where:

mem1 . . . memN are alphanumeric characters following the normal mainframe dataset naming conventions (as mentioned previously).

Generation Datasets

For generation datasets (GDS), a base member defines the overall prefix for the generation members. Generation datasets and their naming conventions are discussed in more detail in Appendix B.

Note: For more information about mainframe datasets, refer to *OS/390 and z/OS Flat Files* on page 1-7.

Using the Transfer Table Tab

The Transfer Table tab, as shown in Figure 2-8, allows you to view details about all transfers initiated from the GUI:

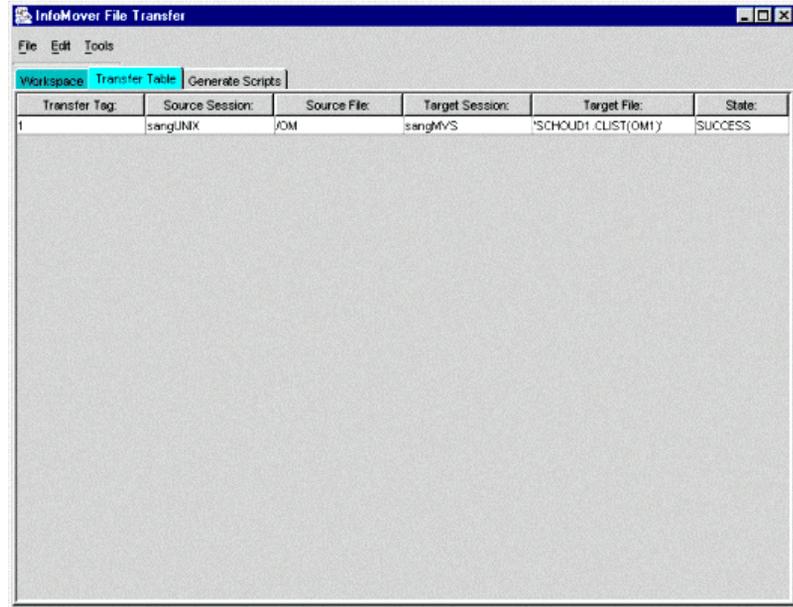


Figure 2-8 Transfer Table Tab

The Transfer Table tab displays details about each transfer you initiate in the GUI. These details are displayed on the Transfer Table in chronological order.

The following details are displayed for each transfer:

- ◆ Transfer Tag — displays the transfer tag number

Important: IFT gives each transfer a sequential number. The first transfer is 1, the second transfer is 2, and so forth.

- ◆ Source Session — displays the name of the source session
- ◆ Source File — displays the full path and name of the file that you are transferring
- ◆ Target Session — displays the name of the target session

- ◆ Target File — displays the full path and name of the file on the target session
- ◆ State — displays the state of the transfer. Possible states are:
 - SUCCESS -- indicates that the file has been successfully transferred to the target session
 - FAILURE -- indicates that the file transfer was not successful

Using the Generate Scripts Tab

The IFT GUI allows you to save file transfers as *scripts* that you can run regularly through the CLI. To do so, you first perform a transfer through the GUI that you want to use as a prototype. You can then save that transfer as a script.

Take the following steps:

1. Invoke the IFT GUI.
2. On the Workspace tab, add or select the sessions you want to use for the script and perform the transfers.
3. Select the Generate Script tab.

Result: You see the Generate Scripts panel shown in Figure 2-9:

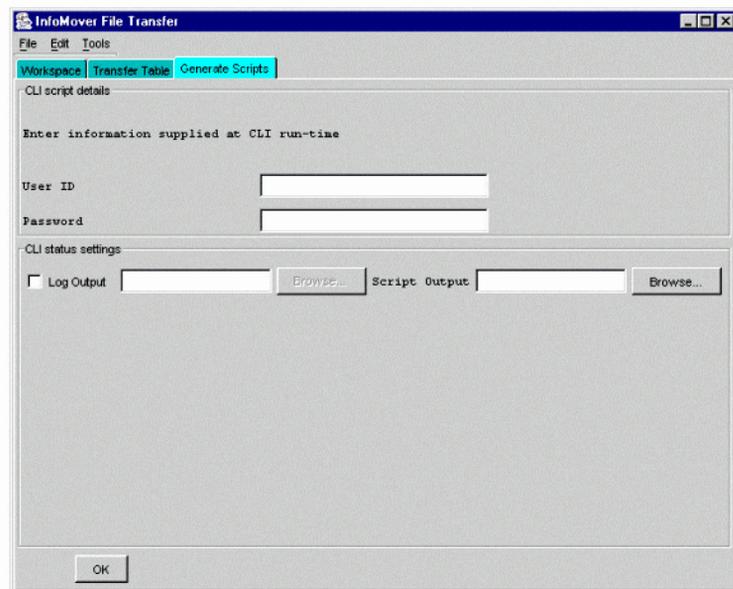


Figure 2-9 The Generate Scripts Panel

4. Enter the user ID of the source session in the User ID text block.
5. Enter the password of the source session in the Password text block.

The maximum length of a client password on a mainframe host is 8 characters. The maximum length of a client password on an open systems host is 16 characters. The characters can be alphanumeric or symbolic.

Important: If the name and password do not match the source session on which you are going to run the script, you cannot run your script in the CLI.

6. To log output from the transfer to a log file, click the Log Output check box. Then, click **Browse**.

Result: A Save dialog box appears, as shown in Figure 2-10:

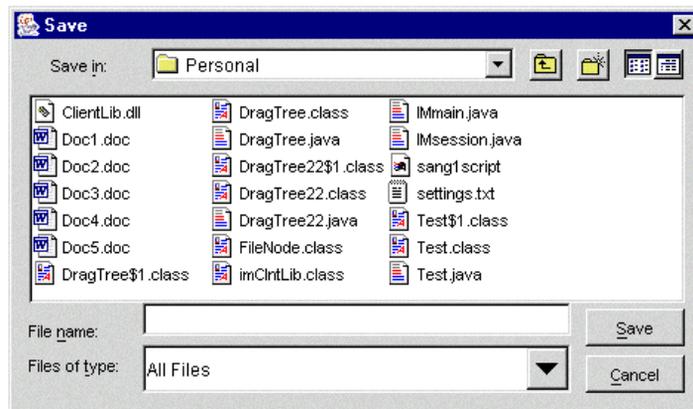


Figure 2-10 Save Dialog Box

7. Navigate to the directory you want to use for logging and specify the log file (either an existing file or a new file).
8. Click **Save**.

Result: You return to the Generate Scripts panel.

9. To save the script to a file, click **Browse** next to the Script Output text box.

Result: A Save dialog box appears.

10. Navigate to the directory you want to use for the script file and specify the file name. You can give the script any file name valid for Windows.
11. Click **Save**.
12. When you return to the Generate Scripts tab, click **OK**.

Result: You can now use the script with the IFT CLI.

More Information: Refer to *Generating a Script with the GUI* on page 3-22.

Exiting the IFT GUI

To exit the IFT GUI, select **File, Exit** from the Menu Bar.

Result: The current IFT GUI session closes.

Closing the GUI does not affect any active transfers that are in progress. If you have active transfers running when you close the GUI, these transfers automatically populate the relevant windows in the GUI if you reopen the GUI on the same system.

Using the Command Line Interface

This chapter describes how to perform data operations with the IFT command line interface (CLI). Topics include:

- ◆ Overview3-2
- ◆ CLI Interactive Mode.....3-3
- ◆ CLI Batch or File Input Mode.....3-19
- ◆ Command Summary3-27

Overview

The command line interface (CLI) is a second way of performing data operations with IFT. The CLI allows you to:

- ◆ Connect to other hosts
- ◆ Change location on your current host or on other hosts to which you are connected
- ◆ Perform data transfers
- ◆ Specify how transferred files are to be handled
- ◆ Specify whether the transferred data is to be translated
- ◆ Disconnect from other hosts at the end of a session

Supported Hosts

You can use the CLI from:

- ◆ OS/390 and z/OS mainframe systems
- ◆ UNIX platforms
- ◆ Windows platforms

CLI Syntax

The CLI is a traditional user interface, consisting of a series of commands you enter in response to a command prompt. The general syntax of a CLI command is:

command <argument> [SPACE<argument> ...]

where:

command

A mnemonic for the command function

<argument>

Value that the command uses to perform its function

In some cases, the arguments are optional. In other cases, the arguments are required. Optional arguments (or parts of arguments) are shown surrounded by square brackets ([]) in syntax statements.

Usage

You can use the CLI in the following ways:

- ◆ Interactive mode
- ◆ Batch or File Input mode

CLI Interactive Mode

To use the CLI to transfer files in interactive mode, follow the procedures outlined in Figure 3-1:

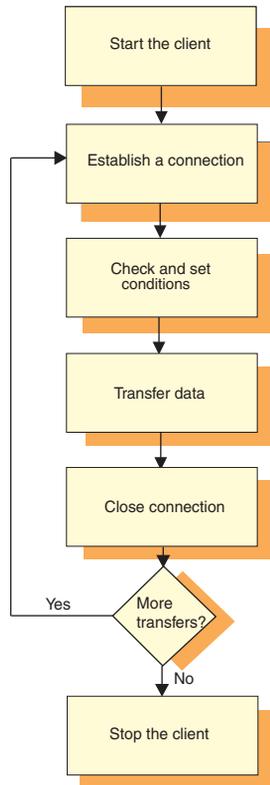


Figure 3-1 Performing Data Operations

As Figure 3-1 shows, you must take the following steps:

1. Start a client.
2. Establish a connection between the IFT servers.

More Information: Refer to *Establishing a Connection* on page 3-10.

3. Check and, if necessary, set transfer conditions.

More Information: Refer to *Checking and Setting Conditions* on page 3-13.

4. Transfer the data.

More Information: Refer to *Transferring Data* on page 3-14.

5. Close the connection.

More Information: Refer to *Closing Connections* on page 3-17.

6. Stop a client.

More Information: Refer to *Stopping a Client* on page 3-17.

Starting a Client

The following sections describe how to start the client with the CLI on:

- ◆ OS/390 and z/OS mainframe systems
- ◆ UNIX platforms
- ◆ Windows platforms

On OS/390 and z/OS Systems

If you are starting a client on an OS/390 or z/OS system, the client must be on one of the hosts involved in the data transfer.

You have three ways to start a client on OS/390 and z/OS systems:

- ◆ You can use the TSO (time-sharing option) **CALL** command to call the INFOMOVR member directly.
- ◆ You can invoke the INFOMOVR CLIST member provided with IFT.

More Information: Refer to the *InfoMover File Transfer System Administrator's Guide* for instructions for modifying the INFOMOVR member for your site.

- ◆ You can run the client as an OS/390 or z/OS batch job.

When you use the first method, invoking INFOMOVR directly, you need to specify:

- ◆ The name of the server to which you want to connect
- ◆ The device ID of the master device (as specified in the IFT configuration file)

Important: If there are multiple versions of IFT at your site, ensure that you are using the correct version of the INFOMOVR CLIST member before you execute it.

Starting the Client with the TSO Call Command

To start the IFT client with the TSO CALL command, take the following steps:

1. From a TSO READY prompt, enter the following command:

```
CALL '<IFT.LOADLIB> (INFOMOVR) ' '<server_name> <master_device>'
```

where:

<IFT.LOADLIB>

The name of the APF-authorized library where the load module is located.

<server_name>

The name of the OS/390 or z/OS server on this host. Server names must be no more than 32 characters long, and must be composed of uppercase alphabetic characters, numeric characters, or symbolic characters, with the exception of slash (/) and semicolon (;).

<master_device>

The device number of the master device specified in the IFT configuration file.

Important: Consult your system programmer or technical support group for the correct values to use for <server_name> and <master_device>. Also, consult your technical support group or the *InfoMover File Transfer System Administrator's Guide* for more information about the APF-authorized library your site uses.

Result: The IFT prompt appears with the current username.

2. You can accept that default username by pressing `ENTER`, or you can enter another username.

Result: You see a prompt for a password:

Password:

3. Enter the password.

Note: Invoke the client with the `ENCPASS` argument for password encryption.

Result: If the username and password are valid, you see the IFT prompt. If the username and password are not valid, you cannot log in.

Hint: Under TSO, when your window becomes full, three asterisks appear (***) at the bottom of the window. Press `RETURN`, `ENTER`, or `CLEAR` to clear the window.

Tip: EMC recommends that your site create a CLIST or REXX EXEC that calls the IFT client with all the appropriate arguments. Your site can use the sample CLIST, `INFOMOVR` member, supplied with the IFT installation kit. The CLIST should then be in the `SYSPROC DD` name.

Starting the IFT Client with CLIST

To start an IFT client from an OS/390 or z/OS (TSO) system with the `INFOMOVR` CLIST member, take the following steps:

1. From a TSO `READY` prompt, enter the following command:

```
%INFOMOVR
```

Result: The IFT prompt appears with the current username.

2. You can accept that default username by pressing `ENTER`, or you can enter another username.

Result: You see a prompt for a password:

Password

3. Enter the password.

Result: If the username and password are valid, you see the IFT prompt. If the username and password are not valid, you cannot log in.

Hint: Under TSO, when your window becomes full, three asterisks appear (***) at the bottom of the screen. Press RETURN, ENTER, or CLEAR to clear the window.

On UNIX Systems

To start a client from a UNIX system, take the following steps:

1. Change to the IFT software directory.

For example, if your IFT software is installed in the default software directory, enter the following command:

```
cd /opt/emc/ift/3.7.mm/bin
```

Tip: Alternatively, you can include the full path of the software directory in the command given in step 2, or you can add the directory to your search path and leave out the ./ in the command given in step 2.

2. Start the client by entering the following command:

```
./infoMover [[<host_IP_address> | s] <port_number> ]
```

where:

<host_IP_address>

The IP address of the host node

s

Connect to the local server running in the same host with the client

<port_number>

The number of the TCP/IP port on the IFT server designated during configuration on the UNIX host

Note: Consult your system administrator or technical support staff for the correct values to use for the <host_IP_address> and <port_number>.

Result: You are prompted for a username (to match an account on the server host to which you are connecting):

```
UserName (Username) :
```

where:

```
Username
```

A default username

3. Accept that default username by pressing **ENTER**, or enter another username.

Result: You see a prompt for a password:

```
Password:
```

4. Enter the password.

Note: The maximum length of an IFT password on UNIX hosts is 16 characters.

Result: If the username and password are valid, you see the IFT prompt. If the username and password are not valid, you cannot log in.

On Windows Systems

The IFT command window allows you to run a CLI with different arguments from the default you access through the CLI menu choice or the CLI shortcut on the desktop.

Starting the CLI From the Start Menu

Take the following steps:

1. Click **Programs** from the Start menu.
2. Click **EMC** from Programs menu.

Result: You see a menu of installed EMC software.

3. Click **InfoMover File Transfer**.

Result: You see the IFT menu.

4. Click **IFT Command Window** from the IFT menu.

Result: You see a command window. You are in the `/bin` folder in the IFT directory.

5. At the prompt, enter the following command:

```
infoMover [[<host_IP_address> | s] <port_number>]
```

where:

<host_IP_address>

The IP address of the host node

s

An option that specifies that you want to connect to the local server running on the same host as **s** the client

<port_number>

The number of the TCP/IP port on the IFT server designated during configuration on the Windows host

Consult your system administrator or technical support staff for the correct values to use for the <host_IP_address> and <port_number>.

Result: You are prompted for a username (to match an account on the server host to which you are connecting):

```
UserName (Username) :
```

where:

Username

A default username

6. Accept that default username by pressing ENTER, or enter another username.

Result: You see a prompt for a password:

```
Password:
```

7. Enter the password.

Note: The maximum length of an IFT password on Windows hosts is 16 characters. The password you use must be nonblank.

Result: If the username and password are valid, you see the IFT prompt. If the username and password are not valid, you cannot log in.

Using the Verbose Mode

When you invoke the client through the CLI interface, you are in *verbose mode* by default. In verbose mode, the CLI echoes any command you enter, and displays detailed information about the results as follows:

```
command <argument>
Command: command argument
This is a detailed display of the results.
```

Background: Most of the examples in this chapter are given in verbose mode.

You can disable verbose mode by issuing the **verbose OFF** CLI command. You can also disable verbose mode by issuing a **verbose** command without an argument whenever verbose mode is enabled.

Establishing a Connection

Before you perform any data operations between hosts with the client, you need to establish a connection between the IFT servers on the hosts. Take the following steps:

1. Enter the following command:

```
open <server>
```

where:

```
<server>
```

The server name of up to 32 characters

Result: You are prompted for a username for the other server.

2. Enter a valid username.

Result: You are prompted for the password.

3. Enter the password.

Note: The maximum length of an IFT password on mainframe hosts is eight characters. The maximum length of an IFT password on open systems hosts is 16 characters.

Result: You see the IFT prompt and the server name.

Important: Entering a username and password puts you into the default software directory defined during configuration.

Client-Server Connections

Generally, IFT allows you to connect to a server on the same host as your client. In addition, if you are starting a client:

- ◆ On a UNIX system:

The client can be on a host connected through TCP/IP to the host of one of the servers involved in the data transfer. The server host can be running any supported UNIX or Windows operating system.

- ◆ On a Windows system:

The client can be on a host connected through TCP/IP to the host of one of the servers involved in the data transfer. The server host can be running any supported UNIX or Windows operating system.

Valid Client Locations Table 3-1 shows valid client locations for various types of transfer:

Table 3-1 Valid Client Locations

Source Host Operating System	Target Host Operating System	Client Location
OS/390 or z/OS	OS/390 or z/OS	Can be on: <ul style="list-style-type: none"> • Source host • Target host
OS/390 or z/OS	UNIX	Can be on: <ul style="list-style-type: none"> • Source host • Target host • Remote UNIX system connected through TCP/IP to the target host • Remote Windows system connected through TCP/IP to the target host
OS/390 or z/OS	Windows	Can be on: <ul style="list-style-type: none"> • Source host • Target host • Remote UNIX system connected through TCP/IP to the target host • Remote Windows system connected through TCP/IP to the target host
UNIX	UNIX	Can be on: <ul style="list-style-type: none"> • Source host • Target host • Remote UNIX system connected through TCP/IP to either the source or target host • Remote Windows system connected through TCP/IP to either the source or target host

Table 3-1 Valid Client Locations (continued)

Source Host Operating System	Target Host Operating System	Client Location
UNIX	OS/390 or z/OS	Can be on: <ul style="list-style-type: none"> • Source host • Target host • Remote UNIX system connected through TCP/IP to the source host • Remote Windows system connected through TCP/IP to the source host
UNIX	Windows	Can be on: <ul style="list-style-type: none"> • Source host • Target host • Remote UNIX system connected through TCP/IP to either the source or target host • Remote Windows system connected through TCP/IP to either the source or target host
Windows	Windows	Can be on: <ul style="list-style-type: none"> • Source host • Target host • Remote UNIX system connected through TCP/IP to either the source or target host • Remote Windows system connected through TCP/IP to either the source or the target host ^a
Windows	OS/390 or z/OS	Can be on: <ul style="list-style-type: none"> • Source host • Target host • Remote UNIX system connected through TCP/IP to the source host • Remote Windows system connected through TCP/IP to the source host ^a
Windows	UNIX	Can be on: <ul style="list-style-type: none"> • Source host • Target host • Remote UNIX system connected through TCP/IP to either the source or target host • Remote Windows system connected through TCP/IP to either the source or target host ^a

- a. You do not need to be an administrator on a Windows system to connect to a server through a remote Windows client. You only need to be added to the IFT group, and be a Power User on the Windows server.

Detailed information about each command is provided in Chapter 4. Many of the command reference pages in Chapter 4 refer to the hosts involved in a transfer as host A and host B. Generally, host A is the local host on which the client resides and host B is a remote host. However, as shown in Table 3-1, you can be using a remote client on

a third host. Thus, both servers involved in the data transfer may be remote from the client.

Checking and Setting Conditions

Before you transfer data, you may need to:

- ◆ Change the current working directory or file group on the source host
- ◆ Check the current settings
- ◆ Change any settings that are not valid for this data transfer

Consult Chapter 4 for more information about the CLI commands you can use to check and set conditions.

Translating Text Mode Data

Text mode data translation is a special case. By default, IFT performs translation for text mode transfers on the local host (host A) unless the transfer is between an open systems and a mainframe host, in which case IFT performs translation on the open systems host - even if the open systems host is the remote host.

However, your site can configure IFT (through configuration file parameters) to perform the translation on either the local host (host A) or the remote host (host B), even if that host is a mainframe system. You can also use CLI commands to override the configured translation procedure for particular transfers.

Following Your Site's Translation Procedure

To follow either the default translation procedure or whatever translation procedure your site has configured, take the following steps:

1. Enable translation by issuing the **imxlate ON** CLI command or **imxlate** without an argument.
2. If necessary, specify any other conditions you need to make for the translation. (using the **imaddlf**, **imcrlf**, or **imrdw** CLI commands).
3. Follow the procedures in *Transferring Data* on page 3-14.

Changing Your Site's Translation Procedure

To change the translation procedure from the default or from whatever translation procedure your site has configured, take the following steps:

1. Enable translation by issuing the **imxlate** CLI command.
 - To specify translation on the local host to which your client is connected (host A), use the **LOCAL** argument.
 - To specify translation on the remote host (host B), specify the **REMOTE** argument.
2. Issue the **imrcvx** or **imsendx** CLI commands to specify the translation tables or files you want to use.

Background: A translation table is a set of characters used for translating data. On open hosts, each translation table is contained in its own translation file. You refer to the translation tables by the translation filename. On mainframe hosts, all translation tables are contained in a sequential dataset or a member of a PDS. You refer to the translation table by the name given it in the dataset.

If you issue **imxlate** with **LOCAL** or **REMOTE** and issue **imrcvx** or **imsendx** without specifying a translation file or table, IFT uses the translation file or table defined in the configuration file on the host specified by that argument, local or remote.

3. If necessary, specify any other conditions you need for the translation. (using the **imaddlf**, **imcrlf**, or **imrdw** CLI commands).
4. Follow the procedures in *Transferring Data* on page 3-14.
5. Use the **status** command to find out whether translation is being done on the local host or a remote host and what **imsendx** and **imrcvx** translation files or tables are being used.

More Information: Refer to the *InfoMover File Transfer System Administrator's Guide* for more information.

Transferring Data

IFT supports transferring datasets or files that are structured as a continuous sequence of data bytes.

If you transfer a dataset from a mainframe host to an open systems host and then back to the mainframe system, the record format of the returned dataset may differ from that of the original dataset. The data

is preserved, but the structure and size of the file may change. However, you do not lose any data integrity with this possible change.

Keeping Track of Data Characteristics

When you transfer data, you should know the characteristics of the data. For example, if you transfer a dataset from a mainframe host to an open systems host in binary format, the target data must be read by an application that understands the binary format without translation. Otherwise, the results are unpredictable, and you may receive an error message.

Using the `get` Command

You use the `get` CLI command to transfer the data in a source from a storage device on host B to a target on host A, when the client is controlling data operations through host A.

More Information: Refer to `get` on page 4-23.

Using the `mget` Command

You use the `mget` CLI command to transfer files that match a pattern (wildcards, pathnames, or pieces of filenames and extensions) you specify from a storage device on host B to a target on host A, when the client is controlling data operations through host A.

More Information: Refer to `mget` on page 4-81.

Using the `put` Command

You use the `put` CLI command to transfer a the data in a source on host A to a target on host B when the client is controlling data operations through host A.

More Information: Refer to `put` on page 4-90.

Using the `mput` Command

You use the `mput` CLI command to transfer files that match a pattern (wildcards, pathnames, or pieces of filenames and extensions) you specify from a storage device on host A to a target on host B when the client is controlling data operations through host A.

More Information: Refer to `mput` on page 4-84.

Concurrent Transfer Operations

All IFT servers can initiate a maximum of eight concurrent connections for data operations (information commands or transfers) across a given set of transfer devices. To transfer files or datasets concurrently, take the following steps:

1. Make sure that an IFT server has been started on each host to be involved in the data operation.

More Information: Refer to *Overview* on page 3-2.

2. Start a client for each transfer operation you want to perform concurrently.

More Information: Refer to *Starting a Client* on page 3-4.

3. From each client, establish an IFT connection between the servers to be involved in that client's transfer operation.

More Information: Refer to *Establishing a Connection* on page 3-10.

4. From each client, use the **get** or **put** CLI commands to perform the transfer.

Incomplete Transfer Operations

If a transfer operation fails, IFT generates an error message.

More Information: Refer to Appendix A, *Error and Event Messages* for an explanation of IFT errors and a list of error and event messages.

If the failure occurs on one host, IFT attempts to recover. If the failure occurs on both hosts or on one host and the Symmetrix storage subsystem, you must take the actions described in step 1.

1. After an incomplete transfer, take the appropriate action(s):

If	then
the failure occurred on both hosts or on one host and the Symmetrix storage subsystem,	<ol style="list-style-type: none"> 1. Stop the IFT servers. 2. Reformat the IFT transfer devices. 3. Start the IFT servers. <p>Refer to the <i>InfoMover File Transfer System Administrator's Guide</i>.</p>

If	then
the target host is an OS/390 or z/OS host, and the transfer is halted because the mainframe client cancelled the operation or the operation has abnormally terminated,	An existing target dataset is truncated to zero, but is not deleted. So delete the target. a new target dataset is deleted.
the target host is a UNIX or Windows system, and the transfer is halted because the OS/390 or z/OS client cancelled the operation or the operation has abnormally terminated,	The target is removed. Go to step 2.
you are transferring a large file (2 GB or more) to or from a 32-bit UNIX system that is not enabled for large files,	Break the file into several files, and transfer those smaller files.

2. Restart the transfer.

Closing Connections

You must close a connection from the client session that opened that connection. At the IFT prompt, enter the following CLI command:

```
close
```

More Information: Refer to *close* on page 4-11 for a description of the **close** command.

Stopping a Client

You must stop a client from the system where it was started. At the IFT prompt, enter the following CLI command:

```
quit
```

The **quit** command ends the IFT session, and exits from the CLI client. The **quit** command is the same for all hosts.

More Information: Refer to *quit* on page 4-98 for a description of the **quit** command.

Explanation: A CLI client times out after the client is idle for the period of time specified in the `client_timeout` configuration parameter. You can set the `client_timeout` configuration file only by changing that argument in the IFT configuration file. Refer to the *InfoMover File Transfer System Administrator's Guide* for information about setting configuration parameters.

After you have finished using the IFT client, you can shut it down.

CLI Batch or File Input Mode

You can automate your use of CLI commands to control IFT data operations by creating batch or script files whenever you need to perform the same file transfer regularly.

You can create batch and script files in any of the following ways:

- ◆ Manually
- ◆ Through the GUI client
- ◆ Through the non-privileged IFT Management Utility

If you need to maintain security in your use of scripts, be aware that the scripts you create manually or through the GUI client have clear text passwords. To encrypt the passwords in these scripts, you need to use the IFT Management Utility, as described *Encrypting Script File Passwords* on page 3-23.

If you create scripts through the IFT Management Utility, you do not need to take this extra step. The IFT Management Utility encrypts passwords as part of the script creation process.

Creating a Script Manually

You can create batch and script files manually with the editor of your choice. Within each batch or script file, you follow the same steps you do when you invoke CLI commands interactively. In other words, you must take the following steps:

1. Start a IFT client.

More Information: Refer to *Starting a Client* on page 3-4.

2. Establish a connection between the IFT servers.

More Information: Refer to *Establishing a Connection* on page 3-10.

3. Check and, if necessary, set data transfer conditions.

More Information: Refer to *Checking and Setting Conditions* on page 3-13.

4. Make the transfer.

More Information: Refer to *Transferring Data* on page 3-14.

5. Close the connection with the **close** command.

More Information: Refer to *Closing Connections* on page 3-17.

6. Stop the client with the **quit** command.

More Information: Refer to *Stopping a Client* on page 3-17.

Mainframe Security Issues

You should keep security in mind if you write your own JCL to run a client. If the first statement in the SYSIN stream is blank, IFT uses the access privileges of the user ID that submitted the batch job to establish the security environment of the client.

This means that if you submit a batch job with a blank first SYSIN line (with no user ID or password specifications) from TSO, IFT uses the user ID and group ID of the TSO user as the security access for the client.

However, if you include any valid local user ID and password pair as the first two lines in SYSIN, IFT uses the security access of the specified local user ID as the security access for the CLI client.

OS/390 and z/OS Systems

You can run the IFT CLI interface as a mainframe BATCH job, but you must supply the JCL.

UNIX Systems

On any UNIX system, you can write and use your own scripts to execute CLI commands. The following example is a sample script that illustrates how to run an IFT client. This sample shows the script and control statements necessary to:

- ◆ Log on to the local host.
- ◆ Connect to the remote OS/390 or z/OS host, `ILAB07`, and navigate to a desired high-level qualifier.
- ◆ Check on current status and the directory contents.
- ◆ Transfer `f1.txt` from the local host to `var.test.jcab.lsw169.p1` on the remote host.

The script in Example 3-1 is a manually-written script. You can run it on any UNIX host:

Example 3-1 Sample UNIX IFT Script

```
#!/bin/csh -f
##
## Sample UNIX IFT Script
##
## NOTE:
## This sample script will run under 'csh'; other UNIX shells/languages
## may also be used to automate your process.
##
## This script will execute an instance of the IFT Client software,
## piping in a 'commandFile' which contains an opening of a log file,
## the IFT Server login, a series of commands, a quit and a close. This
## 'commandFile' is simply a text file which may look something like the
## following:
## =====
## =====
#
## Setup environment variables:
COMMAND_FILE='/opt/ift/3.6.mm/bin/commandFile'
IFT_CLIENT='/opt/ift/3.6.mm/bin/InfoMover'
IFT_SRVR_PORT='5572'
IFT_SRVR_HOST='hostXYZ'
##
## Now, execute the IFT client, piping in the commands (contained
## in the commandFile). All screen output will also be sent to the log
## file indicated above.
$IFT_CLIENT $IFT_SRVR_HOST $IFT_SRVR_PORT < $COMMAND_FILE
status = $?
if [ $status -ne 0 ]; then
echo 'Level '$status' error detected from InfoMover client execution'
fi
##
## End of Sample UNIX EMC IFT Script
###=====
## Sample UNIX IFT Command File
##
## This file can be directed as
## input to the IFT client
###=====
##
## Login to local server
###
root
stormy
###
## Open a remote connection
###
open ILAB07
APPLA
foobar
replace on
cd 'var.test.jcab.lsw169.p1'
status
put /opt/jcaal/fl.txt 'var.test.jcab.lsw169.p1'
###
## Close connection
###
close
quit
```

Windows Systems In a Windows environment, you can write and run your own script files to invoke the CLI client interface with redirected input and output.

Example Command Line

The following text is an example of a command line to execute such a script file.

```
releasecpp\infomover.exe foobarlab1 5592 < auto_ti_101.scr > \ au_to_101.out
```

Sample Batch File Contents

The content of the `auto_ti_101.scr` script file could be something like that shown in the following example:

```
open labmec
userid
password
get /usr/dtest/file2 d:\temp\imData\file2copy
close
quit
```

Generating a Script with the GUI

You can generate a script file, executable through the CLI client, from the actions you take during a transfer session on the GUI. This action allows you to create a template of a commonly-performed transfer automatically.

Important: You can generate a script through the GUI client on Windows hosts. You can run the script through the CLI client on AIX, HP-UX, Solaris, or Windows hosts.

To create a script through the GUI client, follow the steps outlined in *Using the Generate Scripts Tab* on page 2-23 to generate the script. You can then use the saved script with the CLI.

Keep in mind, however, that the password for this script is in clear text. It is not encrypted. Follow the steps outlined in *Encrypting Script File Passwords* on page 3-23 to encrypt the password.

Generating a Script with the IFT Management Utility

To create a new script file using the non-privileged IFT Management Utility, take the following steps:

1. On any account with access to the IFT Configuration file, navigate to the IFT software directory, and log into the non-privileged IFT Management Utility, as follows:

```
./imManager -u (On UNIX)
```

imManager -u (On Windows)

where

-u

Allows non-privileged users to access the non-privileged IFT Management Utility

Result: The IFT Management Menu appears.

2. Select **(6) Create/Modify InfoMover Script Files**.
3. When prompted, enter the name of the new script file.
4. Enter your username.
5. Enter your password.

Note: The password is encrypted automatically.

6. Enter the IFT command strings you need for the transfer.

Result: IFT creates a script file in the location that you requested. The password is encrypted, and no longer stored in a Cleartext format.

7. Select **quit** to return to the IFT Management Menu.
8. Select **X** to exit the IFT Management Menu.

If you have a root or administrator account on a UNIX or Windows system or if you are on a mainframe account that is authorized to run the IFT Management Utility and have read/write access to the configuration dataset(s), you can use the full IFT Management Utility to perform this task.

More Information: Refer to the *InfoMover File Transfer System Administrator's Guide* for details about using the IFT Management Utility.

Encrypting Script File Passwords

If you create a script file manually, or through the IFT GUI, the password in that script file is not encrypted, but is in clear text. For security reasons, you may want to encrypt the password in a script file.

Take the following steps through the non-privileged IFT Management Utility:

1. On any account with access to the IFT Configuration file, navigate to the IFT software directory, and log into the non-privileged IFT Management Utility, as follows:

```
./imManager -u (On UNIX)
```

```
imManager -u (On Windows)
```

where

```
-u
```

Allows non-privileged users to access the non-privileged IFT Management Utility

Result: The IFT Management Menu appears.

2. Select **(6) Create/Modify InfoMover Script Files.**
3. Select **(2) Modify existing password file.**

Result: You are asked if you would like the existing script file backed up.

4. Select **y.**

Result: You see the following message:

```
Do you want to modify an existing cleartext password file or an existing encrypted password file? (1 for cleartext password file/0 for encrypted password file):
```

5. Select **1** to encrypt a clear text password file.

Result: You see the following message:

```
Enter the name of the script_file you want to edit. If the command/script file is in any directory other than the current directory, enter the full path of the command script file:
```

6. Enter the name or path of the script file.

7. **Result:** You see the message:

```
The filename file is successfully changed and a copy is stored in backupIMscripts directory
```

The script file you specified now has an encrypted password file.

8. Select **quit** to return to the IFT Management Menu.
9. Select **X** to exit the IFT Management Menu.

You can also use the privileged IFT Management Utility to perform this task.

More Information: Refer to the *InfoMover File Transfer System Administrator's Guide* for details about using the privileged IFT Management Utility.

Running a Script

Take one of the following steps when you are ready to use the script (on the host you specified for its use):

On Mainframe Hosts

To use the script file you created as input, invoke the client program load module INFOMOVR with ENCPASS as the argument so that the client accepts encrypted passwords. For example:

```
CALL 'IFT.LOADLIB(INFOMOVR)' 'FOSERVER 1234 ENCPASS'
```

where:

FOSERVER

The name of the initiator server component

1234

The master device

ENCPASS

The ENCPASS argument

On UNIX Hosts

To use a script file with either a clear text or encrypted password, take the following steps:

1. Change to the IFT software directory.
2. Start the script by entering the following command:

```
./infoMover [-e] [[<host_IP_address>| s] <port_number>] <script>
```

where:

-e

Required if the password in the script is encrypted

<host_IP_address>

The IP address of the host node

s

Required if you want to connect to the local server running on the same host as the client

<port_number>

The number of the TCP/IP port on the IFT server designated during configuration on the UNIX host

<script>

The path and name of the script

On Windows Hosts

To use a script file with either a clear text or encrypted password, take the following steps:

1. Click **Start, Programs, EMC, InfoMover File Transfer, IFT Command Window**.

Result: You see a command window. You are in the `/bin` folder in the IFT directory.

2. At the prompt, enter the following command:

```
infoMover [-e] [[<host_IP_address> | s] <port_number>] <script>
```

where:

-e

Required if the password in the script is encrypted

<host_IP_address>

The IP address of the host node

s

Required if you want to connect to the local server running on the same host as the client

<port_number>

The number of the TCP/IP port on the IFT server designated during configuration on the Windows host

<script>

The path and name of the script generated by the GUI

Command Summary

Table 3-2 lists the CLI client commands, their associated arguments, their interdependencies, and the hosts on which they are valid. Refer to Table 3-1 on page 3-11, and the note that follows it, for a definition of host A and host B:

Table 3-2 Client Command Relationships

Command	Purpose	Dependency ^a	Side Effects ^b	Host Availability ^c
bin [ary]	Enables data translation byte-by-byte without translation.		Turns imxlate OFF.	
blksize [bytes]	Specifies the maximum length of a block (in bytes) to reserve for a target dataset.		Overrides the value in the default_blksize configuration parameter for new dataset allocations.	New OS/390 and z/OS targets only.
cd [name]	Changes the working directory or file group on host B when the client is controlling data transfer through host A.			
close	Terminates the connection between two IFT servers without disconnecting the client.			
del filename	Deletes a file that is located on a remote IFT server's system.	The IFT client must be connected to the remote server through which the specified file is to be deleted.		
dir [name]	Lists directory entries or files in a file group on a host B when the client is controlling data transfer through host A.			
dir [name][>dataset]	Redirects (or pipes) the directory listing to a specified output dataset.			OS/390 or z/OS only.
get source target	Transfers a file from host B to host A when the client is controlling data transfer through host A.	The IFT client must be connected to the remote server through which the specified file is to be retrieved.		

Table 3-2 Client Command Relationships (continued)

Command	Purpose	Dependency ^a	Side Effects ^b	Host Availability ^c
help ? [<i>command</i>]	Prints help information.			
imaddlf [ON OFF]	Enables or disables the appending of record linefeeds in OS/390 and z/OS mainframe sources and targets.	The imrdw command should be cleared (OFF) to enable setting the imaddlf command.	Setting the imaddlf OFF command also turns off the imwrap command.	OS/390 or z/OS only.
imbufno [<i>size</i>]	Specifies a new number of buffers to assign to the Data Control Block (DCB).		Overrides the value in the default_imbufno configuration parameter.	OS/390 or z/OS only.
imcrlf [ON OFF]	Enables or disables the automatic conversion of record terminator characters during text file transfer between different platforms.			
imcyl [<i>inder</i>] [<i>primary</i> [, <i>secndry</i>]]	Specifies the number of cylinders to reserve for a target file on a local or remote mainframe system. Without an argument, resets the default number of cylinders to reserve for a target file to any default_cylinders parameter value in the configuration file.		Overrides (or resets to) the value in the configuration file default_cylinders parameter.	New OS/390 and z/OS targets only.
imdirblocks [<i>blocks</i>]	Sets a value for the default number of directory blocks to be allocated for the directory of a new partitioned data set (PDS).		Overrides the value in the configuration file for the default number of directory blocks to be allocated for the directory of a new partitioned data set (PDS).	New OS/390 or z/OS target only.
imfilenumber [<i>sequence</i>]	Specifies a new relative position of a dataset on a tape volume.		Overrides the value in the default_imfilenumber configuration parameter for new dataset allocations.	New OS/390 or z/OS target only.
imlog [ON <i>logfile</i> OFF]	Sets or clears the writing of output to a log file and specifies the log filename.			Open systems only.

Table 3-2 Client Command Relationships (continued)

Command	Purpose	Dependency ^a	Side Effects ^b	Host Availability ^c
imrdw [_transfer] ON/OFF	Determines whether mainframe-specific record descriptor words (RDWs), found in variable (V) or variable blocked (VB) datasets, are retained on transfer.	The imxlate and imaddlf commands must be OFF.		Either local or remote server must be OS/390 or z/OS.
imrcvx [late] [translation_file]	Specifies the single-byte translation file to use to translate text files received by the client system.	The imrdw command must be OFF.	Turns the imxlate command ON.	All. ^d
imrelease [ON OFF]	Enables or disables automatic release of unused space when deallocating a dataset.			New OS/390 and z/OS targets only.
imsendx [late] [translation_file]	Specifies the single-byte translation file to use in translating text files sent by the client system.	The imrdw command must be OFF.	Turns the imxlate command ON.	All. ^d
imtrail [blanks] [ON OFF]	Enables or disables retention of trailing blanks in an OS/390 or z/OS source file during transfer.	The imxlate command must be ON.		OS/390 and z/OS Fixed (RECFM=F) or Fixed Block (RECFM=FB) source file.
imwrap ON OFF	Enables or disables correct wrapping of lines in a UNIX or Windows source file transferred to an OS/390 or z/OS mainframe target dataset.	The imaddlf command must be ON.		New OS/390 and z/OS targets only.
imx [late] [ON LOCAL REMOTE OFF]	Enables the transfer of data with single-byte character translation.	The imrdw command must be OFF.	The imxlate ON command turns text mode translation and the imaddlf command on. The imxlate LOCAL command specifies translation to be done on the local host. The imxlate REMOTE command specifies that translation will be done on the remote host. The imxlate OFF command turns text mode translation and the imaddlf command OFF.	All. ^d

Table 3-2 Client Command Relationships (continued)

Command	Purpose	Dependency ^a	Side Effects ^b	Host Availability ^c
lcd [<i>name</i>]	Changes the working directory or file group on host A when the client is controlling the transfer through host A.			
ldir [<i>name</i>]	Lists directory entries or a list of files in a file group on host A when the client is controlling the connection through host A. This command also provides additional information about files.			
likedsn [<i>model</i>]	Names a dataset as a model for the allocation attributes to use for any subsequent OS/390 and z/OS mainframe datasets that are the targets of IFT transfer commands.			New OS/390 and z/OS targets only.
lpwd	Displays the name of the current working directory on host A when the client is controlling the connection through host A.			
lrecl [<i>bytes</i>]	Sets a new record length value. Overrides the default record length value in the IFT configuration file for new dataset allocations (the <code>default_lrecl</code> configuration parameter).		Overrides the default record length value in the IFT configuration file for new dataset allocations (the <code>default_lrecl</code> configuration parameter).	New OS/390 and z/OS targets only.
lsp [<i>volser</i>]	Lists the available space on a specified volume on a remote OS/390 or z/OS mainframe.			Mainframe only.
mget [<i>pattern</i>]	Retrieves files that match the wildcards described in <i>pattern</i> from host B to host A when the client is controlling the connection through host A.	The IFT client must be connected to the remote server through which the specified files are to be retrieved.		

Table 3-2 Client Command Relationships (continued)

Command	Purpose	Dependency ^a	Side Effects ^b	Host Availability ^c
mput [<i>pattern</i>]	Sends local files that match the wildcard(s) described in <i>pattern</i> from host A to host B when the client is controlling the connection through host A.	The IFT client must be connected to the remote server through which the specified files are to be sent.		
open <i>server</i>	Opens an IFT server (login required).			
prompt [ON OFF]	Enables or disables interactive prompting.			Not applicable to OS/390 and z/OS in batch.
put <i>source target</i>	Copies a file from the current host to another host.	The IFT client must be connected to the remote server to which the specified file is to be sent.		
pwd	Displays the name of the working directory on a connected host.			
qu [<i>it</i>]	Disconnects from the connected host, and ends the client process.			
rcfm [<i>format</i>]	Overrides the value in the default record format value in the IFT configuration file for new dataset allocations (the default_rcfm configuration parameter).			New OS/390 and z/OS targets only.
rep [<i>lace</i>] [ON OFF]	Enables or disables file replacement mode.		Clears volume settings.	New OS/390 and z/OS targets only.
stat [<i>us</i>]	Displays the current settings for file transfer.			

Table 3-2 Client Command Relationships (continued)

Command	Purpose	Dependency ^a	Side Effects ^b	Host Availability ^c
unit [<i>unitname</i>]	Overrides the value in the configuration file <code>default_unit</code> parameter to specify a new default unit. Without an argument, resets the unit name to any <code>default_unit</code> parameter value in the configuration file.			New OS/390 and z/OS targets only.
unitcount [<i>unit_count</i>]	Specifies the number of units to be allocated for a new OS/390 or z/OS dataset allocation.			New OS/390 and z/OS targets only.
up [<i>per</i>] [ON OFF]	Enables or disables the uppercase conversion of text.	The imxlate command must be ON.		
verb [<i>ose</i>] [ON OFF]	Enables or disables display of output on the window.			
volcount [<i>volume_count</i>]	Sets the maximum number of volumes in future multivolume tape dataset transfers, or returns to the system defaults.			OS/390 and z/OS targets only.
vol [<i>ume</i>] [<i>volser</i>]	Specifies the serial number of the volume on which you want to dynamically allocate new target OS/390 or z/OS mainframe dataset.			New OS/390 and z/OS targets only.

- a. **Dependency:** Any prerequisites for a command are listed in this column. A prerequisite is the need for another command to be in a certain state before the command in question may be performed. If a dependency is not met, then the command is invalid, and it fails. For example, if the **imxlate** command is OFF, you receive an error message if you specify an **imtrail** command.
- b. **Side Effects:** If a command causes another action to be taken, it is listed in this column. Usually the other action is a change of state of another command. A side effect never causes the command to fail. For example, if **imxlate** is OFF and you enter an **imsendx AimEim.imxlate** command, IFT sets the **imxlate** command to ON. If the **imxlate** command is already ON, IFT makes no change to its state.
- c. **Host Availability:** Any restriction on platform use of a command is listed in this column. If only a host is listed, the command is valid in any use involving that host. If the command is only valid when data is moving to the host (the file is a target file), “target only” is listed. If the command is only valid when data is moving from the host (the file is a source), “source only” is listed. In some cases, the command is only valid when a new mainframe dataset is being created. These instances are marked by the word “new” in the Host Availability column.
- d. **Host Applicability:** Not applicable to OS/390 or z/OS servers if those servers have not been configured for character translation.

This chapter provides reference information about the CLI commands. Topics include:

◆ Introduction	4-3
◆ binary	4-4
◆ blksize (OS/390 and z/OS)	4-5
◆ cd	4-7
◆ close	4-11
◆ del	4-12
◆ dir	4-14
◆ get	4-23
◆ help [?]	4-30
◆ imaddlf (OS/390 and z/OS)	4-32
◆ imbufno (OS/390 and z/OS)	4-35
◆ imcrlf	4-37
◆ imcylinder (OS/390 and z/OS)	4-40
◆ imdirblocks (OS/390 and z/OS)	4-44
◆ imfilenumber (OS/390 and z/OS)	4-46
◆ imlog (Open Systems)	4-48
◆ imrdw (OS/390 and z/OS)	4-50
◆ imrecvx	4-52
◆ imrelease (OS/390 and z/OS)	4-54
◆ imsendx	4-55
◆ imtrail (OS/390 and z/OS)	4-57
◆ imwrap (Open Systems)	4-59
◆ imxlate	4-61
◆ lcd	4-64
◆ ldir	4-68
◆ likedsn (OS/390 and z/OS)	4-75

◆ lpwd	4-78
◆ lrecl (OS/390 and z/OS)	4-79
◆ lsp (OS/390 and z/OS)	4-80
◆ mget	4-81
◆ mput	4-84
◆ open	4-87
◆ prompt	4-89
◆ put	4-90
◆ pwd	4-97
◆ quit	4-98
◆ recfm (OS/390 and z/OS)	4-99
◆ replace (OS/390 and z/OS)	4-101
◆ status	4-103
◆ unit (OS/390 and z/OS)	4-107
◆ unitcount (OS/390 and z/OS)	4-108
◆ upper	4-110
◆ verbose	4-112
◆ volcount (OS/390 and z/OS)	4-113
◆ volume (OS/390 and z/OS)	4-115

Introduction

The following sections describe the CLI commands. Refer Table 3-1 on page 3-11, and the note that follows it, for a definition of host A and host B. Each command reference section is divided into the following subsections:

- ◆ **Purpose** — Explains the purpose of the command
- ◆ **Syntax** — Shows the command
- ◆ **Arguments** — Shows any required or optional arguments
- ◆ **Usage Notes** — Lists detailed information about the command and its usage
- ◆ **Example** (optional) — Shows how to use the command
- ◆ **Related Topics** — Lists any CLI commands or other topics that may be related to this CLI command

binary

Purpose Use **binary** to specify that subsequent transfers will be in binary mode, byte-by-byte, without translation. To reenable translation, issue **imxlate ON** (or **imxlate** without an argument).

Syntax **bin**[ary]

Argument None

Usage Notes Keep the following points in mind:

Alternate Ways of Enabling Binary Mode Translation

You can use either **binary** or **imxlate OFF** to enable transfer data without translation. Binary mode translation allows for efficient storage and retrieval of datasets or files, and for the transfer of binary data.

Displaying Current Settings

Use **status** to display the current setting for translation.

Example The following example shows the dialog when you issue **binary** interactively with verbose mode enabled:

```
binary  
Command: binary  
Data translation is now off.
```

Related Topics

- ◆ **imrecvx**: page 4-52
- ◆ **imsendx**: page 4-55
- ◆ **imxlate**: page 4-61
- ◆ **status**: page 4-103
- ◆ **verbose**: page 4-112

blksize (OS/390 and z/OS)

Purpose Use **blksize** to specify the block size (in bytes) for any future target dataset allocations on mainframe hosts. The block size value is used for DCB blksize JCL parameter.

Syntax **blksize** [*<bytes>*]

Argument *<bytes>*

The maximum length of a block in bytes to reserve for a target dataset. The value can be 1 through 32760.

Usage Notes Keep the following points in mind:

Using the blksize Command with an Argument

If you have set the default_blksize parameter in your configuration file, you can use **blksize** with an argument to override the default_blksize value for future target dataset allocations.

Using the blksize Command Without an Argument

If you have set the default_blksize parameter in your configuration file, you can use **blksize** without an argument to reset the default_blksize value as the block size to use for future dataset allocations.

Using Neither the blksize Command nor default_blksize

If you do not set default_blksize configuration parameter or issue the **blksize** command, OS/390 or z/OS calculates the block size for any target dataset allocations.

Displaying Current Settings

Use **status** to display the current maximum length of a block in bytes.

Example The following example changes the maximum number of bytes in a block to the maximum permitted value of 32760:

```
blksize 32760  
Command: blksize 32760
```

Related Topics

- ◆ **imbufno (OS/390 and z/OS):** page 4-35
- ◆ **imcylinder (OS/390 and z/OS):** page 4-40
- ◆ **imdirblocks (OS/390 and z/OS):** page 4-44
- ◆ **imfilenumber (OS/390 and z/OS):** page 4-46
- ◆ **imrdw (OS/390 and z/OS):** page 4-50
- ◆ **lrecl (OS/390 and z/OS):** page 4-79
- ◆ **recfm (OS/390 and z/OS):** page 4-99
- ◆ **status:** page 4-103
- ◆ **unit (OS/390 and z/OS):** page 4-107

cd

Purpose Use **cd** to change the working directory or file group on host B when the client is controlling the transfer through host A.

Syntax **cd** [*<name>*]

Argument *<name>*
The name of a file directory. The maximum length is 150 characters.

Usage Notes Keep the following points in mind:

Using the Default Directory

To go to the default IFT directory, issue **cd** without a name argument.

Argument Values

The value for *<name>* when host B is an OS/390 or z/OS host must be a fully qualified dataset or a prefix. The value for *<name>* on a UNIX or Windows system must be a valid path.

Using the cd Command Without an Argument

If you do not specify *<name>*, IFT changes the working file directory or file group to that specified in the IFT configuration file on host B.

Verifying the Argument

When you issue **cd** with a directory or file group name as an argument, IFT checks to see whether the name is valid. If the name is invalid, IFT does not change the working directory or file group.

If the directory name is valid, but does not include a path in the syntax of the target operating system, IFT appends that directory to the current working directory or file group.

If the directory name includes a path in the syntax of the target operating system as an argument, the directory specified by that path becomes the current working directory or file group.

Mainframe Qualifiers and Dataset Names

Although **cd** is foreign to a mainframe host, IFT treats the mainframe hierarchy of qualifiers and dataset names as though they were part of an open systems directory structure.

Syntax in a Mainframe Batch Environment

In the mainframe batch environment, you can continue **cd** commands with SYSIN DD statements across multiple lines.

More Information: Refer to *Continuation in SYSIN Control Statements* on page B-4.

Directory and Filenames in a Windows Environment

In a Windows environment, you can use directory names with embedded spaces if you enclose the directory name with double quotes. For example:

```
cd "c:\Program Files"
```

Mainframe Examples

1. If the current working directory is TCPUSR14.TEST, the following **cd** command makes the new working directory:

```
TCPUSR14.TEST.FILES:
```

```
cd FILES
```

To override the existing directory, rather than append to the directory, specify the directory argument with single quotation marks.

2. If a partitioned data set (PDS) exists with the same name as the current working directory, IFT considers the working directory to be that PDS. Otherwise, IFT considers the working directory to be a common prefix qualifier for sequential datasets.

If a PDS exists with the same name as the current working directory, but you want the current working directory to be treated as a common prefix for sequential datasets, specify the working directory with a period (.) at the end of the name.

For example, if a PDS named TCPUSR14.TEST exists, the following **cd** command makes the PDS TCPUSR14.TEST the current working directory:

```
cd 'TCPUSR14.TEST'
```

Note: Single quotes are supported on mainframe hosts only.

A subsequent use of the **put** command with NAME1 adds a member NAME1 to the TCPUSR14.TEST PDS.

- In contrast, the following **cd** command makes `TCPUSR14.TEST.` the current working directory, a prefix for sequential datasets:

```
cd 'TCPUSR14.TEST.'
```

A subsequent **put** command, used to transfer dataset `NAME1`, would create the `TCPUSR14.TEST.NAME1` sequential dataset.

UNIX Examples

- If the current working directory is `/tcpusr14/test`, the following **cd** command makes the new working directory `/tcpusr14/test/files`:

```
cd files
```

- When you specify a path as the argument, the **cd** command makes the directory specified by that path the working directory. If the current working directory is `/tcpusr14/test`, the following **cd** command makes the new working directory `/ftp/files`:

```
cd /ftp/files
```

Windows Examples

- If the current working directory is `\tcpusr14\test`, the following **cd** command makes `\tcpusr14\test\files` the new working directory:

```
cd files
```

- When you specify a path as the argument, the **cd** command makes the directory specified by that path the working directory. If the current working directory is `\tcpusr14\test`, the following **cd** command makes `\ftp\files` the new working directory:

```
cd \ftp\files
```

- If the current working directory is `\tcpusr14\test` on drive `C:`, the following **cd** command makes `e:\ftp\files` the current working directory:

```
cd e:\ftp\files
```

- If the directory name has embedded spaces in it, you must enclose the entire directory name in double quotes:

```
cd c:\  
cd "c:\Program Files\ftp\files"
```

5. To go to the root level of a drive, enter the `cd` command with the drive letter. For example:

```
cd D:\
```

Related Topics

- ◆ `dir`: page 4-14
- ◆ `lcd`: page 4-64
- ◆ `ldir`: page 4-68
- ◆ `lpwd`: page 4-78
- ◆ `pwd`: page 4-97

close

Purpose Use **close** to end the connection between the local and remote IFT servers without disconnecting the client from the local server.

Syntax **close**

Argument None

Usage Notes Keep the following points in mind:

Command Behavior

The **close** command ends the connection between the IFT server on host A and the IFT server on host B. It also prints an informational message that includes the date and time.

Where to Issue the close Command

Issue **close** to shut down an IFT connection from the client session that opened the connection.

Example When you issue **close**, IFT responds with the name of the disconnected IFT server on host B. The following example is a **close** command from a UNIX host:

close

Command: close

```
Closing connection to remote InfoMover File Transfer server labmec(11/26/2004  
17:20:08)
```

Related Topics

- ◆ **open**: page 4-87
- ◆ **quit**: page 4-98
- ◆ **verbose**: page 4-112

del

Purpose Use **del** to delete a specified file or dataset on the remote server's system.

Syntax **del** [ete] | **rm** | **remove** <filename>

Argument <filename>

The name of the file or data set to be deleted. The maximum length you can use (including any path specification) is 150 characters.

Usage Notes Keep the following points in mind:

del Synonyms

You can give the **del** command as the full word, **delete**, or the abbreviation, **del**. You can use **remove** or **rm** as synonyms for the **del** command.

Directories

You cannot use **del** to delete a directory. If you try to delete a directory, you receive an IM072E: Error: Delete request failed message.

File Permissions

You cannot delete a file when your login account on the remote system does not have permission to delete the file.

Locked Files

You cannot delete a file on a remote system when that file has been locked by another user.

Disabled del Commands

There may be some hosts from which you cannot delete files with the **del** command. By setting configuration parameters system managers or system programmers can disable **del** and **put** commands for selected users or selected hosts. Consult your system administrator or system programmer for more information.

Miscellaneous

IFT does not support the deletion of tapes, VSAM files, PDS or PDSE data sets, or generation data group (GDG) base names.

IFT does support the deletion of the following file types: physical sequential files, individual generations of GDGs by absolute or relative name, or members of PDS and PDSE datasets.

Examples None.

Related Topics

- ◆ **cd**: page 4-7
- ◆ **dir**: page 4-14
- ◆ **lcd**: page 4-64
- ◆ **ldir**: page 4-68
- ◆ **put**: page 4-90

dir

Purpose Use **dir** to return a list of directory entries or a list of files in a file group on host B when your client is controlling the transfer through host A. The **dir** command also gives additional information about the files.

There are two variants of **dir**:

- ◆ Direct **dir**
- ◆ Redirected **dir**

Direct dir

Direct **dir** returns a list of the specified files and auxiliary information about the files to your current output device. If host B is a mainframe system, **dir** returns a list of the members of a partitioned dataset as well as auxiliary information about the members.

Redirected dir

Redirected **dir** is supported only when host B is a mainframe host. Redirected **dir** allows you to redirect the returned list to a specified mainframe output dataset instead of displaying the results on your window. There is no limit on the number of datasets the redirected **dir** can return, provided that the output dataset has sufficient allocated space to hold the directory listing.

Syntax **dir** [*<name>*]

dir [*<name>*] >*<dataset>* (OS/390 and z/OS only)

Arguments *<name>*

The (optional) directory or file group. The default is the current directory or file group. The maximum number of characters you can specify in *<name>* is 150. The name must be fully qualified.

>

The redirection symbol. If you do not specify a file group (that is, if you use the current file group), leave one space between **dir** and the redirection symbol (**dir** >). Follow the redirection symbol with the output dataset name, *<dataset>*. Do not leave any embedded spaces between the redirection symbol and *<dataset>*.

If you do specify a file group, make sure you do not embed spaces between *<name>* and the redirection symbol, and between the redirection symbol and *<dataset>*.

<dataset>

The name of the output dataset. The name must be a fully qualified dataset name, specified without quotes.

The output dataset can be either sequential or an existing GDG member, and must have been allocated before you issue the redirected **dir** command. If you specify a member of a PDS as the output dataset, that member does not have to exist. The output dataset must have an LRECL of 80 bytes or greater (for FB).

Mainframe Direct dir Usage Notes

Keep the following points in mind when you use **dir** on mainframe datasets:

Syntax in a Mainframe Batch Environment

You can continue **dir** commands that are specified with SYSIN DD statements across multiple lines. When placed at the end of a line, the minus sign (dash) character indicates that the statement is continued on the following line.

More Information: Refer to *Continuation in SYSIN Control Statements* on page B-4.

Alternate Command Word

You can use **ls** as a synonym for the **dir** command. For example, the following commands have the same result:

```
dir 'patrec'  
ls 'patrec'
```

Maximum Number of Returned Datasets

There is no limit to the number of mainframe datasets that can be returned to the client. However, because the list of datasets being sent from the server to the client is in buffers of 32767 bytes, performance may be slow with long lists.

Behavior of dir Without an Argument

If the current working directory is not a PDS, **dir** without an argument provides a list of all datasets starting with an index equal to the current working directory.

More Information: Refer to *dir and Partitioned Datasets* on page 4-17 for more information about the behavior of **dir** without an argument when directed at partitioned datasets.

Mainframe Wildcarding

Using **dir**, you can perform catalog searches using DFSMS wildcarding. Table 4-1 shows the valid wildcards when host B is an IBM mainframe:

Table 4-1 Mainframe dir Wildcard Characters

Character	Meaning
*	Either a qualifier or one or more characters within a qualifier. An asterisk can precede or follow any literal character, including a period (.) or a blank ().
**	Zero or more qualifiers. A double asterisk cannot precede or follow any characters but a period (.) or a blank ().
%	One alphanumeric or national character. You can specify up to eight percent characters in each qualifier.

With mainframe hosts, IFT follows the wildcarding conventions used by the IGGCS100 program.

More Information: Refer to the IBM publication, *DFSMS Managing Catalogs*, for more details about wildcarding conventions.

dir and GDGs

You can use **dir** with a wildcard as part of the argument to display GDG basenames and provide an indicator of the datasets that are part of the GDG basename. For example:

```
dir 'asgform.gdg.test.**'
      gdg . . . ASGFORM.GDG.TEST
APP003 gdg vb 27998 4092 0 ASGFORM.GDG.TEST.G0007V00
APP000 gdg vb 27998 4092 0 ASGFORM.GDG.TEST.G0008V00
APP003 ps vb 27998 4092 0 ASGFORM.GDG.TEST.G0004V00
APP001 ps vb 27998 4092 0 ASGFORM.GDG.TEST.G0005V00
APP007 ps vb 27998 4092 0 ASGFORM.GDG.TEST.G0006V00
```

To list the associations of the GDG, use the form (*) as a wildcard:

```
dir 'asgform.gdg.test(*)'
APP003 gdg vb 27998 4092 0 DMOSHIE.GDG.TEST.G0007V00
APP000 gdg vb 27998 4092 0 DMOSHIE.GDG.TEST.G0008V00
```

dir and Partitioned Datasets

The **dir** command follows the same rules as IBM FTP. For example, **dir** without an argument provides a list of the members if the current local directory is a PDS:

```
cd 'invent.pds'
```

```
dir
```

```
A
```

```
B
```

```
B
```

where:

A, B, and C are the partitioned dataset members.

If you specify the fully qualified PDS, you receive only the PDS name:

```
dir 'invent.pds'
```

```
INVENT.PDS
```

dir and Cataloged Tape Datasets

You cannot use **dir** to obtain any information about cataloged tape datasets.

Open Systems Direct dir Usage Notes

Keep the following in mind when you use **dir** with open systems files:

Alternate Command Word

You can use **ls** as a synonym for the **dir** command. For example, the following commands have the same result:

```
dir /usr/imqa
```

```
ls /usr/imqa
```

Open Systems Wildcarding

Table 4-2 shows the wildcard characters that are valid when host B is an open system:

Table 4-2 Open Systems dir Wildcard Characters

Wildcard Character	Meaning
* (asterisk)	Zero or more alphanumeric characters in a string of characters. An asterisk can precede or follow any literal character. For example, <code>dat*</code> , returns <code>dat</code> , <code>date</code> , <code>data</code> , <code>daten1</code> and any other string that starts with <code>dat</code> and has any other terminating characters.
? (question mark)	One alphanumeric character in a string of characters. You can specify one to eight question marks in each qualifier. For example, <code>dat?</code> returns <code>date</code> or <code>data</code> . It does not return <code>dat</code> or any string that starts with <code>dat</code> and has more than one additional suffix characters.

Windows Command Behavior

The `dir` command on a Windows host shows hidden files as well as files with other attributes.

Directory Names in Windows Systems

In a Windows environment, you can use directory names with embedded spaces if you enclose the directory name in double quotes. For example:

```
dir "Warehouse Files"
```

The dir Command Memory Usage

The `dir` command uses up memory. Multiple simultaneous invocations of `dir` can slow down your system significantly. If you have a memory slow down, close the server and restart it to clear the memory.

Redirected dir Usage Notes

Keep the following points in mind:

Redirected dir Command Argument

The redirected form of `dir` accepts one argument. That is, you can only specify one dataset for redirected output. To redirect output to multiple datasets, you must issue a separate redirected `dir` commands for each dataset.

Configuration Parameters

Because of the long processing delay that accompanies a redirected **dir**, you may have to increase the `client_timeout` configuration parameter, and `command_timeout` configuration parameter values in the configuration file on the host.¹

Diagnostic Messages

You can find diagnostic messages for the status of redirected **dir** output in the mainframe host connector log. These messages have a debug level of 4.

PDS Member Lists

The redirected **dir** command cannot send a partitioned data set (PDS) member list to the specified output dataset.

Direct dir Command Mainframe Examples

1. The following example uses the double asterisk wildcard to list all datasets beginning with INVENT:

```
dir 'INVENT.**'
APP006 ps vb 3120      80  0      INVENT.A-A
VISA06 ps vb 27998   4092 0      INVENT.BB.BB
APP006 po fb 27920    80  0      INVENT.BMC.JCL
APP007 po fb 27920    80  0      INVENT.BMCDR.JCL
APP011 po vb 27998   4092 0      INVENT.B14.PDS
APP009 po fb 5600     80  0      INVENT.CBTTAPE
APP002 ps fb 3120     80  0      INVENT.CBT492.XMI
APP000 po fb 6160     80  0      INVENT.CLIST
APP005 po fb 3120     80  0      INVENT.CNTL
APP002 ps fb 27920    80  0      INVENT.COMMENTS
APP010 ps fb 27920    80  0      INVENT.COMMENTS.BKUP
APP004 ps vb 27998    240 0      INVENT.COUNT
APP010 ps vb 27998   4092 0      INVENT.DANTEST
APP010 ps vb 27998   4092 0      INVENT.DANTEST1
APP005 po fb 27920    80  0      INVENT.DB2.COBOLE
APP003 po fb 27920    80  0      INVENT.DB2.JCL
APP005 po fb 27920    80  0      INVENT.DB2.PLI
APP002 po fb 27920    80  0      INVENT.DB2.REXX
APP006 po fb 27900    100 0      INVENT.DB2OLD.JCL
APP001 po fb 3120     80  0      INVENT.DB2UTIL.JCL
.
.
.
```

1. You can only change these configuration parameters by editing the IFT configuration file. Refer to the *InfoMover File Transfer Installation Guide* for more information about configuration parameters.

2. The following examples show how to use wildcarding with **dir** in a mainframe situation:

- `dir 'PATREC.*.SET` returns the following:

```
PATREC.DATA1.SET  
PATREC.DATA2.SET
```

but, does not return `PATREC.DATA.SET.KSDS`

- `dir 'PATREC.*A'` returns the following:

```
PATREC.A  
PATREC.BA  
PATREC.BBA
```

but, does not return:

```
PATREC.B  
PATREC.AB
```

- `dir 'PATREC.**'` returns the following:

```
PATREC  
PATREC.DATA.SET1  
PATREC.DATA.SET2
```

but, does not return `PATREC1.DATA.SET`

- `dir 'patrec.data.set%'` returns the following:

```
PATREC.DATA.SET1  
PATREC.DATA.SET2
```

but, does not return `PATREC.DATA.SET30`

- `dir 'patrec.data.set%%'` returns the following:

```
PATREC.DATA.SET30  
PATREC.DATA.SET40
```

but, does not return:

```
PATREC.DATA.SET3  
PATREC.DATA.SET4
```

- You can list the datasets that start with a specific “index.”

For example, if the current working directory were set to the index level DBUTCHA.IM in a mainframe host environment, the **dir** command may display the following files:

```

dir
TS0003  PS FB 3120  80 0  DBUTCHA.IM.CHUNK1
TS0002  PO FB 2400  80 0  DBUTCHA.IM.JCL
  
```

Diagram labels for the above output:

- Volume Serial Number (Volser): TS0003, TS0002
- Data Set Organization: PS, PO
- Record Format: FB, FB
- Block Size: 3120, 2400
- Logical Record Length: 80, 80
- Key Length: 0, 0
- Data Set Names: DBUTCHA.IM.CHUNK1, DBUTCHA.IM.JCL

Direct dir Command UNIX Example You can use the **dir** command with a directory name to display a list of files. Look at the following example:

```

dir /usr/imga
total 150
-rw-r--r-- 1 root other 461 Sep 16 09:45 SOL.1.17
-rw-r--r-- 1 root other 0 Sep 12 17:01 encfmt.lock
drwxr-xr-x 2 root other 512 Sep 16 17:30 foo.bar
-rwxr-xr-x 1 root other 17922 Sep 18 11:58 text.dat
-rwxr-xr-x 1 root other 13103 Sep 18 11:58 xlab.tst
-rw-r--r-- 1 root other 318 Sep 16 16:00 xlab1.txt
-rwxr-xr-x 1 root other 41159 Sep 18 11:58 fillup.1
  
```

Diagram labels for the above output:

- Permissions: -rw-r--r--
- Link Count: 1
- Owner: root
- Group: other
- Size in Bytes: 461, 0, 512, 17922, 13103, 318, 41159
- Date and Time Last Modified: Sep 16 09:45, Sep 12 17:01, Sep 16 17:30, Sep 18 11:58, Sep 18 11:58, Sep 16 16:00, Sep 18 11:58
- File Name: SOL.1.17, encfmt.lock, foo.bar, text.dat, xlab.tst, xlab1.txt, fillup.1

Direct dir Command Windows Example

You can use **dir** with the name of a directory to display a list of files. Look at the following example:

```
dir C:\medrec\smith\2000
Volume in drive C has no label.
Volume Serial Number is 311F-0801

Directory of C:\medrec\smith\2000

04-22-2000  8:21a    <DIR>      tests
05-14-2000  3:31p    <DIR>      billing
05-14-2000  3:34p    34466     history.txt
```

Date and Time Last Modified File Size in Bytes File or Directory Name

Redirected dir Command Mainframe Examples

1. The following example redirects a directory of USER1 datasets to AS26.DIROUT:

```
cd USER1
dir >AD26.DIROUT
```

2. The following example redirects a **dir** command of USER1 datasets to the existing PDS, AD26.PDS.DIROUT(B):

```
dir USER1>AD26.PDS.DIROUT(B)
```

Note: Member B does not have to exist prior to the redirected **dir** command.

Related Topics

- ◆ **cd:** page 4-7
- ◆ **lcd:** page 4-64
- ◆ **ldir:** page 4-68
- ◆ **lpwd:** page 4-78
- ◆ **pwd:** page 4-97

get

Purpose Use **get** to transfer a file or dataset from host B to host A when your client is controlling the transfer through host A.

Syntax **get** *<source>* [*<target>*]

Arguments *<source>*

The name of the file or dataset to be transferred from host B. The maximum number of characters you can use (including any path specification) is 150.

<target>

The optional name of the file or dataset on host A to which the source is transferred. The maximum number of characters you can use (including any path specification) is 150.

Usage Notes Keep the following points in mind:

Server Connections

The IFT client must be connected to the remote server through which the specified file is to be retrieved.

Valid Client Server Connections

In many cases, host A is a local host and host B is a remote host. However, if you are using an open systems based client, you may be using a thin client on a third host. In that case, both hosts are remote.

More Information: Refer to Table 3-1 on page 3-11 for a list of the valid client-server connections you can make.

Command Failure

In the event that **get** fails because *<source>* does not exist (or there is some other error condition in *<source>*) and *<target>* is a new allocation of an OS/390 or z/OS dataset (or a new GDG generation), IFT deletes *<target>*.

Source Specifications on Mainframe Systems

If host B is a mainframe system, you must fully qualify *<source>* by enclosing *<source>* in single quotes, if *<source>* is not under the current high-level qualifier. If you do not fully qualify *<source>*, IFT looks for it under the current high-level qualifier.

Source Specifications on Open Systems

If *<source>* is in the current working directory on host B, you do not need to specify the directory path. If *<source>* is not in the current working directory on host B, you must specify the directory path as part of *<source>*.

If *<source>* is on a Windows system, you can specify *<source>* with either:

- ◆ Windows syntax for the pathname, for example:
`d:\temp\file_name`
- ◆ The Universal Naming Conventions (UNC) to specify *<source>* or *<target>*, for example:
`//machine_name/share_name/directory_name/file_name`

Target Specifications on Mainframe Systems

If host A is a mainframe host system and you specify *<target>* without single quotes, IFT creates a dataset named *<target>* in the default high-level qualifier specified during installation or configuration on host A.

If you do not specify *<target>*, IFT extracts the file and/or dataset name from *<source>*, and creates a dataset with that name in the default high-level qualifier specified during installation or configuration on host A.

If you specify *<source>* with single quotes, IFT creates *<target>* as specified.

Target Specifications on Open Systems

If you specify *<target>* without a directory path in a transfer to an open system, IFT creates a file named *<target>* in the default directory specified during IFT installation and/or configuration.

If you do not specify *<target>* in a transfer to an open system, IFT extracts the filename from *<source>*, and creates a *<target>* of that name in the default directory specified during IFT installation and/or configuration.

If *<target>* is on a Windows system, you can specify *<target>* with either:

- ◆ Windows syntax for the pathname, for example:
`d:\temp\file_name`
- ◆ The UNC to specify the *<source>* or *<target>*, for example:
`//machine_name/share_name/directory_name/file_name`

Invalid Target Name

If the name specified for *<target>* is not valid on host A, *<source>* is not transferred.

Read Privileges on Sources

To get a file or dataset from host B, the server on host B must have read privileges to the file.

Tape Sources and Targets

IFT supports the use of datasets on magnetic tape devices as sources or targets on mainframe hosts. IFT does not support the use of files on magnetic tape devices as sources or targets on UNIX or Windows hosts.

Pipe Sources and Targets

IFT supports the use of named pipes as sources or targets on UNIX hosts. IFT does not support the use of batch pipes as sources or targets on a mainframe host system, or named pipes as sources or targets on Windows hosts.

Using Named Pipes on UNIX Systems

When you use a named pipe as a source or target on a UNIX system, the other participant in the transfer can be located on any host platform.

To use a pipe as a UNIX source or target, you must:

- a. Create the pipe outside IFT at some point before the transfer begins.
- b. Start the process to the pipe either before the transfer is started or within two minutes after the transfer is started.

If the process is not started to the pipe within two minutes after the transfer, you receive the following error:

```
IM003E: Error: Failed to open file
```

To use a UNIX pipe as a target, you must have previously issued a **replace ON** command to enable replacement mode.

Empty Datasets and Files as Sources

A **get** command, performed on an empty dataset or file on host B, erases the contents of an existing local dataset with the same name on host A.

Mainframe Target Dataset Default Attributes

In the absence of any other allocation specifications (through setting optional configuration values or using CLI commands), IFT creates mainframe target datasets with the following attributes:

- ◆ Variable block record format
- ◆ A 4092-record length
- ◆ A 2-cylinder primary extent
- ◆ A 5-cylinder secondary extent
- ◆ A default unit of SYSDA

The target dataset is given the default block size of the system on which it is being created.

More Information: Refer to *Specifying Allocation Attributes* on page B-13 for a description of target dataset default attributes.

Target Permissions

Target permissions are as follows:

- ◆ In a mainframe host environment, the target dataset has only the permissions imposed upon it by the security management system. See your site security administrator for special requirements.
- ◆ On UNIX systems, all newly-created UNIX target files have 644 permission set for them.
- ◆ In a Windows environment, the target file permission settings are those of the directory into which you place *<target>*.

Transfer Behavior in Replace Mode

If the target dataset or file on host A already exists and you previously specified replace mode (that is, issued **replace ON**), the data are overwritten. The source dataset or file on host B retains its existing characteristics.

GDGs as Sources and Targets

IFT accepts relative and absolute generation data group, dataset names as *<source>* and *<target>* specifications in **get** commands. You cannot transfer an entire GDG, only an individual generation dataset.

More Information: Refer to *IFT Server* on page 1-3 for an explanation of GDGs as sources and targets.

Syntax in a Mainframe Batch Environment

In the mainframe batch environment, you can continue **get** commands you specify in SYSIN DD statements across multiple lines.

More Information: Refer to *Continuation in SYSIN Control Statements* on page B-4.

Transfer Verification Messages

After the file is transferred, IFT displays:

- ◆ Results of the transfer
- ◆ Source and target names
- ◆ Completion time
- ◆ Elapsed time in seconds
- ◆ File sizes in bytes
- ◆ Transfer speed in kilobytes per second

Displayed Transfer Times

The transfer time displayed with **get** commands is expressed in whole seconds rather than fractions of a second. Thus, some transfers, especially those of small files, appear to take longer than they actually do. For instance, a **get** command in verbose mode would report that a transfer that took 1.2 seconds as taking 2 seconds.

Failed Transfers to a Mainframe Target Host

If *<target>* already exists on a mainframe system and a re-transfer of *<source>* to *<target>* fails, IFT truncates *<target>* on the host to a length of zero bytes, but does not delete it. Therefore, you must delete *<target>*. If *<target>* does not exist on the mainframe and the transfer fails, IFT automatically deletes *<target>*.

Source and Target Filenames on Windows Systems

In a Windows host environment, you can use *<source>* and *<target>* filenames with embedded spaces, if you enclose the path name with double quotes. For example:

```
get usr/reports/new "D:\temp\Warehouse Files\New Data"
```

Disabled **get** Commands

There may be some hosts from which you cannot transfer files with the **get** command. By setting configuration parameters system managers or system programmers can disable **get** commands for selected users or selected hosts. Consult your system manager or system programmer for more information.

Examples

1. The following example shows a **get** command being used to transfer a dataset from the working directory on host B, a mainframe system, to the specified directory and target filename on host A, a UNIX system:

```
get LXR110 /usr/dtest/file1
```

IFT indicates the transfer type and source file name in its response:

```
Data will be translated using (Default)
Transferring file LXR110
IM00I Information: File transfer successfully completed.
Source = LXR110,
Target = /usr/dtest/file1
Time: (09/18/2004 13:42:55) (1)
Elapsed=29, FileSize=147465112, KbytePerSec=5085
```

2. If the source file is not in the current working directory on host B, you must specify the source directory as part of *<source>*.

For example, if host B is a mainframe system and the complete name of the source file is CCH.C113.C(CIM10), you can issue the **get** command as follows to override use of the current source working directory:

```
get 'CCH.C113.C(CIM10)' /usr/dtest/file1
```

IFT responds as follows:

```
Data will be translated using (Default)
Transferring file 'CCH.C113.C(CIM10)'
IM00I Information: File transfer successfully completed.
Source = 'CCH.C113.C(CIM10)'
Target = usr/dtest/file1
Time (09/18/2004 13:51:55) (1)
Elapsed=29, FileSize=147461112, KbytePerSec=5085
```

- a. If both host B and host A are UNIX systems, the sequence may be as follows:

```
get /usr/dtest/file2 /usr/dtest/file2copy
```

```
command: get /usr/dtest/file2 /usr/dtest/file2copy
```

```
Data will not be translated
Transferring file /usr/dtest/file2
IM000I Information: File transfer successfully completed
Source = /usr/dtest/file2,
Target = /usr/dtest/file2copy
Time: (09/18/2004 13:51:55) (1)
Elapsed=29, FileSize=147987076, KbytePerSec=5103
```

- b. If host B is a UNIX system and host A is a Windows system, the sequence may be as follows:

```
get /usr/dtest/file2 d:\temp\imData\file2copy
```

```
Command: get /usr/dtest/file2 d:\temp\imData\file2copy
```

```
Data will not be translated
```

```
Transferring file /usr/dtest/file2
```

```
IM000I Information: File transfer successfully completed.
```

```
Source File = '/usr/dtest/file2'
```

```
Target File = 'd:\temp\imData\file2copy'
```

```
Elapsed=29,FileSize=147987076,KbytePerSec=5103
```

Related Topics

- ◆ **mget**: page 4-81
- ◆ **mput**: page 4-84
- ◆ **put**: page 4-90
- ◆ **replace (OS/390 and z/OS)**: page 4-101
- ◆ **status**: page 4-103
- ◆ **verbose**: page 4-112

help (?)

Purpose Use **help** to display help information about the CLI commands.

Syntax **help** | **?** [*<command>* | *]

Arguments *<command>*

Displays help information for the specified command.

Displays help information for all commands.

Usage Notes If you issue **help** without an argument, it displays the list of valid commands. The commands you see depend on the following:

- ◆ The state of the client
- ◆ Whether the client is connected to a server
- ◆ The type of host on which the server resides

Examples The following example shows the dialog when you issue **help** from a mainframe client connected to an OS/390 or z/OS system. An asterisk (*) preceding a command name indicates that the command is available only when the local or remote host is a mainframe. Alternate forms of a command appear in parentheses:

help

Command: help

Valid commands are:

```

cd          lcd connect close
get         imcrLf open put
pwd         lpwd prompt
binary     (bin)
dir         (ls)
ldir       (lls)
delete     (del, remove, rm)
help       (?)
imlog      (iml)
imrdw      (imrdw_transfer)
imrecvx    (imrecvxlate)
imsendx    (imsendxlate)
imtrail    (imtrailblanks)
imxlate    (imx)
quit       (qu)
replace    (rep)

```

```
imwrap
status      (stat)
upper       (up)
verbose     (verb)
mput
mget
*imrelease
*imcyl      (imcylinder, cyl, cylinder, cylinders)
*imaddlf    (imlf, imlfs)
*lsp
*volume     (vol)
*likedsn
*unit
*blksize
*imbufno
*imdirblocks(imdirblks)
*lrecl
*recfm
*imfilenumber (imfileno)
*volcount    (volcnt)
```

NOTE: A '*' preceding a command indicates that the command applies to OS/390 or z/OS systems only.

The following example shows a **Help** listing for a specific command, in this case, the **quit** command:

```
IM> help quit
Command Name: quit

quit = Terminate InfoMover File Transfer session/connection and exit client

IM>
```

Related Topics None

imaddlf (OS/390 and z/OS)

Purpose Use **imaddlf** to enable or disable the appending of record linefeeds to OS/390 and z/OS sources or targets in future transfers. When enabled, **imaddlf** adds linefeeds to the end of each record that IFT transfers from a mainframe source file, or to a mainframe target file.

Syntax **imaddlf** | **imlf** | **imlfs** [**ON** | **OFF**]

Arguments **ON**

Enables appending linefeeds to each record.

OFF

Disables appending linefeeds to each record.

Usage Notes Keep the following in mind:

Command Synonyms

You can use the forms **imlf** and **imlfs** as synonyms for **imaddlf**.

Using imaddlf Without an Argument

You can use **imaddlf** without an argument. The **imaddlf** command without an argument toggles linefeed appending on and off.

If your server connection involves a mainframe host and an open systems host, linefeed appending is enabled by default. Thus, when you first issue **imaddlf** without an argument, you disable linefeed appending on any subsequent transfers.

Each subsequent argumentless **imaddlf** that you issue reverses the state of linefeed appending.

Effects on Different Platforms

The effect of **imaddlf** depends on the server platforms involved in the transfers.

Mainframe to Mainframe

Text dataset on both source and target servers have the same format. Issuing **imaddlf** has no effect.

Mainframe to UNIX

When **imaddlf** conversion is enabled, the mainframe side adds linefeeds to the outgoing datasets. When **imaddlf** conversion is disabled, the mainframe side does not add linefeeds to outgoing text datasets.

Mainframe to Windows

When **imaddlf** conversion is enabled, the mainframe side adds linefeeds to outgoing text datasets. Depending on the state of the **imcrlf** command, the Windows side may add carriage returns to the text files on arrival.

When **imaddlf** conversion is disabled, the EOL linefeeds are not added to text datasets transferred from OS/390 or z/OS to Windows.

Windows to Mainframe

When **imaddlf** conversion is enabled, the Windows side removes linefeeds from text file records. When EOL conversion is disabled, the Windows side does not remove linefeed characters from the end of text file records.

UNIX to Mainframe

When **imaddlf** conversion is enabled, the UNIX side removes linefeed EOL characters from text file records. When EOL conversion is disabled, the UNIX side does not remove linefeed EOL characters from text file records.

Mainframe Datasets with ISO Characters

There are certain constraints you must keep in mind if you are transferring a mainframe file containing ISO control characters to an open systems host.

More Information: Refer to *Managing Files with ISO Control Characters* on page B-27.

Displaying Current Settings

Use **status** to determine the current setting of **imaddlf**.

Example None

- Related Topics**
- ◆ **imcrif**: page 4-37
 - ◆ **imtrail (OS/390 and z/OS)**: page 4-57
 - ◆ **imwrap (Open Systems)**: page 4-59
 - ◆ **upper**: page 4-110

imbufno (OS/390 and z/OS)

Purpose Use **imbufno** when an OS/390 or z/OS system is involved in subsequent transfers. The **imbufno** command controls the allocation of a mainframe dataset as if `JCL DCB=imbufno=size` were specified. The **imbufno** command allows you to:

- ◆ Override the value in the `default_imbufno` configuration parameter ¹ that specifies the number of buffers to be assigned to the DCB when allocating a source or target dataset on a mainframe system.
- ◆ Specify a new number of buffers to assign to the Data Control Block (DCB).

Using **imbufno** without an argument resets the number of buffers to be assigned to the DCB to the value in the configuration file. If you did not specify a value for `default_imbufno`, IFT uses the default of two (2).

Syntax **imbufno** [*<size>*]

Argument *<size>*

The number of buffers you want to assign to the DCB. The values you can use are integers from 1 through 255.

Usage Notes Keep the following in mind:

When to Use the imbufno Command

You should use **imbufno** to modify the `default_imbufno` configuration parameter to improve performance. Consult your EMC Customer Support Representative.

Displaying Current Settings

Use **status** to determine the current number of buffers IFT assigns to the DCB in future transfers.

1. The `default_imbufno` configuration parameter is an advanced parameter that you can only change by editing the IFT configuration file. Refer to the *InfoMover File Transfer System Administrator's Guide* for more information about configuration parameters.

Example None

Related Topics

- ◆ **blksize (OS/390 and z/OS):** page 4-5
- ◆ **imaddlf (OS/390 and z/OS):** page 4-32
- ◆ **imbufno (OS/390 and z/OS):** page 4-35
- ◆ **imcylinder (OS/390 and z/OS):** page 4-40
- ◆ **imdirblocks (OS/390 and z/OS):** page 4-44
- ◆ **imfilenumber (OS/390 and z/OS):** page 4-46
- ◆ **imrdw (OS/390 and z/OS):** page 4-50
- ◆ **imrelease (OS/390 and z/OS):** page 4-54
- ◆ **irecl (OS/390 and z/OS):** page 4-79
- ◆ **lsp (OS/390 and z/OS):** page 4-80
- ◆ **recfm (OS/390 and z/OS):** page 4-99
- ◆ **status:** page 4-103
- ◆ **verbose:** page 4-112
- ◆ **volcount (OS/390 and z/OS):** page 4-113

imcrLf

Purpose Use **imcrLf** to enable or disable automatic conversion of EOL (end of line) characters (record terminators) during subsequent transfers of text files.

Syntax **imcrLf** [**ON** | **OFF**]

Arguments **ON**

Enables EOL conversion mode.

OFF

Disables EOL conversion mode.

Usage Notes Keep the following in mind:

Valid Targets

The **imcrLf** command is appropriate when transferring text files or datasets. IFT does not perform record terminator conversion on binary transfers.

Using the imcrLf Command Without an Argument

You can use **imcrLf** without an argument. Without an argument, **imcrLf** toggles EOL character conversion on and off. Each subsequent **imcrLf** without an argument reverses the state of EOL character conversion.

Effects on Different Platforms

The effect of **imcrLf** depends on the server platforms involved in the transfers.

Mainframe to Mainframe

Text dataset on both source and target servers have the same format. Issuing **imcrLf** has no effect.

UNIX to UNIX

Text files on both source and target servers have the same EOL characters. Issuing **imcrLf** has no effect.

Windows to Windows

Text files on both source and target servers have the same EOL characters. Issuing **imcrLf** has no effect.

Mainframe to UNIX

Issuing **imcrLf** has no effect.

Mainframe to Windows

When EOL conversion is enabled, the Windows side adds carriage returns to the text files on arrival. When EOL conversion is disabled, IFT does not add the EOL carriage returns to text datasets transferred from an OS/390 or z/OS system to a Windows system.

UNIX to Windows

When EOL conversion is enabled, the Windows side appends a carriage return to the linefeed at the end of each text record. When EOL conversion is disabled, the UNIX side sends files with linefeeds as record terminators, but the Windows side does not add the carriage returns to the incoming data records.

Many Windows-based applications add carriage returns to the EOL when you open the file. Others, however, do not. To make sure that each record in a text file transferred from UNIX has a carriage return as well as a linefeed character, issue **imcrLf** to set EOL character conversion.

Windows to UNIX

When EOL conversion is enabled, the Windows side strips carriage returns from text files transferred from Windows to UNIX, leaving only linefeeds as record terminators. When EOL conversion is disabled, the Windows side does not strip out carriage returns from text files transferred to UNIX.

Because UNIX cannot use the carriage return, issue **imcrLf** before you transfer text files from Windows to OS/390 or z/OS systems to enable the EOL conversion. Windows text files then go to an OS/390 or z/OS system without the carriage return.

Windows to Mainframe

When EOL conversion is enabled, the Windows side removes carriage returns from text file records. When EOL conversion is disabled, the Windows side does not remove carriage return characters from the end of text file records.

Because OS/390 and z/OS systems cannot use the carriage return, issue **imcrLf** before you transfer text files from Windows to OS/390 and z/OS systems to enable the EOL conversion. Windows text files then go to an OS/390 or z/OS system without the carriage return.

UNIX to Mainframe

Issuing **imcrLf** has no effect.

Displaying Current Settings

Use **status** to display the current setting of the record terminator conversion switch.

Example None

Related Topics

- ◆ **imaddlf (OS/390 and z/OS):** page 4-32
- ◆ **imtrail (OS/390 and z/OS):** page 4-57
- ◆ **imwrap (Open Systems):** page 4-59
- ◆ **upper:** page 4-110

imcylinder (OS/390 and z/OS)

Purpose Use **imcylinder** when an OS/390 or z/OS system is the target host of a subsequent transfer. The **imcylinder** command allows you to:

- ◆ Override the current values for the primary and secondary cylinder allocations in the IFT configuration file (the `default_cylinders` and `default_sec_cylinders` configuration parameters).
- ◆ Specify a new default primary allocation and secondary allocation extent to reserve for future target datasets on a mainframe system.

Using **imcylinder** without an argument returns the number of cylinders to the values in the IFT configuration file. These are either the defaults that were supplied with the system, or the values you set when you performed optional configuration.

If unspecified in the configuration file, the default values supplied in the IFT configuration file with the install kit are:

- ◆ 2 for the primary cylinders
- ◆ 5 for the secondary cylinders

Syntax **imcyl** [*inder*] | **cylinder** | **cylinders** | **cyl** [*<primary>*] [*<secndry>*]

Arguments *<primary>*

The number of cylinders ¹ to allocate on a mainframe system for a target dataset's primary allocation extent. ²

<secndry>

The number of cylinders to allocate with each new secondary extent. If you do not specify a secondary value during optional configuration, IFT uses the configuration value for the default secondary cylinders, supplied with the IFT configuration file in the IFT install kit (the `default_sec_cylinders` configuration parameter).

-
1. A cylinder is about 0.8 MB.
 2. An extent is a continuous space on a disk.

Usage Notes Keep the following points in mind:

Using Alternate Command Names

IFT allows you to use **cylinder**, **cyl**, or **cylinders** as aliases for **imcylinder**.

Maximum Size

Because a mainframe dataset cannot be more than 16 extents (primary and secondary), the maximum dataset size that is created without specifying cylinder size is 77 cylinders.

More Information: Refer to the IBM documents *OS/390 JCL User's Guide* and *OS/390 JCL Reference* for information about space allocation in a mainframe environment.

Using with Local and Remote Hosts

You can use **imcylinder** for new dataset allocations on either the local or remote hosts.

Argument Values and Space Available

The values you use with **imcylinder** are subject to the amount of space available on a volume (specified through the **volume** command), or a unit (specified through the **unit** command). You cannot specify a primary size greater than the actual size of the volume.

If the target dataset is larger than the number of cylinders you specify with **imcylinder**, IFT completes the transfer, if the volume has enough space, and if the number of extents does not exceed 16.

If an OS/390 or z/OS system cannot find enough contiguous cylinders to allocate the number of cylinders you specify with **imcylinder**, OS/390 or z/OS software may allocate a smaller number of cylinders than you requested.

Interaction of **imcylinder**, **likedsn**, and the Configuration File

The following text describes the interaction of the **likedsn** and **imcylinder** command, and the default cylinder value in the IFT configuration file (the `default_cylinders` configuration parameter) in setting the space attributes of new target datasets.

- ◆ If you issue both **likedsn** and **imcylinder**, IFT uses the space attributes of **imcylinder** for the target dataset rather than the space attributes of **likedsn**.

- ◆ If you issue **likedsn** and issue **imcylinder** without an argument (or if you did not issue **imcylinder**), IFT uses the space attributes of **likedsn** for the target dataset.
- ◆ If you do not issue either **likedsn** or **imcylinder**, IFT uses the default primary cylinder value in the configuration file (in the `default_cylinders` configuration parameter) for the target dataset.

More Information: Refer to *likedsn (OS/390 and z/OS)* on page 4-75.

Effect of the **imrelease** Command

If you issue **imrelease ON**, IFT releases any unused space when the transfer is completed successfully.

Displaying Current Settings

Use **status** to display the current setting.

Examples

1. The following example sets the primary allocation to three cylinders, and the secondary allocation to four cylinders:

```
imcylinder 3 4
```

```
Command: cylinder 3 4
```

After receiving this command, IFT performs the following:

- Clears the allocation field specified previously
- Creates a primary extent of three cylinders
- Creates secondary extents of four cylinders

2. The following example sets the primary allocation to five cylinders:

```
imcylinder 5
```

```
Command: imcylinder 5
```

After receiving this command, IFT performs the following:

- Creates a primary extent of five cylinders
- Keeps the secondary extent the same as specified previously

Related Topics

- ◆ **blksize (OS/390 and z/OS):** page 4-5
- ◆ **imbufno (OS/390 and z/OS):** page 4-35
- ◆ **imdirblocks (OS/390 and z/OS):** page 4-44
- ◆ **imfilenumber (OS/390 and z/OS):** page 4-46
- ◆ **imrelease (OS/390 and z/OS):** page 4-54
- ◆ **likedsn (OS/390 and z/OS):** page 4-75
- ◆ **lrecl (OS/390 and z/OS):** page 4-79
- ◆ **recfm (OS/390 and z/OS):** page 4-99
- ◆ **volcount (OS/390 and z/OS):** page 4-113
- ◆ **volume (OS/390 and z/OS):** page 4-115

imdirblocks (OS/390 and z/OS)

Purpose Use **imdirblocks** when an OS/390 or z/OS system is the target host of a subsequent transfers. The **imdirblocks** command allows you to:

- ◆ Override the value in the configuration file for the default number of directory blocks to be allocated for the directory of future target partitioned datasets (PDS)
- ◆ Specify allocation of future target partitioned dataset/extended (PDSE)

Using **imdirblocks** without an argument resets the number of directory blocks to be reserved in future transfers to the value specified in the IFT configuration file (in the `default_imdirblocks` configuration parameter).

Syntax `imdirblocks | imdirblks | imdirbkls [<blocks> | "PDSE"]`

Argument `<blocks>`

The number of directory blocks to allocate for the directory of a PDS. The values you can use are one (1) through 16777215.

"PDSE"

Indicates that the allocation is for a partitioned dataset/extended.

Usage Notes Keep the following points in mind:

Valid Targets

The **imdirblocks** command is valid only for transfers involving a new target PDS dataset.

The **imdirblocks** Command Synonym

You can use **imdirblks** as a synonym for **imdirblocks**.

Displaying Current Settings

You can use **status** to display the current number of directory blocks to reserve for PDS directories.

Example None

Related Topics

- ◆ **blksize (OS/390 and z/OS):** page 4-5
- ◆ **imbufno (OS/390 and z/OS):** page 4-35
- ◆ **imcylinder (OS/390 and z/OS):** page 4-40
- ◆ **imdirblocks (OS/390 and z/OS):** page 4-44
- ◆ **imfilenumber (OS/390 and z/OS):** page 4-46
- ◆ **imrdw (OS/390 and z/OS):** page 4-50
- ◆ **imrelease (OS/390 and z/OS):** page 4-54
- ◆ **likedsn (OS/390 and z/OS):** page 4-75
- ◆ **lrecl (OS/390 and z/OS):** page 4-79
- ◆ **recfm (OS/390 and z/OS):** page 4-99
- ◆ **volcount (OS/390 and z/OS):** page 4-113
- ◆ **volume (OS/390 and z/OS):** page 4-115

imfilenumber (OS/390 and z/OS)

Purpose Use **imfilenumber** when a new target OS/390 or z/OS dataset resides on magnetic tape. The **imfilenumber** command allows you to:

- ◆ Override the value in the default_imfilenumber configuration parameter ¹ for future dataset allocations
- ◆ Specify a new file number for a future target dataset on a tape volume

Using **imfilenumber** with an argument is equivalent to LABEL=(n,SL) (where *n* = file number) for new target datasets.

Using **imfilenumber** without an argument resets the relative position of the dataset on a tape volume in future transfers to the value in the default_imfilenumber configuration parameter.

Syntax **imfilenumber** | **imfileno** [*<sequence>*]

Argument *<sequence>*

The relative position of the dataset on the tape volume. The values you can use are from zero (0) through 9999.

Usage Notes Keep the following points in mind:

The imfilenumber and volume Commands

If you use **imfilenumber** to set a file number greater than one, you must use a **volume** command to specify the volume of the new target dataset.

Displaying Current Settings

You can use **status** to display the current relative position IFT is using.

The imfilenumber Command Synonym

You can use **imfileno** as a synonym for **imfilenumber**.

Example None

-
1. The default_imfilenumber configuration parameter is an advanced parameter that you can only change by editing the IFT configuration file. Refer to the *InfoMover File Transfer System Administrator's Guide* for more information about configuration parameters.

Related Topics

- ◆ **blksize (OS/390 and z/OS):** page 4-5
- ◆ **imbufno (OS/390 and z/OS):** page 4-35
- ◆ **imcylinder (OS/390 and z/OS):** page 4-40
- ◆ **imdirblocks (OS/390 and z/OS):** page 4-44
- ◆ **imrelease (OS/390 and z/OS):** page 4-54
- ◆ **lrecl (OS/390 and z/OS):** page 4-79
- ◆ **recfm (OS/390 and z/OS):** page 4-99
- ◆ **unit (OS/390 and z/OS):** page 4-107
- ◆ **status:** page 4-103
- ◆ **verbose:** page 4-112
- ◆ **volcount (OS/390 and z/OS):** page 4-113
- ◆ **volume (OS/390 and z/OS):** page 4-115

imlog (Open Systems)

Purpose Use **imlog** to enable or disable the logging of your CLI commands and the system's responses to a *client log file*.

The client log file is distinct from the server event log file, specified in the `sessionLogFile` parameter in the IFT configuration file. Client log files keep track of CLI command dialog during a client session, and are stored in the path you specify in the **imlog** argument. The server event log files keep track of all IFT events, errors, and transfers, and are stored on each server on which `sessionLogFile` has been set.

Syntax `iml [og] [ON | <logfile> | OFF]`

Arguments **ON**

Enables the logging of subsequent IFT dialog. When you issue **imlog ON**, IFT prompts you to enter a path and filename to use for client dialog logging.

`<logfile>`

Enables the logging of subsequent IFT dialog to the file named.

OFF

Disables the logging of all subsequent dialog. When you issue **imlog ON**, IFT closes the current transaction log file, and adds an extension with the date (four-digit year) and time that the log file closed.

Usage Notes Keep the following points in mind:

Valid Hosts

The **imlog** command applies only to UNIX or Windows clients.

Using the imlog Command Without an Argument

You can use **imlog** without an argument to control the writing of window output to a transaction log file. The **imlog** command without an argument toggles dialog logging on and off.

On all hosts, dialog logging is disabled by default. The first time you issue **imlog** without an argument, IFT prompts for a log filename. After you enter a name, IFT enables transaction logging to the file you specify.

Each subsequent **imlog** you issue without an argument reverses the current state of transaction logging. Each time you toggle dialog logging on with an argumentless **imlog**, you supply a log filename.

The log files and Verbose Mode

The dialog logged to the log file is the same as the dialog you see on the window. If your client is in verbose mode, the dialog in the log file is in verbose mode. If your client is not in verbose mode, the dialog is not in verbose mode.

Filenames on Windows Systems

You can use file paths and names with embedded spaces in them on Windows systems as long as you enclose the path and name with double quotes ("").

Displaying Current Settings

Use **status** to learn if logging is enabled or disabled.

Example None

Related Topics **status:** page 4-103

imrdw (OS/390 and z/OS)

Purpose Use **imrdw** to enable or disable the retention of record/span descriptor words (RDWs and SDWs) in V, VS, and VBS format source OS/390 and z/OS host datasets during subsequent transfers.

Syntax **imrdw**[_transfer] **ON** | **OFF**

Arguments **ON**

Enables RDW and SDW retention. On subsequent transfers, IFT treats RDWs and SDWs as if they are part of each record. IFT does not discard the RDWs and SDWs during transmission of V, VB, or VBS datasets from a mainframe system.

OFF

Disables RDW and SDW retention.

Usage Notes Keep the following points in mind:

Valid Hosts and Conditions

You can use **imrdw** from any client (mainframe or open systems) only when you have set data transmission to binary mode. In addition, either the local or remote server must be a mainframe system.

Available Forms

The **imrdw** command does not have a form without an argument. You must always supply an **ON** or **OFF** argument with **imrdw**.

RDW Format

The full-word RDW information is stored in binary format, and precedes each variable format logical record. The first two bytes contain the record length. The second two bytes are always zero.

More Information: Refer to IBM's *DFSMS/MVS: Using Data Sets* ¹ publication.

1. P/N SC26-4922, International Business Machines Corporation.

RDW Treatment

Issuing **imrdw ON** does not affect the transfers targeted to a mainframe dataset. If data targeted to a mainframe dataset contains embedded RDWs, IFT treats and stores the RDWs as data. IFT does not treat the RDWs as control information.

Displaying Current Settings

You can use **status** to display the current **imrdw** setting.

Example None

Related Topics

- ◆ **imaddlf (OS/390 and z/OS)**: page 4-32
- ◆ **imcrlf**: page 4-37
- ◆ **imtrail (OS/390 and z/OS)**: page 4-57
- ◆ **imwrap (Open Systems)**: page 4-59
- ◆ **upper**: page 4-110

imrecvx

Purpose Use **imrecvx** to specify a single-byte translation file or table to use for subsequent client **get** commands (overriding the default single byte translation file or table specified in the configuration file). Use **imrecvx** without a parameter to use the default translation file or table again.

Syntax **imrecvx**[late] [*<translation_file>*] (open systems)
imrecvx[late] [*<translation_table>*] (mainframe)

Argument *<translation_file>*

The (optional) name of the translation file residing on a UNIX or Windows.

The maximum length of the filename (including any path specification) is 150 characters. The file must have the extension *.imxlate*.

<translation_table>

The name of a translation table that resides in a sequential dataset or member of a partitioned dataset on the mainframe host involved in the transfer. The maximum length of the table name is 150 characters. The table does not have an extension.

Usage Notes Keep the following points in mind:

General Operation

The **imrecvx** command sets the translation table or file of either the local host or remote host, depending on where translation is to occur.

Valid Files

The **imrecvx** command is appropriate when transferring text files.

imrecvx and the Translation Location

The **imrecvx** command specifies the translation file or table on the host (either local or remote) designated to perform the translation. Make sure that you know where translation will occur before issuing this command.

Using the `imrecvx` Command Without an Argument

If you enter `imrecvx` without a `<translation_file>` or `<translation_table>` argument, IFT uses the receive translation file or table defined in the local IFT configuration file. If you previously issued `imxlate` with a `LOCAL` or `REMOTE` argument and then issue `imrecvx` without specifying a translation file or table, IFT uses the translation file or table defined in the configuration file on the local or remote host involved in the transfer.

Duration of Command Effect

The effect of `imrecvx` is temporary as it only sets the translation file or table for the currently-active connection. After you close the connection, the translation file or table settings return to those in the configuration file local to your client.

More Information: Refer to the *InfoMover File Transfer System Administrator's Guide* for a description of character translation.

Filenames on Windows Systems

You can use file paths and names with embedded spaces in them on Windows systems as long as you enclose the path and name with double quotes (“”).

`imrecvx` and the `status` command

Use the `status` command to verify both the translation file or table being used and the host (local or remote) that is performing the data translation.

Example None

Related Topics

- ◆ `binary`: page 4-4
- ◆ `get`: page 4-23
- ◆ `imsendx`: page 4-55
- ◆ `imxlate`: page 4-61
- ◆ `status`: page 4-103

imrelease (OS/390 and z/OS)

Purpose Use **imrelease** when an OS/390 or z/OS system is the target host of a subsequent data transfer. The **imrelease** command allows you to enable or disable automatic release of unused space when deallocating a new target dataset.

Syntax **imrelease** [**ON** | **OFF**]

Arguments **ON**

Enables automatic deallocation of unused space.

OFF

Disables automatic deallocation of unused space.

Usage Notes Keep the following points in mind:

When to Use the imrelease Command

Automatic deallocation is disabled when you start IFT clients and servers. Therefore, you need to issue **imrelease ON** to automatically release unused space for a future transfer; and issue **imrelease OFF** to return to the default, disabled state.

Displaying Current Setting

Use **status** to determine the current state of the automatic deallocation.

Example None

Related Topics

- ◆ **replace (OS/390 and z/OS)**: page 4-101
- ◆ **status**: page 4-103
- ◆ **verbose**: page 4-112

imsendx

Purpose Use **imsendx** to specify a single byte translation file or table for future client **put** commands (overriding the default single byte translation file or table specified in the configuration file). Issue **imsendx** without a parameter to use the default translation file or table again.

Syntax **imsendx**[late] [<translation_file>] (open systems)
imsendx[late] [<translation_table>] (mainframe)

Argument <translation_file>

The (optional) name of the translation file residing on a UNIX or Windows.

The maximum length of the filename (including any path specification) is 150 characters. The file must have the extension .imxlate.

<translation_table>

The name of a translation table that resides in a sequential dataset or in a member of a partitioned dataset on the mainframe host involved in the transfer. The maximum length of the table name is 150 characters. The table does not have an extension.

More Information: Refer to the *InfoMover File Transfer System Administrator's Guide* for a complete description of translation.

Usage Notes Keep the following points in mind:

General Operation

The **imsendx** command sets the translation table or file of either the local host or remote host, depending on where translation is to occur.

When to Use the imsendx Command

The **imsendx** command is only appropriate when transferring text files.

imsendx and the Translation Location

The **imsendx** command specifies the translation file or table on the host (either local or remote) designated to do the translation. Make sure that you know where translation will occur before issuing this command.

Using the **imsendx** Command Without an Argument

If you enter **imsendx** without a `<translation_file>` or `<translation_table>` argument, IFT uses the send translation file or table defined in the local configuration file.

If you previously issued **imxlate** with a **LOCAL** or **REMOTE** argument and then issue **imsendx** without specifying a translation file or table, IFT uses the translation file or table defined in the configuration file on the local or remote host involved in the transfer.

Duration of **imsendx** Effect

The **imsendx** command is temporary, as it only sets the translation file or table for the currently-active connection between IFT servers. After you close the connection between the IFT servers, the translation file or table settings return to those in the IFT configuration file local to your client.

imsendx and the **status** command

Use the **status** command to verify both the translation table or file being used and the host (local or remote) that is performing the translation.

Example

None

Related Topics

- ◆ **binary**: page 4-4
- ◆ **imrcvx**: page 4-52
- ◆ **imxlate**: page 4-61
- ◆ **put**: page 4-90
- ◆ **status**: page 4-103

imtrail (OS/390 and z/OS)

Purpose	Use imtrail to enable or disable retention of the trailing blanks in a mainframe host source during future transfers.
Syntax	imtrail [blanks] [ON OFF]
Arguments	ON Enables retention of trailing blanks. OFF Disables retention of trailing blanks.
Usage Notes	Keep the following points in mind: Valid Hosts The imtrail ON command retains trailing blanks in a mainframe fixed or fixed-block sequential dataset. The imtrail ON command is only valid if a mainframe server is active in the connection, and imxlate is enabled. <hr/> More Information: Refer to <i>imxlate</i> on page 4-61. Using the imtrail Command Without an Argument You can use imtrail without an argument to control retention of trailing blanks. The imtrail command without an argument toggles trailing blank retention on and off. On all hosts, trailing blank retention is disabled by default at client login. The first time you issue imtrail without an argument after client login, you enable trailing blank retention in a mainframe source dataset. Each subsequent imtrail you issue without an argument reverses the state of trailing blank retention. Mainframe Datasets with ISO Characters There are certain constraints you must keep in mind if you are transferring a mainframe file containing ISO control characters to an open systems host. Refer to <i>Managing Files with ISO Control Characters</i> on page B-27. Displaying Current Setting Use status to display the current imtrail setting.

Example None

- Related Topics**
- ◆ **imaddlf (OS/390 and z/OS):** page 4-32
 - ◆ **imcrlf:** page 4-37
 - ◆ **imwrap (Open Systems):** page 4-59
 - ◆ **upper:** page 4-110

imwrap (Open Systems)

Purpose Use **imwrap** to enable or disable wrap mode. Many records in UNIX and Windows systems have line lengths greater than target OS/390 and z/OS dataset logical record lengths. When you enable wrap mode by issuing **imwrap ON**, IFT wraps any line in a UNIX or Windows source file being transferred to a target mainframe system so that the line does not exceed the logical record length of the target datasets.

When you disable wrap mode by issuing **imwrap OFF**, IFT does not handle wrapping lines in a UNIX or Windows source file being transferred to a mainframe target.

Syntax **imwrap ON | OFF**

Arguments **ON**

Enables wrap mode.

OFF

Disables wrap mode.

Usage Notes Keep the following points in mind:

When to Use

The **imwrap** command is only valid when character translation (with **imxlate**) is enabled.

Command Form

The **imwrap** command does not have a “toggle” form without an argument. You must use **imwrap** with an **ON** or **OFF** argument. Otherwise, you receive an error.

Mainframe Target Dataset Size

When wrap mode is enabled, the resulting target dataset on a mainframe host is larger than the original source file by as much as 1%. This size adjustment is caused by the target dataset having more characters (linefeeds) than the source file.

You need to keep this size increase in mind when you preallocate the mainframe target dataset. Otherwise, you can receive *SX37* out of space abends.

Example None

- Related Topics**
- ◆ **imaddlf (OS/390 and z/OS):** page 4-32
 - ◆ **imcrif:** page 4-37
 - ◆ **imtrail (OS/390 and z/OS):** page 4-57
 - ◆ **upper:** page 4-110

imxlate

Purpose Use **imxlate** to enable single-byte character translation during subsequent transfers. You can use **imxlate** to implement either default translation (on the local host or, in mainframe to open systems transfers, on open systems host) or directly specify translation on either host involved in a transfer.

Syntax **imx**[late] [**ON** | **LOCAL** | **REMOTE** | **OFF**]

Arguments **ON**

Enables single-byte character translation for future data transfers. Uses translation method and translation files or tables specified in the local configuration file.

LOCAL

Overrides any value in the `imXlate_local_remote` configuration parameter in the local configuration file and enables single-byte character translation to take place on the local host (host A).

REMOTE

Overrides any value in the `imXlate_local_remote` configuration parameter in the local configuration file and enables single-byte character translation to take place on the remote host (host B).

OFF

Disables single-byte character translation for future data transfers.

Usage Notes Keep the following points in mind:

Using imxlate with the On Argument

Issuing **imxlate ON** enables translation. The specification of which host IFT uses for translation depends on the value of the configuration file parameter `imXlate_local_remote`.

For example, if you issue **imxlate ON** when `imXlate_local_remote` has the value of `local`, then translation is done on the local host (host A). If you issue **imxlate ON** when `imXlate_local_remote` has the value `remote`, then translation is done on the remote host (host B).

If `imXlate_local_remote` is not defined in your local `imconfig.ini` file, then IFT follows the default translation procedure. Translation is

done on the local host, unless the local host is a mainframe, in which case the translation is done on the open systems host.

Using the **imxlate** Command Without an Argument

You can use **imxlate** without an argument to enable single-byte character translation. The **imxlate** command without an argument implements default character translation (local host or, in mainframe to open systems transfers, open systems host).

If one or both of the hosts involved in a connection is an open system, translation mode is off by default. Issuing **imxlate** without an argument enables single-byte character translation mode on the open systems host.

If one of the hosts involved in a connection is a mainframe system, translation mode is on by default. Issuing **imxlate** without an argument, disables single-byte character translation mode.

Each subsequent **imxlate** you issue without an argument reverses the state of single-byte character translation.

Important: If you use **imxlate** without an argument, you can turn translation on or off, but you cannot specify whether translation occurs on the local or the remote host. Using the **imxlate** command without an argument always performs translation in the default mode; that is, translation is done on the local host unless the local host is a mainframe, in which case translation is done on the open systems host.

Effect of **imrecvx** and **imsendx** Commands

The **imxlate ON** command enables transferring files with data translation. IFT uses the translation files or translation tables specified in the `imXlateDir`, `recvXlateFile`, and `sendXlateFile` configuration parameters unless you override those values by specifying different translation files or tables as arguments of **imsendx** and **imrecvx**.

More Information: Refer to the *InfoMover File Transfer System Administrator's Guide* for a description of character translation.

imxlate and the **imxlate_local_remote** Configuration Parameter

You can use the `imXlate_local_remote` configuration to override the host during the translation when **imxlate ON** is specified. However, the **imxlate local** and **imxlate remote** provide complete control over where translation is done.

The `imxlate OFF` and `binary` Commands

You can use either `imxlate OFF` or `binary` to disable data translation.

Displaying Current Setting

Use `status` to display the current settings of data translation. The `status` command displays the translation file/table being used and indicates which host (local or remote) will perform the data translation.

Example None

Related Topics

- ◆ `binary`: page 4-4
- ◆ `imrcvx`: page 4-52
- ◆ `imsendx`: page 4-55
- ◆ `status`: page 4-103

lcd

Purpose Use **lcd** to change the working directory or file group on host A when the client is controlling the transfer through host A.

Syntax **lcd** [*<name>*]

Argument *<name>*

The (optional) name of the file directory. The maximum length of *<name>* is 150 characters.

Usage Notes Keep the following points in mind:

Using the Default Directory

To go to the default IFT directory, issue **lcd** without an argument.

Argument Values

The value you use for *<name>* when host A is a mainframe host must be a fully qualified dataset or a prefix. The value you use for *<name>* on a UNIX or Windows system must be a valid path.

Using lcd Without an Argument

If you do not specify *<name>*, IFT changes the working file directory or file group to that specified in the IFT configuration file on host A.

Verifying the Argument

When you issue **lcd** with a directory or file group name as an argument, IFT checks to see whether *<name>* is valid.

If *<name>* is invalid, IFT does not change the working directory or file group.

If the directory *<name>* is valid, but does not include a path in the syntax of the target operating system, IFT appends that directory to the current working directory or file group.

If the directory *<name>* includes a path in the syntax of the target operating system as an argument, the directory specified by that path becomes the current working directory or file group.

Mainframe Qualifiers and Dataset Names

Although **lcd** is foreign to a mainframe host, IFT treats the hierarchy of qualifiers and dataset names as though they were part of an open systems directory structure.

Syntax in a Mainframe Batch Environment

In the mainframe batch environment, you can continue **lcd** commands you specify with SYSIN DD statements across multiple lines.

More Information: Refer to *Continuation in SYSIN Control Statements* on page B-4.

Directory and Filenames in a Windows Environment

In a Windows environment, you can use directory names with embedded spaces if you enclose the directory name with double quotes. For example:

```
lcd "c:\Program Files"
```

Mainframe Examples

1. If the current working directory is TCPUSR14.TEST, the following **lcd** command makes the new working directory: TCPUSR14.TEST.FILES:

```
lcd FILES
```

2. To override the existing directory, rather than append to the directory, specify the directory argument with single quotation marks.
3. If a partitioned dataset (PDS) exists with the same name as the current working directory, IFT considers the working directory to be that PDS. Otherwise, IFT considers the working directory to be a common prefix qualifier for sequential datasets.

If a PDS exists with the same name as the current working directory, but you want the current working directory to be treated as a common prefix for sequential datasets, specify the working directory with a period (.) at the end of the name.

For example, if a PDS named TCPUSR14.TEST exists, the following **lcd** command makes the PDS TCPUSR14.TEST the current working directory:

```
lcd 'TCPUSR14.TEST'
```

A subsequent use of the **put** command with NAME1 adds a member NAME1 to the TCPUSR14.TEST PDS.

- In contrast, the following **lcd** command makes TCPUSR14.TEST., the current working directory, a prefix for sequential datasets:

```
lcd 'TCPUSR14.TEST.'
```

A subsequent **put** command, used to transfer dataset NAME1, would create the sequential dataset TCPUSR14.TEST.NAME1.

UNIX Examples

- If the current working directory is /tcpusr14/test, the following **lcd** command makes the new working directory /tcpusr14/test/files:

```
lcd files
```

- When you specify a path as the argument, the **lcd** command makes the directory specified by that path the working directory. If the current working directory is /tcpusr14/test, the following **lcd** command makes the new working directory /ftp/files:

```
lcd /ftp/files
```

Windows Examples

- If the current working directory is \tcpusr14\test, the following **lcd** command makes \tcpusr14\test\files the new working directory:

```
lcd files
```

- When you specify a path as the argument, the **cd** command makes the directory specified by that path the working directory. If the current working directory is \tcpusr14\test, the following **lcd** command makes \ftp\files the new working directory:

```
lcd \ftp\files
```

- If the current working directory is \tcpusr14\test on drive C:, the following **lcd** command makes e:\ftp\files the current working directory:

```
lcd e:\ftp\files
```

- If the directory name has embedded spaces in it, you must enclose the entire directory name in double quotes:

```
lcd c:\  
lcd "c:\Program Files\ftp\files"
```

5. To go to the root level of a drive, enter the **lcd** command with the drive letter.

For example: **lcd D:**

Related Topics

- ◆ **cd**: page 4-7
- ◆ **dir**: page 4-14
- ◆ **pwd**: page 4-97
- ◆ **ldir**: page 4-68
- ◆ **lpwd**: page 4-78

ldir

Purpose Use **ldir** to obtain a list of directory entries or a list of files in a file group on host A when the client is controlling the connection through host A. If host A is a mainframe system, **ldir** can list the members of a partitioned dataset (PDS) as well as auxiliary information about the members.

Syntax **ldir** | **lls** [*<name>*]

Arguments *<name>*

The (optional) directory or file group. The default is the current directory or file group. The maximum length of *<name>* is 150 characters. On mainframes, *<name>* must be fully qualified.

Mainframe Usage Notes

Keep the following points in mind when you use **ldir** on a mainframe host:

Alternate Command Word

You can use the characters **lls** as a synonym for **ldir**.

Maximum Number of Returned Mainframe Datasets

There is no limit to the number of mainframe datasets that can be returned to the client. However, because the list of datasets being sent from the server to the client is in buffers of 32767 bytes, performance may be slow.

Behavior of ldir Without an Argument

If the current working directory is not a PDS, **ldir** without an argument provides a list of all datasets starting with an index equal to the current working directory.

More Information: Refer to *ldir and Partitioned Data Sets* on page 4-69 for more information about the behavior of **ldir** without an argument when directed at partitioned datasets.

dir and GDGs

You can use **ldir** with a wildcard as part of the argument to display GDG basenames and provide an indicator of the datasets that are part of the GDG basename. For example:

```
ldir 'GSMFORM.gdg.test.**'
.      gdg .      .      .      GSMFORM.GDG.TEST
APP003 gdg vb 27998 4092 0      GSMFORM.GDG.TEST.G0007V00
APP000 gdg vb 27998 4092 0      GSMFORM.GDG.TEST.G0008V00
APP003 ps vb 27998 4092 0      GSMFORM.GDG.TEST.G0004V00
APP001 ps vb 27998 4092 0      GSMFORM.GDG.TEST.G0005V00
APP007 ps vb 27998 4092 0      GSMFORM.GDG.TEST.G0006V00
```

To list the associations of the GDG, use the form (*) as a wildcard:

```
ldir 'dtacks.gdg.test(*)'
APP003 gdg vb 27998 4092 0      DTACKS.GDG.TEST.G0007V00
APP000 gdg vb 27998 4092 0      DTACKS.GDG.TEST.G0008V00
```

ldir and Partitioned Data Sets

The **ldir** command follows the same rules as IBM FTP. For example, **ldir** without an argument provides a list of the members if the current local directory is a PDS:

```
cd 'invent.pds'
```

```
ldir
```

```
A
```

```
B
```

```
B
```

where:

A, B, and C are the partitioned dataset members.

If you specify the fully qualified PDS, you receive only the PDS name:

```
ldir 'invent.pds'
```

```
INVENT.PDS
```

If you do not want a list of PDS members, add a period character at the end of the partitioned dataset:

```
ldir 'invent.pds.'
```

This returns a search for any datasets that start with `invent.pds` and have further qualifiers, that is, this statement performs a search equivalent to using `invent.pds.**`.

Mainframe Wildcarding

Using **ldir**, you can perform catalog searches using DFSMS wildcarding. Table 4-3 shows the valid wildcards when host A is an IBM mainframe:

Table 4-3 Mainframe Idir Wildcard Characters

Character	Meaning
*	Either a qualifier or one or more characters within a qualifier. An asterisk can precede or follow any literal character, including a period (.) or a blank ().
**	Zero or more qualifiers. A double asterisk cannot precede or follow any characters but a period (.) or a blank ().
%	One alphanumeric or national character. You can specify up to eight percent characters in each qualifier.

With mainframe hosts, IFT follows the wildcarding conventions used by the IBM Catalog Search Interface (IGGCS100) program.

More Information: Refer to the IBM publication, *DFSMS Managing Catalogs*, for details about wildcarding conventions.

Cataloged Tape Datasets and GDG Base Entries

You cannot use **ldir** to obtain any information about cataloged tape datasets or GDG base entries.

Syntax in a Batch Environment

You can continue **ldir** commands in OS/390 and z/OS SYSIN DD statements across multiple lines. When placed at the end of a line, the minus sign (dash) character indicates that the statement is continued on the following line.

More Information: Refer to *Continuation in SYSIN Control Statements* on page B-4.

Open Systems Usage Notes

Keep the following points in mind when you use **ldir** on an open systems host:

Alternate Command Word

You can use the characters **lls** as a synonym for **ldir**.

Open Systems Wildcarding

Table 4-4 shows the wildcard characters that are valid when host A is an open systems host:

Table 4-4 Open Systems Idir Wildcard Characters

Wildcard Character	Meaning
* (asterisk)	Zero or more alphanumeric characters in a string of characters. An asterisk can precede or follow any literal character. For example, <code>dat*</code> , returns <code>dat</code> , <code>date</code> , <code>data</code> , and any other string that starts with <code>dat</code> and has any kind of ending.
? (question mark)	One alphanumeric character in a string of characters. You can specify one to eight question marks in each qualifier. For example, <code>dat?</code> returns <code>date</code> or <code>data</code> . It does not return <code>dat</code> or any string that starts with <code>dat</code> and has more than one additional terminating character.

Windows Command Behavior

The `Idir` command on a Windows host shows hidden files as well as files with other attributes.

Directory Names in Windows Systems

In a Windows environment, you can use directory names with embedded spaces if you enclose the directory name with double quotes.

For example:

```
ldir "Warehouse Files"
```

Mainframe Examples

1. The following example lists all datasets with the high level qualifier INVENT that end with a qualifier that is exactly one character and have any number and length of intervening qualifiers.:

```

ldir 'INVENT.**.%'
APP004 ps vb 6144 1028 0 INVENT.FTP.X
APP011 po u 27998 0 0 INVENT.LONG.DATASET.A.B.C.D.E.F.G.H.I.J.K
APP002 ps fb 27920 80 0 INVENT.P
APP007 po vb 27998 180 0 INVENT.PERFTEST.H
APP006 ps vb 6233 256 0 INVENT.SAMSAI.H
APP002 po vb 6144 1028 0 INVENT.SOCKET.C
VISA00 po vb 27998 240 0 INVENT.S165Y.H
VISA06 po vb 27998 240 0 INVENT.V1R6M09A.H
APP001 po vb 27998 240 0 INVENT.V1R6M10.H
VISA06 po vb 27998 240 0 INVENT.V1R6M10A.H
APP010 po vb 27998 240 0 INVENT.V1R6M12.H
APP003 po vb 27998 240 0 INVENT.V1R6M12A.H
...
..
..

```

2. The following examples show how to use wildcarding with **ldir** in a mainframe situation:

- `ldir 'PATREC.*.SET` returns the following:

```

PATREC.DATA1.SET
PATREC.DATA2.SET

```

but, does not return `PATREC.DATA.SET.KSDS`

- `ldir 'PATREC.*A'` returns the following:

```

PATREC.A
PATREC.BA
PATREC.BBA

```

but, does not return:

```

PATREC.B
PATREC.AB

```

- `ldir 'PATREC.**'` returns the following:

```

PATREC
PATREC.DATA.SET1
PATREC.DATA.SET2

```

but, does not return `PATREC1.DATA.SET`.

- `ldir patrec.data.set%` returns the following:

```
PATREC.DATA.SET1
PATREC.DATA.SET2
```

but, does not return `PATREC.DATA.SET30`

- `ldir patrec.data.set%%` returns the following:

```
PATREC.DATA.SET30
PATREC.DATA.SET40
```

but, does not return:

```
PATREC.DATA.SET3
PATREC.DATA.SET4
```

3. You can list the datasets that start with a specific “index.”

For example, if the current working directory were set to the index level `DBUTCHA.IM` in a mainframe environment, the `ldir` command may display the following files:

```
ldir
TS0003      PS      FB      3120      80      0      DBUTCHA.IM.JCL
TS0002      PO      FB      2400      80      0      DBUTCHA.IM.CHUNK1
```

The diagram illustrates the output of the `ldir` command. It shows two lines of output: `TS0003` and `TS0002` on the first line, and `PS`, `PO`, `FB`, `FB`, `3120`, `2400`, `80`, `80`, `0`, `0` on the second line. Below the output, several labels with arrows point to specific fields: `Volume Serial Number (volser)` points to `TS0003` and `TS0002`; `Dataset Organization` points to `PS` and `PO`; `Record Format` points to `FB` and `FB`; `Block Size` points to `3120` and `2400`; `Logical Record Length` points to `80` and `80`; `Key Length` points to `0` and `0`; and `Dataset Names` points to `DBUTCHA.IM.JCL` and `DBUTCHA.IM.CHUNK1`.

UNIX Example You can use **ldir** with a directory name to display a list of files. Look at the following example:

```
ldir /usr/imqa
total 150
-rw-r--r-- 1 root other 461      Sep 16 09:45 SQL.1.17
-rw-r--r-- 1 root other 0       Sep 12 17:01 emcfmt.lock
drwxr-xr-x 2 root other 512      Sep 16 17:30 foo.bar
-rwxr-xr-x 1 root other 17922   Sep 18 11:58 text.dat
-rwxr-xr-x 1 root other 13103   Sep 18 11:58 xlab.tst
-rw-r--r-- 1 root other 318      Sep 16 16:00 xlab1.txt
-rwxr-xr-x 1 root other 41159   Sep 18 11:58 fillup.1
```

Windows Example You can use **ldir** with the name of a directory to display a list of files. Look at the following example:

```
ldir C:\medrec\smith\2000
Volume in drive C has no label.
Volume Serial Number is 311F-0801

Directory of C: \medrec\smith\2000

04-22-2004  8:21a  <DIR>  tests
05-14-2004  3:31p  <DIR>  billing
05-14-2004  3:34p  34466  history.txt
```

- Related Topics**
- ◆ **cd**: page 4-7
 - ◆ **lcd**: page 4-64
 - ◆ **lpwd**: page 4-78
 - ◆ **pwd**: page 4-97
 - ◆ **dir**: page 4-14

likedsn (OS/390 and z/OS)

Purpose Use **likedsn** when an OS/390 or z/OS system is the target host of a subsequent data transfer. You use **likedsn** to set allocation attributes for new mainframe host datasets that are the targets of IFT commands.

The *<model>* argument specifies the name of a dataset you want to use as a model for the allocation attributes for any new mainframe datasets that are the targets of the subsequent **get** or **put** commands.

The **likedsn** command without an argument clears the model dataset name used previously.

Syntax **likedsn** [*<model>*]

Argument *<model>*

The name of a model dataset whose attributes you want to use for future dataset allocations.

Usage Notes Keep the following points in mind:

Valid Targets

You can only use **likedsn** for transfers involving a mainframe target.

Specifications for the Model Dataset

The model dataset must be:

- ◆ An existing cataloged dataset
- ◆ A sequential dataset
- ◆ Reside on a direct access volume

The model dataset can be a generation of a generation data group (GDG). You can refer to the generation dataset (GDS) by either the GDG absolute or relative dataset name.

To qualify the dataset, enclose *<model>* in single quotes. If you do not enclose *<model>* in single quotes, IFT adds the current high-level qualifier as a prefix to *<model>*.

Effect in SMS and Non-SMS Environments

In an SMS environment, IFT uses the same attributes that would be copied if you had specified the LIKE argument in a JCL data definition statement. These include:

- ◆ Record format (RECFM)
- ◆ Record length (LRECL)
- ◆ Space allocation (SPACE)

Keep in mind that the system determines the space to be allocated by adding up the space allocated in the first three extents of the model dataset. Therefore, the space allocated for the new dataset generally does not match the space that you specified for the model.

You can insure proper space allocation by using the **imcylinder** command to specify the space allocation explicitly.

In a non-SMS environment, IFT uses the same attributes that you be copied if you had used a DCB=<dsname> argument in a JCL data definition statement. These include:

- ◆ Record format (RECFM)
- ◆ Record length (LRECL)
- ◆ Block size (BLKSIZE)

Note that space allocation is determined by the IFT default, or by the value specified in **imcylinder**.

More Information: Refer to *Specifying Allocation Attributes* on page B-13 for the complete set of rules that determine the allocation attributes of a new dataset in an SMS environment and a non-SMS environment.

Interaction of likedsn and imcylinder, and the Configuration File

The following points describe the interaction of **likedsn** and **imcylinder**, and the default cylinder value in the configuration file (the default_cylinders configuration parameter) in setting the space attributes of new target datasets:

- ◆ If you issue both **likedsn** and **imcylinder**, IFT uses the space attributes of **imcylinder** rather than the space attributes of **likedsn** argument.
- ◆ If you issue **likedsn** and issue **imcylinder** without an argument (or if you did not issue **imcylinder**), IFT uses the space attributes of the **likedsn** argument.
- ◆ If you do not issue either **likedsn** or **imcylinder**, IFT uses the value in the default cylinder value in the IFT configuration file.

Displaying Current Settings

You can use **status** to display the current setting of the model dataset name.

Examples

1. The following example specifies that new dataset allocations obtain the attributes from the sequential dataset AGTEST.DATASET:

```
likedsn 'AGTEST.DATASET'
```

2. The following example specifies that new dataset allocations obtain their attributes from the GDG dataset AGTEST.GDG.DATA.SET.G0003V00.

Because the dataset is not fully qualified within quotes, IFT appends the current high-level qualifier, AGTEST, to the dataset name:

```
likedsn GDG.DATA.SET.G0003V00
```

3. The following example specifies that new dataset allocations obtain their attributes from the current generation of the AGTEST.GDG.DATA.SET GDG dataset.

As the following example shows, **likedsn** accepts GDG absolute or relative dataset names:

```
likedsn 'AGTEST.GDG.DATA.SET(0)'
```

Related Topics

- ◆ **imbufno** (OS/390 and z/OS): page 4-35
- ◆ **imcylinder** (OS/390 and z/OS): page 4-40
- ◆ **imdirblocks** (OS/390 and z/OS): page 4-44
- ◆ **imrdw** (OS/390 and z/OS): page 4-50
- ◆ **lrecl** (OS/390 and z/OS): page 4-79
- ◆ **recfm** (OS/390 and z/OS): page 4-99
- ◆ **unit** (OS/390 and z/OS): page 4-107
- ◆ **volcount** (OS/390 and z/OS): page 4-113
- ◆ **volume** (OS/390 and z/OS): page 4-115

lpwd

Purpose Use **lpwd** to display the name of the current working directory on host A when the client is controlling the connection through host A.

Syntax **lpwd**

Argument None

Usage Notes None

Examples 1. If host A is an OS/390 or z/OS system, the response to **lpwd** may be as follows:

```
lpwd  
Command: lpwd  
  
CM.CM113
```

2. If host A is a UNIX system, the response to **lpwd** may be as follows:

```
lpwd  
Command: lpwd  
  
/tmp
```

3. If host A is a Windows system, the response to **lpwd** may be as follows:

```
lpwd  
Command: lpwd  
  
\Temp
```

Related Topics

- ◆ **cd**: page 4-7
- ◆ **dir**: page 4-14
- ◆ **pwd**: page 4-97
- ◆ **lcd**: page 4-64
- ◆ **ldir**: page 4-68

lrecl (OS/390 and z/OS)

Purpose Use **lrecl** when an OS/390 or z/OS system is the target host of a subsequent transfer. Refer to `JCL DCB=lrecl=bytes`. The **lrecl** command overrides the default record length value in the IFT configuration file for new dataset allocations (the default `_lrecl` configuration parameter). The **lrecl** command allows you to specify:

- ◆ Length for fixed length records
- ◆ Maximum length for variable-length records

Using **lrecl** without an argument resets the length to use in future dataset allocations to the default record length value in the configuration file.

Syntax **lrecl** [*<bytes>*]

Argument *<bytes>*

The length (in bytes) for fixed length records or the maximum length for variable length records. The values you can use are from 0 through 32768.

Usage Notes Use **status** to determine the length in bytes that IFT is currently using.

Example None

Related Topics

- ◆ **imbufno** (OS/390 and z/OS): page 4-35
- ◆ **imcylinder** (OS/390 and z/OS): page 4-40
- ◆ **imdirblocks** (OS/390 and z/OS): page 4-44
- ◆ **imfilenumber** (OS/390 and z/OS): page 4-46
- ◆ **imrdw** (OS/390 and z/OS): page 4-50
- ◆ **status**: page 4-103
- ◆ **verbose**: page 4-112
- ◆ **volcount** (OS/390 and z/OS): page 4-113

lsp (OS/390 and z/OS)

Purpose Use **lsp** to list the available space (free space) on a specified volume on a remote mainframe host.

Syntax **lsp** <volser>

Argument <volser>

The 6-character serial number of the volume on a mainframe system for which you want to check the available space.

Usage Notes Keep the following points in mind:

Valid Hosts

You can enter **lsp** from a CLI client on any host, but you can only direct **lsp** to a mainframe system.

Example The following command lists the available space on device SIVA01:

```
lsp SIVA01
```

```
Command: lsp SIVA01
```

```
Free Cylinders in SIVA01 704
```

Related Topics

- ◆ **cd**: page 4-7
- ◆ **dir**: page 4-14
- ◆ **pwd**: page 4-97

mget

Purpose Use **mget** to transfer all files on host B that match the *<pattern>* argument to the current directory on host A when the client is controlling the transfer through host A.

Syntax **mget** *<pattern>*

Argument *<pattern>*

The pattern that specifies the general criteria for the files you want to transfer. If you use multiple patterns in the argument, separate each pattern with a single space.

Usage Notes Keep the following points in mind:

Dependencies

The IFT client must be connected to the remote server through which the specified files are to be retrieved.

The prompt Command Toggle

The **prompt** command specifies whether you are prompted to verify transfers. If you issue **prompt ON** (the default), **mget** displays the names of the files that match *<pattern>*, and asks you if you want to transfer those files.

If you issue **prompt OFF**, **mget** does not display the names of the files that match *<pattern>*, and does not ask you if you want to transfer those files.

Interactive Prompting on Mainframe Systems

When you run batch jobs on mainframe systems, interactive prompting is always off. You cannot turn interactive prompting on when you run in batch mode.

Patterns

The **mget** command *<pattern>* argument can be a string value, or the name of a file that contains the pattern you want to use as a criteria. Whether as a file or a string, the pattern itself can contain specific values such as a relative or absolute pathname, a filename or extension, or wildcards.

Wildcarding

Table 4-5 shows the wildcard characters that are valid on various host types:

Table 4-5 Valid mget Wildcard Characters

Wildcard Character	Valid if host B is	Meaning
* (asterisk)	An IBM mainframe ^a , UNIX, or Windows host	<p>On a mainframe: Represents either a qualifier or one or more characters within a qualifier. An asterisk can precede or follow any literal character, including period (.) or blank ().</p> <p>On an open system: Represents zero or more characters within a string of characters. An asterisk can precede or follow any literal character.</p>
** (double asterisk)	An IBM mainframe host ^a	Represents zero or more qualifiers. A double asterisk cannot precede or follow any characters but a period (.) or a blank ().
% (percent)	An IBM mainframe host ^a	Represents one alphanumeric or national character. You can specify up to eight percent characters in each qualifier.
? (question mark)	A UNIX or Windows host	Represents one alphanumeric character in a path specification.

a. IFT on mainframe hosts follows the wildcarding conventions used by the IGGCS100 program. Refer to the IBM publication, *DFSMS Managing Catalogs*, for more information about mainframe wildcarding.

Pathnames Used in Patterns

If you do not use a pathname in `<pattern>`, IFT searches for files that match the pattern in the current directory on the source host (host A).

When you use a pathname in a pattern, the action takes depends on whether the pathname is relative or absolute. If you enter a relative pathname, IFT uses the current directory as a base to find the files specified by the relative pathname. If you enter an absolute pathname, IFT goes to that pathname to find matches.

Limitations

There is a limited number of possible matches from the specified pattern with wildcards. This value is set by the configuration file `max_pattern_matches` parameter. The default value for this parameter is 2000. The server returns an error if the number of matches is equal to `max_pattern_matches` parameter.

- Examples**
1. The following example shows the first step in a transfer between two UNIX hosts: alpha and beta. In the example, all files that match the pattern `text[0-9]` from the current directory on the remote host (beta) to the working directory on the local host (alpha):

```
IM:alpha> mget text[0-9]
```

If prompting is enabled, this command lists all the files `text0` through `text9` in the working directory on beta, and asks you to specify whether you want to transfer each file.

2. The following example shows a transfer between a local mainframe host and a remote UNIX host. In this command (given in batch), the files `file1`, `file2`, and files with the extension `.c` are transferred from the remote UNIX host:

```
mget file1 file2 *.c
```

3. The following example uses a relative filename in a pattern to transfer all files in the directory one level above the current directory on the remote host. All the files in the directory one level above the this directory match this pattern:

```
IM: WIN> mget ../*
```

Related Topics

- ◆ **get**: page 4-23
- ◆ **mput**: page 4-84
- ◆ **prompt**: page 4-89
- ◆ **put**: page 4-90

mput

Purpose Use **mput** to transfer all files on host A that match the *<pattern>* argument to the current directory on host B when the client is controlling the transfer through host A.

Syntax **mput** *<pattern>*

Argument *<pattern>*

The pattern that specifies the general criteria for the files you want to transfer. If you use multiple patterns in the argument, separate each pattern with a single space.

Usage Notes Keep the following points in mind:

Dependencies

The IFT client must be connected to the remote server through which the specified files are to be sent.

The prompt Command Toggle

The **prompt** command specifies whether or not you are prompted to verify transfers. If you issue **prompt ON** (the default), **mput** displays the names of the files that match *<pattern>*, and asks you if you want to transfer those files.

If you issue **prompt OFF**, **mput** does not display the names of the files that match *<pattern>*, and does not ask you if you want to transfer those files.

Interactive Prompting on Mainframe Systems

When you run batch jobs on mainframe systems, interactive prompting is always off. You cannot turn interactive prompting on when you run in batch mode.

Patterns

The **mput** command *<pattern>* argument can be a string value, or the name of a file that contains the *<pattern>* argument you want to use as a criteria. Whether as a file or a string, *<pattern>* itself can contain specific values such as a relative or absolute pathname, a filename or extension, or wildcards.

Wildcarding

Table 4-6 shows the wildcard characters that are valid on various host types:

Table 4-6 Valid Input Wildcard Characters

Wildcard Character	Valid if host A is	Meaning
* (asterisk)	An IBM mainframe ^a , UNIX, or Windows host	<p>On a mainframe: Represents either a qualifier or one or more characters within a qualifier. An asterisk can precede or follow any literal character, including period (.) or blank ().</p> <p>On an open system: Represents zero or more characters within a string of characters. An asterisk can precede or follow any literal character.</p>
** (double asterisk)	An IBM mainframe host ^a	Represents zero or more qualifiers. A double asterisk cannot precede or follow any characters but a period (.) or a blank ().
% (percent)	An IBM mainframe host ^a	Represents one alphanumeric or national character. You can specify up to eight percent characters in each qualifier.
? (question mark)	A UNIX or Windows host	Represents one alphanumeric character in a path specification.

- a. IFT on mainframe hosts follows the wildcarding conventions used by the IGGCS100 program. Refer to the IBM publication, *DFSMS Managing Catalogs*, for more information about mainframe wildcarding.

Pathnames Used in Patterns

If you do not use a pathname in a pattern, IFT searches for files that match *<pattern>* in the current directory on the target host (host B).

When you use a pathname in *<pattern>* the action IFT takes depends on whether the pathname is relative or absolute. If you specify a relative pathname, IFT uses the current directory as a base to find the files specified by the relative pathname. If you specify an absolute pathname, IFT goes to that pathname to find matches.

Limitations

There is a limited number of possible matches from the specified pattern with wildcards. This value is set by the configuration file `max_pattern_matches` parameter. The default value for this parameter is 2000. The server returns an error if the number of matches is equal to `max_pattern_matches` configuration parameter.

Examples

1. The following example shows the first step in a transfer between two UNIX hosts: alpha and beta. In this case, **mput** finds all the files with the extension `.c` in the working directory on local host alpha:

```
IM: beta> mput *.c
```

If prompting is enabled, the user is prompted to specify whether they want to transfer these files to the working directory on remote host beta.

2. The following example finds all files with the extension `.exe` in the directory `C:\test` to the current directory on the remote host:

```
IM: WIN> mput C:\test\*.exe
```

3. The following example transfers the files IFT.TEST1, IFT.TEST2, IFT.TEST3, and all matching members in the PDS IFT.PDS to the working directory on the remote host:

```
mput 'IFT.TEST1' 'IFT.TEST2' 'IFT.TEST3' 'IFT.PDS(TEST*)'
```

Related Topics

- ◆ **get**: page 4-23
- ◆ **mget**: page 4-81
- ◆ **prompt**: page 4-89
- ◆ **put**: page 4-90

open

Purpose Use **open** to connect to a server on a remote host. When you issue **open**, IFT prompts you to enter a username and password that has valid access to the server.

If you enter a valid username and password, IFT connects to the server. IFT puts you into the default IFT directory of the host to which you connected. If you enter a username and password that is not valid on the server host, you cannot connect.

Note: The maximum length of an IFT password on mainframe hosts is eight characters. The maximum length of an IFT password on open systems hosts is 16 characters. You cannot enter a null (blank) password.

Syntax **open** <server>

Argument <server>

The name of the IFT server with which you want to set up a session.

Usage Notes Keep the following points in mind:

Valid Client Server Connections

Table 3-1 on page 3-11 contains a list of the valid client-server connections you can make.

Server Names

Server names on mainframe systems can be up to 32 characters, and must be composed of all uppercase alphabetic characters, numeric characters, or symbolic characters, with the exception of slash (/) and semicolon (;).

Server names on open systems can be up to 32 characters long.

Open and Close Commands

The **open** command is the opposite of the **close** command. The **open** command establishes a connection between the IFT current server to which you are connected, and the IFT server you specified as an argument value. When it performs the **open**, **open** prints an informational message that includes the date and time.

Incorrect Server Names

If you spell the name of the server incorrectly, the IFT client does nothing, and eventually times out. This action produces an error at the client interface.

Leaving a Client Idle After Establishing a Server Connection

If you use a client to open a connection to a server and then perform no actions before the server times out and closes that connection, any subsequent transfer command from that client causes a nondescriptive error message. The client then exits.

There are no other side effects to leaving an idle client to time out.

You can use the client to reopen the connection to the server.

Subsequent connection requests you make with the client complete normally.

Opening a Server that is Already Opened

If you open a server that is already open, **open** returns a code of zero (0 = success). This action is similar to client-only commands (commands that do not go to the server) that execute correctly, and also return a code of zero.

Example

When you issue **open** and successfully log in, IFT responds with the name of the IFT server on host A that is connected. The following example opens a mainframe server:

```
IM: HOMEHST> open CARLACN
Command: open CARLACN

Username (root): ccaput1
Password:
Attempting connection to CARLACN...

Connected to IFT Server: CARLACN (10/26/2004 08:07:18)
Server CARLACN is an MVS server type
Enter file transfer commands
IM: CARLACN>
```

Related Topics

- ◆ **close**: page 4-11
- ◆ **quit**: page 4-98

prompt

Purpose Use **prompt** to enable or disable prompting on the client.

Syntax **prompt** [**ON** | **OFF**]

Argument **ON**

Enable prompting.

OFF

Disable prompting.

Usage Notes Keep the following points in mind:

Interactive Prompting Behavior

When interactive prompting is on, the client requests extra information from the user. For instance, if interactive prompting is on when you use **mget**, the client asks you to confirm the file transfer.

Using the **prompt** Command Without Arguments

If you specify **prompt** without arguments, you change the current state of prompting. That is, when prompting is disabled, issuing **prompt** without an argument enables prompting. When prompting is enabled, issuing **prompt** without an argument disables prompting.

Interactive Prompting on Mainframe Systems

When you run jobs on mainframe systems in batch mode, interactive prompting is always disabled. You cannot enable it with **prompt**.

Example None.

Related Topics

- ◆ **mget**: page 4-81
- ◆ **mput**: page 4-84

put

Purpose Use **put** to copy a file or dataset from the host A to the host B when the client is controlling operations through the host A.

Syntax **put** *<source>* [*<target>*]

Arguments *<source>*

The name of the source file or dataset on host A. The maximum length of *<source>* (including any path specification) is 150 characters.

<target>

Specifies the name of the target file or dataset on host B. The maximum length of *<target>* (including any path specification) is 150 characters.

Usage Notes Keep the following points in mind:

Dependencies

The IFT client must be connected to the remote server to which the specified file is to be sent.

Valid Client Server Connections

In many cases, host A is a local host and host B is a remote host. However, if you are using an open systems-based client, you may be using a thin client on a third host; both hosts may be remote.

More Information: Refer to Table 3-1 on page 3-11 for a list of the valid client-server connections you can make.

Command Failure

If **put** fails because *<source>* does not exist (or there is some other error condition in the source file) and *<target>* is a new allocation of an OS/390 or z/OS dataset (or a new GDG generation), IFT deletes the target dataset.

Mainframe Source Specifications

IFT adds the current high-level qualifier as a prefix to *<source>* if you do not fully qualify source by enclosing it in single quotes.

Open Systems Source Specifications

If *<source>* is in the current working IFT directory on host A, you do not need to specify the directory path in *<source>*. If *<source>* is not in the current working IFT directory on host A, you must specify the directory path in *<source>*.

Mainframe Systems Target Specifications

If host B is a mainframe system and you specify *<target>* without single quotes, IFT creates a dataset named *<target>* in the default high-level qualifier specified during installation and/or configuration on host B.

If you do not specify *<target>*, IFT extracts the file/dataset name from source, and creates a dataset with that name in the default high-level qualifier specified during installation and/or configuration on host B.

If you specify *<source>* with single quotes, IFT creates *<target>* as specified.

Open Systems Target Specifications

If you specify *<target>* without a directory path in a transfer to an open system, IFT creates a file named *<target>* in the default directory specified during IFT installation and/or configuration.

If you do not specify *<target>* in a transfer to an open system, IFT extracts the filename from *<source>*, and create a target file of that name in the default directory specified during IFT installation and/or configuration.

If *<target>* is on a Windows system, you can specify *<target>* with either:

- ◆ Windows syntax for the pathname
For example: `d:\temp\file_name`
- ◆ The Universal Naming Conventions (UNC) to specify *<source>* or *<target>*
For example:
`//machine_name/share_name/directory_name/file_name`

Mainframe Target Dataset Default Attributes

In the absence of any other allocation specifications (through setting optional configuration values or using CLI commands), IFT creates mainframe target datasets with the following attributes:

- ◆ Variable block record format
- ◆ A 4092-record length
- ◆ A 2-cylinder primary extent
- ◆ A 5-cylinder secondary extent
- ◆ A default unit of SYSDA

The target dataset is given the default block size of the system on which it is being created.

More Information: Refer to *Specifying Allocation Attributes* on page B-13 for a description of dataset attributes.

Target Permissions

Target permissions are as follows:

- ◆ In OS/390 and z/OS host environments, the target dataset only has the permissions imposed upon it by the security management system. See your site security administrator for special requirements.
- ◆ In UNIX host environments, all newly created UNIX target files have 644 permission set for them.
- ◆ In a Windows host environment, the target file permission settings are those of the directory into which you place the target.

Transfer Behavior in Replace Mode

If the target dataset or file on host A already exists, and you previously specified replace mode (that is, issued **replace ON**), the data are overwritten. The source dataset or file on host B retains its existing characteristics.

GDGs as Sources and Targets

IFT accepts absolute and relative generation data group, dataset names as source and target specifications in **get** commands. You cannot transfer an entire GDG, only an individual generation dataset.

More Information: Refer to *Using Generation Data Groups* on page B-7 for a further discussion of GDGs.

Magnetic Tape Sources and Targets on Mainframe Systems

IFT supports the use of datasets on magnetic tape devices as sources and targets on mainframe hosts.

Magnetic Tape Sources and Targets on UNIX and Windows Systems

IFT does not support the use of files on magnetic tape devices as sources or targets on UNIX and Windows hosts.

Syntax in a Mainframe Batch Environment

You can continue **put** commands you specify with **SYSIN DD** statements across multiple lines.

More Information: Refer to *Continuation in SYSIN Control Statements* on page B-4.

Failed Mainframe Transfers

If a transfer fails, IFT truncates the existing target dataset on the mainframe host to zero bytes but it is not deleted. You must delete it. For a new target dataset, it is deleted automatically.

Using Universal Naming Conventions on Windows Systems

If the IFT client, the IFT server, and *<source>* or target are on a Windows system, you can use the Universal Naming Conventions (UNC) to specify the source or target:

```
//machine_name/share_name/directory_name/file_name
```

Windows Path Syntax

If *<source>* or target is on a Windows system, you can also use Windows syntax for the pathname. For example:

```
d:\temp\file_name
```

You can use source and target filenames with embedded spaces if you enclose the directory name with double quotes. For example:

```
put usr/reports/new "D:\temp\Warehouse Files\New Data"
```

Pipe Sources and Targets on Mainframe and Windows Systems

IFT does not support the use of batch pipes as sources or targets on mainframe systems, or named pipes as sources or targets on Windows systems.

Named Pipes as Sources and Targets on UNIX Systems

IFT supports the use of named pipes as source and target files on a UNIX system. When you use a named pipe as a source or target on a UNIX system, the other participant in the transfer can be located on any host platform.

To use a pipe as a UNIX source or target, you must take the following steps:

- a. Create the pipe outside IFT at some point before the transfer begins.
- b. Start the process to the pipe either before the transfer is started or within two minutes after the transfer is started.

If the process is not started to the pipe within two minutes after the transfer begins, you receive the following error:

```
IM003E: Error: Failed to open file
```

To use a UNIX pipe as a target, you must have previously issued **replace ON**.

Transfer Verification Messages

After the file is transferred, IFT displays:

- ◆ The status of the transfer
- ◆ The source and target names
- ◆ The completion time
- ◆ The elapsed time in seconds
- ◆ File sizes in bytes
- ◆ Transfer speed in kilobytes per second

Displayed Transfer Times

The transfer time, displayed with **put**, is expressed in whole seconds rather than fractions of a second. Because of that, some transfers, especially those of small files, appear to take longer than they actually do. For instance, **put** in verbose mode reports that a transfer that took 1.2 seconds as taking 2 seconds.

Disabled put Commands

There may be some hosts from which you cannot transfer files with the **put** command. By setting configuration parameters system managers or system programmers can disable **put** and **del** commands for selected users or selected hosts. Consult your system administrator or system programmer for more information.

Examples

1. The following example shows **put** being used to transfer a file from the default IFT directory on host A, a UNIX system, to a specified directory and target filename on host B, a mainframe system:

```
put file1 'CM.CIM113.C(CIM10)'
```

```
Data will be translated using (Default)
Transferring file file1
IM000I Information: File transfer successfully completed.
Source = file1
Target = 'CM.CIM113.C(CIM10)
Time: (09/18/2004 13:42:55) (1)
Elapsed=29,FileSize=147465112,KbytePerSec=5085
```

2. In the following example, host A is a UNIX system, and the complete name of the source is `/usr/dtest/file1`. You can issue **put** to override use of the default IFT directory as follows:

```
put /usr/dtest/file1 'IM.IM113.C(EMCIM10)'
```

```
Data will be transferred using (Default)
Transferring file file1
IM000I Information: File transfer successfully completed.
Source = file1
Target = 'IM.IM113.C(EMCIM10)
Time: (09/18/2004 13:42:55) (1)
Elapsed=26,FileSize=133068089,KbytePerSec=5118
```

3. If both host A and host B are UNIX systems, the sequence may be as follows:

```
put /usr/dtest/file2 /usr/dtest/file2copy
```

```
Data will not be translated
Transferring file /usr/dtest/file2
IM000I Information: File transfer successfully completed.
Source = /usr/dtest/file2
Target = /usr/dtest/file2copy
Time: (09/18/2004 13:15:55) (1)
Elapsed=29,FileSize=147987076,KbytePerSec=5103
```

4. If host A is a Windows system and host B is a UNIX system, the sequence may be as follows:

```
put d:\temp\imData\file2 /usr/dtest/file2copy
```

```
Command: put d:\temp\imData\file2 /usr/dtest/file2copy  
Data will not be translated  
Transferring file d:\temp\imData\file2  
IM000I Information: File transfer successfully completed.  
Source = 'd:\temp\imData\file2'  
Target = '/usr/dtest/file2copy'  
Time: (09/18/2004 15:33:55) (1)  
Elapsed=29,FileSize=147987076,KbytePerSec=5103
```

Related Topics

- ◆ **del**: page 4-12
- ◆ **get**: page 4-23
- ◆ **mget**: page 4-81
- ◆ **mput**: page 4-84
- ◆ **replace (OS/390 and z/OS)**: page 4-101
- ◆ **status**: page 4-103
- ◆ **verbose**: page 4-112

pwd

Purpose Use **pwd** to display the name of the current working directory on host B when your client is controlling the connection through host A.

Syntax **pwd**

Argument None

Usage Notes None

Examples 1. If host B is an OS/390 or z/OS system, the response to **pwd** may be as follows:

```
pwd  
Command: pwd  
  
CM.CM113
```

2. If host B is a UNIX system, the response to **pwd** may be as follows:

```
pwd  
Command: pwd  
  
/tmp
```

3. If host B is a Window NT/2000 system, the response to **pwd** may be as follows:

```
pwd  
Command: pwd  
  
\Temp
```

Related Topics

- ◆ **cd**: page 4-7
- ◆ **dir**: page 4-14
- ◆ **lcd**: page 4-64
- ◆ **ldir**: page 4-68
- ◆ **lpwd**: page 4-78

quit

- Purpose** Use **quit** to do the following:
- Disconnect from host B (when the client is controlling operations through host A)
 - Close connection from the local server (host A)
 - Exit from the IFT client

If there is no connection to a remote server, **quit** simply disconnects from the local server, and exit from the IFT client. At exit, **quit** prints an informational message, including the date and time of the exit.

Syntax `qu[it]`

Arguments None

Usage Notes None

Example The following is an example of **quit**:

quit

Command: quit

```
Closing connection to remote IFT server cmcSQL. (11/26/2004 08:01:17)
```

```
Closing connection with local server. (11/26/2004 08:01:17)
```

- Related Topics**
- ◆ **close**: page 4-11
 - ◆ **open**: page 4-87

recfm (OS/390 and z/OS)

Purpose Use **recfm** for new mainframe target datasets. Refer to JCL `DCB=recfm=format`. The **recfm** command allows you to:

- ◆ Override the value in the default record format value in the IFT configuration file for new dataset allocations (the `default_recfm` configuration parameter ¹)
- ◆ Specify the format and characteristics for the records in a new dataset

Using **recfm** without an argument resets the format IFT uses for future dataset allocations to the record format value in IMCONFIG, the default IFT configuration file. That value can be either the default supplied with the installation kit, or the value you entered during optional configuration.

Syntax **recfm** [*<format>*]

Argument *<format>*

The record format you want to use for future dataset allocations. Can be:

F	FBSA	UA	VBSA
FA	FBSM	UM	VBSM
FB	FM	V	VBM
FBA	FS	VA	VM
FBM	FSA	VB	VS
FBS	FSM	VBA	VSA
	U	VBS	VSM

1. You can set this configuration parameter when you perform optional configuration either through the installation procedure, or through the IFT management utility.

These formats are made up of the following elements:

Character	Meaning
F	Fixed record length
V	Variable record length
U	Undefined record length
B	Blocked records
S	Spanned records
A	Records with ISO/ANSI characters for printing
M	Records containing machine control characters for printing

Usage Notes Keep the following in mind:

Transferring RECFM=U Datasets on Mainframe Systems

When you transfer RECFM=U datasets, IFT transfers the RECFM=U records as a stream of data without preserving the internal blocking structure of the dataset.

This fact has significance for partitioned datasets. When you transfer a member of a PDS defined with RECFM=U to another dataset, the internal structure of the PDS member is not preserved. Thus, although you can transfer load modules, the transferred data cannot be used as a load module.

Example None

Related Topics

- ◆ **blksize (OS/390 and z/OS):** page 4-5
- ◆ **imbufno (OS/390 and z/OS):** page 4-35
- ◆ **imcylinder (OS/390 and z/OS):** page 4-40
- ◆ **imdirblocks (OS/390 and z/OS):** page 4-44
- ◆ **imfilenumber (OS/390 and z/OS):** page 4-46
- ◆ **lrecl (OS/390 and z/OS):** page 4-79
- ◆ **recfm (OS/390 and z/OS):** page 4-99
- ◆ **volcount (OS/390 and z/OS):** page 4-113
- ◆ **volume (OS/390 and z/OS):** page 4-115

replace (OS/390 and z/OS)

Purpose Use **replace** to enable or disable file replacement mode for subsequent transfers.

Syntax **rep**[lace] [**ON** | **OFF**]

Arguments **ON**

Enables file replacement mode for subsequent transfers.

OFF

Disables file replacement mode for subsequent transfers.

Usage Notes Keep the following points in mind:

Setting File Replacement Mode

When you first issue **replace ON** during a session, **replace** clears any IFT volume settings in any mainframe system involved in a connection.

Using the **replace** Command Without an Argument

You can use **replace** without an argument. The **replace** command without an argument toggles file replacement mode on and off. On all hosts, file replacement mode is off by default when you first invoke the client.

The first time you issue **replace** without an argument, you enable file replacement mode for the following transfer. Each subsequent **replace** that you issue without an argument, reverses the current state of file replacement mode.

File Replacement Mode with a **get** or **put** Command

If IFT is in file replacement mode and the target already exists, a **get** or **put** command overwrites the target. To save a copy of the earlier version of the target, take the following steps:

- ◆ Use a different name for the new target file you are about to create
- ◆ Give the existing file on the target system a different name
- ◆ Turn replacement mode off

If IFT is not in file replacement mode and the target already exists, **get** or **put** generates the following error:

```
IM005E: Error: File already exists
```

File Replacement and Verbose Modes

When you turn replacement mode on while you are in verbose mode, you see the message:

```
Replace mode is ON
```

When you turn replacement mode off while you are in verbose mode, you see the message:

```
Replace mode is OFF
```

File Replacement Mode and Mainframe Systems

When you replace a mainframe host dataset, the dataset retains the original attributes. File replacement on a mainframe system can fail if the used space exceeds the allocated space in the target.

File Replacement Mode and the volume Command

If file replacement mode is on, IFT ignores any setting you make with **volume**.

Displaying Current Settings

To find out whether IFT is in file replacement mode, enter **status**.

Example The following is an example of **replace ON** and **replace OFF**:

```
replace ON
Command: replace
Replace mode is ON

replace OFF
Command: replace
Replace mode is OFF
```

Related Topics

- ◆ **blksize (OS/390 and z/OS):** page 4-5
- ◆ **imbufno (OS/390 and z/OS):** page 4-35
- ◆ **imcylinder (OS/390 and z/OS):** page 4-40
- ◆ **imdirblocks (OS/390 and z/OS):** page 4-44
- ◆ **imfilenumber (OS/390 and z/OS):** page 4-46
- ◆ **imrelease (OS/390 and z/OS):** page 4-54
- ◆ **likedsn (OS/390 and z/OS):** page 4-75
- ◆ **recfm (OS/390 and z/OS):** page 4-99
- ◆ **volcount (OS/390 and z/OS):** page 4-113
- ◆ **volume (OS/390 and z/OS):** page 4-115

status

Purpose Use **status** to display the current settings for data transfers.

Syntax **stat** [us]

Arguments None

Usage Notes The output of **status** shows the following status settings:

Status Item	Meaning
Remote connection to:	The name of the remote server connection.
Remote server type:	Remote server type (OS/390, z/OS, UNIX, or Windows host).
Verbose messaging:	Whether or not verbose messaging is turned ON or OFF.
Session logging:	Whether or not window output is being sent to a log file.
Replace mode:	Whether replacement mode is ON or OFF. Important: If replacement mode is ON and the target already exists, get or put overwrites the target.
Data translation:	Whether or not transferred characters are to be translated and how translation is to take place.
Translation performed at:	Whether translation is to be done at the local or remote host.
Send translation table:	Specified translation file or table for files to be sent.
Recv translation table:	Specified translation file for files or table to be received.
Text data uppercasing:	Whether or not text is to be converted to all uppercase letters (ON or OFF).
Save trailing blanks:	Whether or not trailing blanks in an OS/390 or z/OS mainframe source are to be saved.
Record LF appending:	Whether or not record linefeeds are to be appended to an OS/390 or z/OS mainframe source.
Target MVS volume:	The volser used for new mainframe datasets.
Primary cylinders:	The primary cylinder allocation for the target.
Secondary cylinders: ^a	If the remote server is an OS/390 or z/OS system, the secondary cylinder allocation for the target.

Status Item	Meaning
Recfm set to: ^a	If the remote server is an OS/390 or z/OS system, the recfm setting.
Blksize set to: ^a	If the remote server is an OS/390 or z/OS system, the blksize setting.
Lrecl set to: ^a	If the remote server is an OS/390 or z/OS system, the lrecl setting.
Imbufno set to: ^a	If the remote server is an OS/390 or z/OS system, the imbufno setting.
Imfilenumber set to: ^a	If the remote server is an OS/390 or z/OS system, the imfilenumber setting.
Imdirblocks set to: ^a	If the remote server is an OS/390 or z/OS system, the imdirblocks setting.
Imrelease mode: ^a	Whether imrelease mode is set.
Likedsn set to: ^a	What the current value of likedsn is (if an OS/390 or z/OS mainframe is involved in the transfer).
Transfer RDWs: ^a	Whether RDWs are to be copied (YES or NO).
End-Of-Line conversion:	If host A is a Windows system, whether or not end-of-line sequences is converted (ON or OFF).

a. This setting is shown only if the remote or local server is an OS/390 or z/OS system.

If a particular item is set to the value in the configuration file, **status** displays the value followed by the word (Default). For example:

```
Primary cylinders          2          (Default)
```

If a particular item is set to a value by a CLI command you gave, **status** displays the value followed by the words (User Specified). For example:

```
Primary cylinders          3          (User Specified)
```

There are other values (such as **likedsn** value or target mainframe volume) that you need to specifically set for transfers. If you have not specified them yet, IFT displays:

```
(Not Specified)
```

- Examples**
1. The following example is **status** output when an IFT client is connected to a mainframe server:

```
IM:AMEERCN> status
Command Name: status

Remote Connection To: AMEERCN
Remote Server Type:      MVS
Verbose Messaging:      ON
Session Logging:        OFF
Replace Mode:           OFF
Data Translation:       ON
Translation performed at: Remote host
Send Translation Table:  AimEim.imxlate
Recv Translation Table:  EimAim.imxlate
Text Data Uppercasing:  OFF
Saving Trailing Blanks: OFF
Wrap mode:              OFF
Record LF Appending:    ON
Target MVS volume:      (not_specified)
Likedsn set to:         (not_specified)
Unit set to:            SYSDA (default)
Primary Cylinders:      2      (default)
Secondary Cylinders:    5      (default)
Recfm set to:           VB      (default)
Blksize set to:         unset
Lrecl set to:           4092    (default)
Imbufno set to:         2      (default)
Imfilenumber set to:    1      (default)
ImdirBlocks set to:     20     (default)
Transfer RDWS:          OFF     (default)
Imrelease set to:       OFF     (default)
End-Of-Line conversion: OFF     (default)
```

2. The following example is **status** output when the IFT is connected to an open systems/Windows server:

```
IM:ajje> status
Command: status

Remote Connection To:  ajje
Remote Server Type:    Open Systems
Verbose Messaging:    ON
Session Logging:      OFF
Replace mode:         OFF
Wrap mode:            OFF
Data Translation:     OFF (image/binary)
End-Of-Line conversion: OFF
```

- Related Topics**
- ◆ **blksize (OS/390 and z/OS):** page 4-5
 - ◆ **imaddlf (OS/390 and z/OS):** page 4-32
 - ◆ **imbufno (OS/390 and z/OS):** page 4-35
 - ◆ **imcrlf:** page 4-37
 - ◆ **imcylinder (OS/390 and z/OS):** page 4-40
 - ◆ **imdirblocks (OS/390 and z/OS):** page 4-44
 - ◆ **imfilenumber (OS/390 and z/OS):** page 4-46
 - ◆ **imrdw (OS/390 and z/OS):** page 4-50
 - ◆ **imrecvx:** page 4-52
 - ◆ **imrelease (OS/390 and z/OS):** page 4-54
 - ◆ **imsendx:** page 4-55
 - ◆ **imtrail (OS/390 and z/OS):** page 4-57
 - ◆ **imwrap (Open Systems):** page 4-59
 - ◆ **imxlate:** page 4-61
 - ◆ **likedsn (OS/390 and z/OS):** page 4-75
 - ◆ **lrecl (OS/390 and z/OS):** page 4-79
 - ◆ **lsp (OS/390 and z/OS):** page 4-80
 - ◆ **recfm (OS/390 and z/OS):** page 4-99
 - ◆ **replace (OS/390 and z/OS):** page 4-101
 - ◆ **unit (OS/390 and z/OS):** page 4-107
 - ◆ **upper:** page 4-110
 - ◆ **verbose:** page 4-112
 - ◆ **volcount (OS/390 and z/OS):** page 4-113
 - ◆ **volume (OS/390 and z/OS):** page 4-115

unit (OS/390 and z/OS)

Purpose Use **unit** for new target OS/390 or z/OS mainframe datasets. The **unit** command can:

- ◆ Override the default unit value you set during optional configuration
- ◆ Specify a unit for subsequent transfer to a target mainframe system

Using **unit** without an argument returns the default unit value to that you set in the IFT configuration file as part of basic configuration during installation.

Syntax **unit** [*<unitname>*]

Argument *<unitname>*

The name of the new default unit you want to use.

Usage Notes The value you use for *<unitname>* may be a device number, or a device type.

Example The commands in the following example:

- ◆ Set the default unit to LABWORK
- ◆ Issue a series of commands using LABWORK as the default
- ◆ Issue unit without an argument to return the unit default to that specified in the default_unit configuration parameter

```
unit LABWORK  
Command: unit LABWORK  
  
Unit set to LABWORK  
.  
.  
.  
unit
```

Related Topics

- ◆ **get**: page 4-23
- ◆ **put**: page 4-90
- ◆ **volcount (OS/390 and z/OS)**: page 4-113
- ◆ **volume (OS/390 and z/OS)**: page 4-115

unitcount (OS/390 and z/OS)

Purpose Use **unitcount** when one of the client host platforms is OS/390 or z/OS. The **unitcount** command specifies the number of units to be allocated for a new OS/390 or z/OS dataset allocation. The **unitcount** command functions as the unit count sub-parameter of the JCL DD statement `UNIT= parameter`.

The **unitcount** command is particularly useful in allocating new, multi-volume, DASD datasets when you are specifying an esoteric unit name.

More Information: Refer to *unit (OS/390 and z/OS)* on page 4-107.

Syntax **unitcount** | **unitcnt** [*<unit_count>*]

Argument *<unit_count>*

The number of disk volumes that the target dataset requires. The values you can specify are one through 59. If *<unit_count>* is greater than the number of devices defined in the unit you are using, the command fails.

To set the number of disk volumes that a target dataset requires to one, issue **unitcount** without a parameter.

There is no argument value that resets the value for **unitcount** to "unset."

Usage Notes Keep the following points in mind:

Alternate Command Name

You can also use **unitcnt** as a synonym for **unitcount**.

Effect of the `unitcount` Command Argument

After you specify a disk volume count, this value remains in effect until:

- ◆ You enter `unitcount` with another `<unit_count>` value of anywhere between 1 and 59
- ◆ You issue `unitcount` without a parameter, which sets the volume count to one

Displaying Current Settings

Use `status` to display the current unit count.

Allocating a Multivolume DASD Dataset

To allocate a multivolume DASD dataset:

- ◆ Issue the `unit` command with an esoteric as a argument
- ◆ Issue the `unitcount` command with an argument greater than one (1)

Example

The following command is issued just before a file transfer that results in the allocation of a new multivolume dataset. By setting the unit count to 5, the user ensures that 5 disk volumes can be used to satisfy the allocation request:

```
unitcount 5
```

The following command resets the volume unit count to one for subsequent transfers:

```
unitcount
```

Related Topics

- ◆ `unit` (OS/390 and z/OS): page 4-107

upper

Purpose Use **upper** to set the transfer to text mode, and convert transferred text to all uppercase (capital) letters in subsequent transfers.

Syntax **up**[per] [**ON** | **OFF**]

Arguments **ON**

Enables uppercase conversion.

OFF

Disables uppercase conversion.

Usage Notes Keep the following points in mind:

Valid Usage

The **upper** command is only applicable when character translation (with **imxlate**) is enabled.

Using the upper Command Without an Argument

You can use **upper** without an argument. The **upper** command without an argument toggles uppercase conversion on and off. On all hosts, uppercase mode is disabled by default. The first time you issue **upper** without an argument, you enable uppercase conversion on any following transfers. Each subsequent **upper** that you issue without an argument reverses the state of automatic uppercase conversion.

Displaying Current Status

Use **status** to display the current setting.

Performance Considerations

If you want to translate text data to uppercase during a transfer, you will find it more efficient to do so through a translation table than to use the **upper** CLI command, particularly if the translation is done on a mainframe host. The IFT **upper** CLI command causes IFT to do single-byte translation for all platforms, and call the `toupper C` function to actually perform the uppercasing. This degrades performance.

Example If verbose messaging is turned on and you enter **upper**, the following information appears:

```
upper ON
```

```
Command: upper
```

```
All translated UNIX text data will now be UPPERcased
```

Related Topics

- ◆ **imaddlf (OS/390 and z/OS):** page 4-32
- ◆ **imcrlf:** page 4-37
- ◆ **imtrail (OS/390 and z/OS):** page 4-57
- ◆ **imwrap (Open Systems):** page 4-59

verbose

Purpose Use **verbose** to enable or disable verbose mode. When you enable verbose mode, IFT echoes commands, and displays command status and communication settings on the window. When you disable verbose mode, IFT no longer echoes commands, or displays detailed command status and communication settings on the window.

Syntax **verb**[ose] [**ON** | **OFF**]

Arguments **ON**

Enables verbose mode.

OFF

Disables verbose mode.

Usage Notes Keep the following points in mind:

Using the verbose Command Without an Argument

You can use **verbose** without an argument to toggle verbose mode on and off. On all hosts, verbose messaging is enabled by default. The first time you issue **verbose** without an argument, you disable verbose mode. Each subsequent **verbose** that you issue without an argument reverses the state of verbose mode.

Displaying Current Settings

Use **status** to display the current setting of verbose mode.

Example None

Related Topics **status**: page 4-103

volcount (OS/390 and z/OS)

Purpose Use **volcount** when any transfer command results in a new or target multivolume tape dataset. The **volcount** command specifies the maximum number of volumes that may be used for a target dataset. The command is similar to the volume-count subparameter of the JCL DD VOLUME parameter.

Syntax **volcount** | **volcnt** [*<volume_count>*]

Argument *<volume_count>*

The maximum number of volumes that the target dataset requires. The values you can specify are one through 255. If *<volume_count>* is greater than the number of devices defined in the unit you are using, the command fails.

Usage Notes Keep the following points in mind:

Alternate Command Name

You can also use **volcnt** as a synonym for **volcount**.

Effect of the volcount Command Argument

After you specify a volume count, this value remains in effect until you either enter **volcount**:

- ◆ With another *<volume_count>*
- ◆ Without an argument

If you enter **volcount** without an argument, IFT deletes the current value for *<volume_count>*, and returns to the mainframe system defaults.

Any file transfers issued after this change uses the mainframe system defaults. Then if you issue **status**, the volume count is listed as (Unset).

Displaying Current Settings

Use **status** to display the current volume count.

Example The following command is issued just before a file transfer that results in the allocation of a new multivolume dataset. By setting the volume count to 20, the user ensures that up to 20 volumes can be used to satisfy the allocation request:

```
volcount 20
```

If subsequent transfers can not create multivolume datasets, you should clear the `volume_count` value by issuing the **volcount** command without an argument:

```
volcount
```

Related Topics

- ◆ **blksize (OS/390 and z/OS):** page 4-5
- ◆ **imbufno (OS/390 and z/OS):** page 4-35
- ◆ **imcylinder (OS/390 and z/OS):** page 4-40
- ◆ **imdirblocks (OS/390 and z/OS):** page 4-44
- ◆ **imfilenumber (OS/390 and z/OS):** page 4-46
- ◆ **imrdw (OS/390 and z/OS):** page 4-50
- ◆ **imrelease (OS/390 and z/OS):** page 4-54
- ◆ **likedsn (OS/390 and z/OS):** page 4-75
- ◆ **lrecl (OS/390 and z/OS):** page 4-79
- ◆ **lsp (OS/390 and z/OS):** page 4-80
- ◆ **recfm (OS/390 and z/OS):** page 4-99
- ◆ **unit (OS/390 and z/OS):** page 4-107
- ◆ **volcount (OS/390 and z/OS):** page 4-113
- ◆ **volume (OS/390 and z/OS):** page 4-115

volume (OS/390 and z/OS)

Purpose Use **volume** to specify the volser for new OS/390 or z/OS target datasets. The **volume** command allows you to specify one and only one default volser of the volume on which you want to dynamically allocate a new target mainframe host dataset.

Syntax **vol**[ume] [*<volser>*]

Argument *<volser>*

The six-character serial number of the volume on a mainframe system on which you want to dynamically allocate the target mainframe dataset.

On dataset allocation, IFT treats volume with a *<volser>* argument as allocating a volume with the generic unit name of "SYSDA" unless you have changed the unit as part of basic configuration during installation, or through the **unit** command.

Usage Notes Keep the following points in mind:

Command Application to Local Hosts

The **volume** command applies to new dataset allocations on either the local or remote host.

Limits of the Command

The **volume** command only applies to new dataset allocations. IFT ignores **volume** commands applied to existing datasets. If you use **volume** to specify a volser that is not included in the unit (SYSDA by default), **volume** is ignored.

Displaying Current Settings

Use **status** to display the current setting.

Using the volume Command Without an Argument

Using **volume** without a *<volser>* argument clears the volume serial number you specified as an argument in a previous **volume** command.

Example If verbose messaging is turned on and you enter **volume**, the following information appears:

```
volume SIVA01  
Command: volume SIVA01  
  
Volume set to SIVA01
```

Related Topics

- ◆ **blksize (OS/390 and z/OS):** page 4-5
- ◆ **imbufno (OS/390 and z/OS):** page 4-35
- ◆ **imcylinder (OS/390 and z/OS):** page 4-40
- ◆ **imdirblocks (OS/390 and z/OS):** page 4-44
- ◆ **imfilenumber (OS/390 and z/OS):** page 4-46
- ◆ **imrdw (OS/390 and z/OS):** page 4-50
- ◆ **imrelease (OS/390 and z/OS):** page 4-54
- ◆ **likedsn (OS/390 and z/OS):** page 4-75
- ◆ **lrecl (OS/390 and z/OS):** page 4-79
- ◆ **lsp (OS/390 and z/OS):** page 4-80
- ◆ **recfm (OS/390 and z/OS):** page 4-99
- ◆ **replace (OS/390 and z/OS):** page 4-101
- ◆ **unit (OS/390 and z/OS):** page 4-107
- ◆ **volcount (OS/390 and z/OS):** page 4-113

Error and Event Messages

This appendix discusses IFT error and event messages. Topics include:

- Overview of Event and Error MessagesA-2
- Event Files on AIX, HP-UX, Solaris, and Windows Systems.....A-6
- IFT Event Message SummaryA-7
- Event Messages in Cancelled Mainframe JobsA-27

Overview of Event and Error Messages

When you use IFT in file transfer operations, you can receive either event or error messages at a number of points. You or your EMC Customer Support personnel can direct the dialog from an IFT session to log files for later review.

Refer to the *InfoMover File Transfer 3.6 System Administrator's Guide* for a description of IFT log files, and how to use them.

Event Message Format

All event messages have a format similar to Figure A-1:

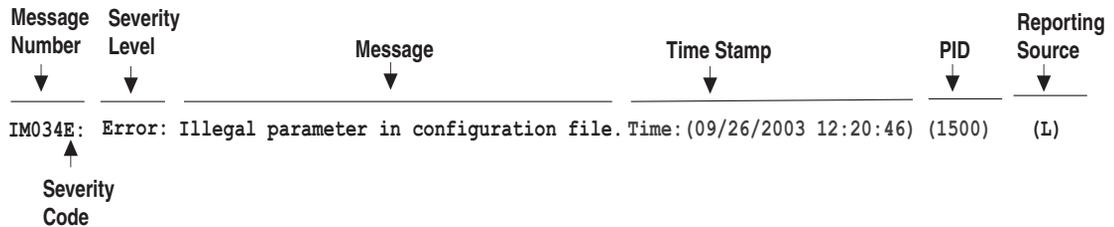


Figure A-1 Message Format

where:

Message Number

The number associated with a particular message.

Severity Code

A single character that explains the type of message being received.

Severity Level

A string that is an expansion of the severity code.

Table A-1 shows the severity codes and levels you can receive:

Table A-1 Message Severity Codes and Severity Levels

Severity Code	Severity Level	Description
I	Information	Provides information about a file transfer.
W	Warning	Usage or user error.
E	Error	Execution problem. Most frequently caused by a bad configuration or user error.
S	Serious Error	Problem related to file system resources, login privileges, or I/O (input/output). Results in a file transfer failure. File transfer must be resubmitted. InfoMover File Transfer software resets itself.
F	Fatal	Unrecoverable error. Causes the InfoMover File Transfer software to close. You may have to reformat the IFT transfer devices.

Message

A short message explaining the event.

Time Stamp

The time the event was recorded.

Reporting Source

Message reporting source. Client (l), Initiator (i), or Connector (c).

Looking Up an Error Code

You can look up an error code from the OS/390, z/OS, UNIX, or Windows system on which the IFT configuration file resides, as listed in *IFT Event Message Summary* on page A-7.

More Information: Refer to the *InfoMover File Transfer System Administrator's Guide* for more about errors and error codes.

Verbose Messages

If the `verbose_server_messages` configuration parameter in the IFT configuration file is set to **y** (the default), you receive more detailed event messages.

What Does Verbose Mean?

More Information: Refer to *InfoMover File Transfer System Administrator's Guide* for a description of `verbose_server_messages` and other configuration parameters.

The information you receive in a verbose event message includes:

- ◆ IFT message (from Table A-2 on page A-7)
- ◆ Initiator server name ¹
- ◆ Connector server name ¹
- ◆ Source and target filenames (for transfers)
- ◆ Directory name (if the specified directory was nonexistent)
- ◆ Job name (if you are on a mainframe client)
- ◆ PID (if you are on a UNIX client)
- ◆ Phrases “open command”, “close command” and “quit command” (if the event involved one of these respective commands)
- ◆ Phrase “translation file=<filename>” (if the event involved a translation file)
- ◆ Date, time, and severity level

Example A-1 is a verbose server event message on a UNIX system:

Example A-1 UNIX Verbose Event Message

```
get EMCIM10 /usr/dtest/file1
Data will be translated using (Default)
Transferring file EMCIM10
IM00I Information: File transfer successfully completed.
Source = EMCIM10,
Target = /usr/dtest/file1
Time: (09/18/2004 13:42:55) (1)
Elapsed=29, FileSize=147465112, KbytePerSec=5085
```

1. Refer to *IFT Server* on page 1-3 for an explanation of initiators and connectors.

What is Nonverbose?

If the `verbose_server_messages` configuration parameter in the IFT configuration file is set to `n`, you receive less detailed messages that only include the:

- ◆ Event message (from Table A-2 on page A-7)
- ◆ Date, time, and severity level

More Information: Refer to *InfoMover File Transfer System Administrator's Guide* for a description of the configuration parameters.

Example A-2 is a nonverbose server event message on a UNIX system:

Example A-2 UNIX Non Verbose Event Message

```
get EMCIM10 /usr/dtest/file1
Data will be translated using (Default)
Transferring file EMCIM10
IM00I Information: File transfer successfully completed.
Time: (09/18/2004 13:42:55) (1)
```

GUI-Based Messages

Whenever you receive an error while using the Windows or UNIX GUI, you see an error or event message window, as shown in Figure A-2:

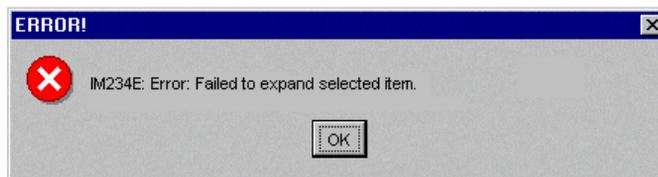


Figure A-2 InfoMover File Transfer Error Window

Event Files on AIX, HP-UX, Solaris, and Windows Systems

As events happen, IFT writes information about those events to log files on AIX, HP-UX, Solaris, and Windows systems.

AIX, HP-UX, and Solaris Events

On AIX, HP-UX, and Solaris systems, IFT logs any of the following types of events to the `/var/adm/imMessages` file:

- ◆ Server startup
- ◆ Server shutdown

The `imMessages` file is never deleted. Therefore, it can grow quite large over the course of time. You may want to monitor the size of this file and, if `imMessages` grows too large, delete it.

Windows Events and the Windows Event Viewer

On Windows systems, IFT logs any of the following types of events to the Windows Event Log:

- ◆ Server startup
- ◆ Server shutdown
- ◆ Error messages (such as failed **get** and **put** commands)

You can view these messages using the Windows Event Viewer.

However, you must periodically clear the log, either when the log reaches a certain size, or when a message notifies you that the log is full.

IFT Event Message Summary

Table A-2 provides explanations of IFT event messages and suggests remedies, as appropriate:

Table A-2 InfoMover File Transfer Event Messages

Message	Explanation	Suggested Remedy
IM000I: Informational: File transfer successfully completed.	The file was successfully transferred.	None required.
IM001E: Error: File does not exist.	The requested file or directory does not exist in the specified location.	Check use of commands and spelling of filenames and directory names.
IM002E: Error: Requested file or directory name is not valid.	The file or directory name is longer than the maximum allowed by IFT, or one or more characters in the name is invalid.	Check the spelling and length of the file or directory name. If the name does exist as you specified it, check to make sure that your permissions are set correctly.
IM003E: Error: Failed to open file.	The current IFT user does not have read and/or write access to the requested file or directory.	Use a different username or work with your system administrator to set up the proper permissions for the file or directory.
IM004S: Serious Error: Failed to create file.	Either: <i>Open Systems:</i> The current IFT user does not have write access to the requested file or directory. <i>OS/390 and z/OS Systems:</i> The IFT software cannot allocate space for and/or completely write the file due to lack of space on the target volume or file system.	Use a different username or work with your system administrator to set up the proper permissions for the directory. Specify an alternative volume name or file system or request that your system administrator make more space available to you on your current one.
IM005E: Error: File already exists.	The specified file already exists. This operation cannot overwrite the existing file unless replacement mode is enabled.	Specify a different filename. If the operation is a get or put command, issue a replace ON command before attempting the transfer again if you want to overwrite the existing file.

Table A-2 InfoMover File Transfer Event Messages (continued)

Message	Explanation	Suggested Remedy
<p>IM006S: Serious Error: File is already being used by another process.</p>	<p>The file location that you specified for a file transfer is already in use and locked by another IFT server (or other software process).</p> <p>You may receive this error if you are using the same target file for repeated (one after another) transfers.</p> <p>This is a result of host file management processing the file beyond the end of the IFT transfer. Several seconds delay may be required between transfers to the same target file.</p>	<p>Resubmit the file transfer request after the target location resource is free or unlocked, or use a different target filename.</p>
<p>IM007S: Serious Error: Device I/O failure.</p>	<p>A read or write of one of the transfer devices, requested by the IFT software, failed or yielded unexpected results. The file transfer is incomplete.</p> <p>You may receive this error if you are moving a file containing control characters in text translation mode.</p>	<p>Verify that your transfer devices and your host-Symmetrix connection are functioning properly. Use the following set of escalating procedures to verify the devices:</p> <ol style="list-style-type: none"> 1. Shut down all servers and run the Query option in the IFT Management Utility. If the Query is successful, resubmit the transfer. If the Query was successful, but the transfer continues to fail and IFT keeps issuing the IM007S message, go to step 2. 2. Shut down the servers and reformat the devices. If the format is unsuccessful, proceed to step 3. If the format is successful, resubmit the file transfer. 3. If step 2 fails, use the following steps: <ol style="list-style-type: none"> a. Stop all of the servers b. Rebuild the database c. Rebuild the configuration file d. Format the devices <p>If these steps are successful, then resubmit the file transfer.</p> 4. If step 3 fails, call EMC Customer Service.

Table A-2 InfoMover File Transfer Event Messages (continued)

Message	Explanation	Suggested Remedy
IM008S: Serious Error: I/O failure unlocking transfer device.	An unlock of one of the transfer devices, requested by the IFT software, failed or yielded unexpected results. The file transfer is incomplete.	Verify that your IFT transfer devices and your host connections are working properly. To verify the transfer devices, run the Query option in the IFT Management Utility. Resubmit any incomplete file transfers.
IM009S: Serious Error: I/O failure locking transfer device.	A lock of one of the transfer devices, requested by the IFT software, failed or yielded unexpected results. The file transfer is incomplete.	Verify that your IFT transfer devices and your host connections are working properly. Resubmit any incomplete file transfers.
IM010S: Serious Error: Unable to open transfer device.	An open of one of the transfer devices, requested by the IFT software, failed or yielded unexpected results. The file transfer is incomplete and all tables are cleared.	Verify that your IFT transfer devices and your host connections are working properly. To verify the transfer devices, run the Query option in the IFT Management Utility. Resubmit any incomplete file transfers.
IM011S: Serious Error: Unable to retrieve device information from transfer device.	An attempt to read inquiry data or the device capacity of one of the transfer devices, requested by the IFT software, failed or yielded unexpected results.	Verify that your IFT transfer devices and your host connections are working properly. To verify the transfer devices, run the Query option in the IFT Management Utility.
IM012S: Serious Error: Failed to close device(s) or file(s).	IFT is unable to close the specified file or raw device properly.	Shut down the process or server on which the error occurred. Verify that your host environment is functioning correctly. To verify the transfer devices, run the Query option in the IFT Management Utility. Restart the process or server.
IM013S: Serious Error: There was an error sending data to the server.	The IFT client is experiencing problems sending data to the IFT server through the designated socket or across the designated device.	<p>If the client is running on open systems, ensure that:</p> <ul style="list-style-type: none"> You are requesting a connection to the correct socket (as designated by the port number in the IFT configuration file) Your open systems network is working properly <p>If the client is running on OS/390 or z/OS, make sure that your IFT master device is functioning properly. Also verify the general validity of the I/O path.</p> <p>If you are using the GUI, verify that both the local and the remote server information is valid and the network connection to both servers is valid.</p>

Table A-2 InfoMover File Transfer Event Messages (continued)

Message	Explanation	Suggested Remedy
IM014S: Serious Error: There was an error receiving data from the server.	The IFT client received invalid or incomplete data from the IFT server through the designated socket or across the designated device.	<p>If the client is running on open systems, ensure that:</p> <ul style="list-style-type: none"> You are requesting a connection to the correct socket (as designated by the port number in the IFT configuration file) Your open systems network is working properly <p>If the client is running on OS/390 or z/OS, make sure that your IFT master device is functioning properly. Also check the general I/O path validity.</p> <p>If you are using the GUI, verify that both the local and the remote server information is valid and the network connection to both servers is valid.</p>
IM015S: Serious Error: There was an error receiving data from the client.	The IFT server initiator received invalid or incomplete data from the IFT client through the designated socket or across the designated device.	<p>If the client is running on open systems, ensure that:</p> <ul style="list-style-type: none"> You are requesting a connection to the correct socket (as designated by the port number in the IFT configuration file) Your local open systems network is working properly <p>If the client is running on OS/390 or z/OS, make sure that your IFT master device is functioning properly. Also check the general I/O path validity.</p>
IM016S: Serious Error: There was an error sending data to the client.	The IFT server initiator is experiencing problems communicating with the IFT client through the designated socket or across the designated device.	<p>If the client is running on open systems, ensure that:</p> <ul style="list-style-type: none"> You are requesting a connection to the correct socket (as designated by the port number in the IFT configuration file) Your local open systems network is working properly. <p>If the client is running on OS/390 or z/OS, make sure that your IFT master device is functioning properly. Also check the general I/O path validity.</p>

Table A-2 InfoMover File Transfer Event Messages (continued)

Message	Explanation	Suggested Remedy
IM017S: Serious Error: Connector server reported an error.	The connector server (IFT server on host B) is flagging an error to the initiator server (IFT server on host A) through the IFT protocol. Refer to Figure 1-1 on page 1-4 for an illustration of host A and host B.	The IFT server on host A interprets the error and notifies you with a subsequent message.
IM018F: Fatal Error: Incompatible InfoMover File Transfer structure version! You must upgrade all host nodes with the latest version and RE-RUN an InfoMover File Transfer format on your devices.	Your devices are formatted with a different version of the IFT software than the current version.	Before executing this version of IFT, you must load all host nodes with the current version and rerun an IFT format for your devices.
IM019F: Fatal Error: InfoMover File Transfer client is terminating due to error.	Either a Serious Error or a Fatal Error caused the client to be unable to continue.	Follow the suggestion for the accompanying error message. Restart the IFT client when conflicts are resolved.
IM020S: Serious Error: Transfer halted due to client termination.	The IFT client is terminated in the middle of a transfer. The transfer is stopped and the target file is deleted.	Restart the IFT client. Resubmit the transfer.
IM021W: Warning: InfoMover File Transfer Management Utility option failed.	The selected option in the IFT Management Utility is unable to be performed due to an error.	Follow the suggested remedy for the accompanying error.

Table A-2 InfoMover File Transfer Event Messages (continued)

Message	Explanation	Suggested Remedy
<pre>IM022F: Fatal Error: Remote server not responding -- connection closed.</pre>	<p>The IFT server on host B has not updated the IFT protocol tables within the allotted time-out period. IFT assumes that the server is inactive or in trouble.</p>	<p>Investigate the condition of the IFT server on host B.</p> <p>Refer to the discussions and illustrations in Chapter 1 of the <i>InfoMover File Transfer User's Guide</i> for a further explanation of hosts A and B.</p> <p>If you are using IFT in an extremely busy environment, one in which demand for host CPU time is high and/or the Symmetrix system is rarely idle, you can encounter occasional transfer time-outs. In such a situation, you may want to adjust the value of the <code>command_timeout</code> configuration parameter in all IFT configuration files. The default <code>command_timeout</code> value is:</p> <ul style="list-style-type: none"> • 300 on OS/390 and z/OS • 1200 on open systems <p>You may need to increase <code>command_timeout</code> value if you get this error message repeatedly.</p>
<pre>IM023F: Fatal Error: Communication timeout -- connection closed.</pre>	<p>The connection between an IFT client and an IFT server is closed for one of the following reasons:</p> <ol style="list-style-type: none"> 1. The IFT client prompt has been idle for over 15 minutes. 2. The IFT server on host A has detected a time-out, indicating problems with the IFT server on host B. 3. You requested a connection to a nonexistent IFT server. 4. The system timed out while waiting for a tape mount on OS/390 and z/OS. 5. You are using IFT in an extremely busy environment, one in which demand for host CPU time is high and/or the Symmetrix system is rarely idle. 	<p>Evaluate the circumstances under which the time-out occurred and, based on the explanations in the column to the left, take the appropriate action(s):</p> <ol style="list-style-type: none"> 1. Log on again to the IFT server on host A. Refer to Figure 1-1 on page 1-4 for an illustration of host A and host B. 2. Verify the functionality of the Symmetrix system and all hosts involved in the connections and file transfers. One way to verify the transfer devices is to run the Query option in the IFT Management Utility. Restart the process or server. 3. Check spelling and capitalization in the IFT server name. Verify that the requested IFT server is active. 4. Verify that all tape mount requests are satisfied. 5. You may need to increase the value of the <code>command_timeout</code> configuration parameter value if you get this error message repeatedly.

Table A-2 InfoMover File Transfer Event Messages (continued)

Message	Explanation	Suggested Remedy
IM024S: Serious Error: The local and remote servers are not able to communicate.	The IFT server on host A and the IFT server on host B disagree as to what piece of the file should be transferred next. Refer to the figures in Figure 1-1 on page 1-4, for an illustration of host A and host B.	Verify that your transfer system is working correctly. You may need to clean the connection table using the Clear Connections option in the IFT Management Utility. In some cases, you may need to close all connections, stop all servers, and reformat the transfer devices.
IM025S: Serious Error: Failed to obtain lock.	IFT cannot access the specified file or device.	Make sure that IFT has access to the specified file or device. Make sure that another IFT session has not locked the specified file or device. To verify the transfer devices, run the Query option in the IFT Management Utility.
IM026S: Serious Error: Failed to release file descriptor lock.	IFT cannot unlock the specified file.	Close the affected connection. If necessary, clean the connection slot, using the Clear Connections option in the IFT Management Utility.
IM027S: Serious Error: Failed to properly clean table.	IFT cannot properly clean the specified control table. This can result in a hung connection.	If the specified table is the connection table, use the Clear Connections option in the IFT Management Utility. If the specified table is any other table, you may need to close all connections, stop all servers, and reformat the transfer devices.
IM028S: Serious Error: Insufficient space to allocate control structures.	There is not enough space available in memory for the necessary control structures.	If IFT is running on open systems, too many applications are running. Stop one and try starting IFT again. If IFT is running on OS/390 or z/OS, set the region size to a higher value.

Table A-2 InfoMover File Transfer Event Messages (continued)

Message	Explanation	Suggested Remedy
IM029S: Serious Error: Failed to connect to server.	<p>If the client receives the message, one of the following may be the cause:</p> <ul style="list-style-type: none"> • If the client is running on an open systems host, InfoMover File Transfer cannot initiate or connect to the specified socket. • If the client is running on a mainframe host, either the client cannot reserve a PID or the local server does not acknowledge the connection request. <p>Note that an attempt to start a client when there is no server running will also generate this message.</p> <p>If the server receives the message, the server is not able to bind to the socket.</p> <p>An example of this is an attempt to start two servers using the same IP address and port number for both servers.</p> <p>Another example of this is an attempt to start a server using a port number that is being used by another application, not necessarily IFT or even an EMC product.</p>	<p>When the client receives the message, take one of the following steps, as appropriate:</p> <ul style="list-style-type: none"> • If the client is running on an open systems host, ensure that you are requesting a connection to the correct socket (as designated by the port number in the IFT configuration file). Use a different socket or wait for the socket to be freed. • If the client is running on a mainframe, host, ensure that at least one PID is available and ensure that the local initiator server is running. • If you are using the GUI, verify that both the local and remote server information is valid and the network connection to both servers is valid. <p>When the server receives the message, change to an unused port number.</p>
IM029I: Informational: Failed to connect to server.	<p>The remote server did not respond to the connection request as expected. A retry takes place.</p>	None.
IM030E: Error: Security validation failed -- access denied.	<p>This error occurs in the following circumstances:</p> <ul style="list-style-type: none"> • Your username is not authorized on the host on which the IFT server is running. • You entered an incorrect password when attempting to connect with the host. The maximum length of a client password is 16 characters. • Your username cannot be granted the requested access to a file, dataset, or directory. 	<p>Check spelling, username, and user permissions. If necessary, verify with your system administrator that your requested actions are appropriate for your username.</p>

Table A-2 InfoMover File Transfer Event Messages (continued)

Message	Explanation	Suggested Remedy
IM031F: Fatal Error: Initialization of multitask environment failed.	<p>Mainframe systems only:</p> <ol style="list-style-type: none"> 1. "C" runtime fails to create OS/390 or z/OS multitasking environment. 2. The Language Environment (LE/370) Multi Tasking Facility (MTF) service, that IFT uses to schedule work, cannot locate the associated modules in the IFT load library. This failure occurs even when the load library is specified in the STEPLIB DD statement in the JCL PROCs used to start the initiator and connector address spaces in the OS/390 or z/OS IFT server. 	<ol style="list-style-type: none"> 1. Verify that the OS/390 or z/OS server region size conforms to the recommendations specified in this manual. In addition, view the failing server's OS/390 or z/OS Joblog for additional messages related to this failure. These messages further specify the reason for the failure. 2. Provide an additional EDCMTF DD statement to the start up JCL. This statement identifies the IFT load library to the LE/370 MTF service. Change the following DD statement to match the name of your IFT load library and then add it to the JCL for the #IMINTR and #IMCONNR members: <pre>//EDCMTF DD \ DSN=INFOMOVR.VvRrMm. \ LOADLIB, DISP=SHR</pre>
IM032F: Fatal Error: Schedule of parallel function failed.	<ol style="list-style-type: none"> 1. "C" runtime failed when scheduling an OS/390 or z/OS subtask. 2. You attempted to use the PDSE format for your IFT load library. The Language Environment (LE/370) Multi Tasking Facility (MTF) service does not support load modules in the PDSE format. 	<ol style="list-style-type: none"> 1. Verify that the OS/390 or z/OS server region size conforms to the recommendations specified in this manual. In addition, view the failing server's OS/390 or z/OS Joblog for additional messages related to this failure. These messages further specify the reason for the failure. 2. Verify that the IFT load library is a dataset with the PDS format, not the PDSE format.
IM033S: Serious Error: InfoMover File Transfer failed to properly interact with the operating system environment.	The specified system call failed and IFT cannot complete the current operation.	Make sure that all input filenames, directory names, and log in data are valid. Verify that your operating system version is supported by IFT.

Table A-2 InfoMover File Transfer Event Messages (continued)

Message	Explanation	Suggested Remedy
IM034W: Warning: Illegal parameter value in configuration file. The max_dir_datasets parameter is no longer supported. Parameter is ignored.	Your configuration file contains the max_dir_datasets parameter that is no longer needed by IFT. This parameter was indicated a maximum number of datasets that could be returned from a mainframe host by the dir CLI command. On IFT version 3.6 and higher, there is no limit on the number of datasets that can be returned.	To avoid receiving this warning, edit your configuration file to remove the max_dir_datasets parameter.
IM034E: Error: Illegal parameter value in configuration file.	The parameter is unrecognized or it is outside the valid boundaries.	Check the case and spelling of the parameter. Check the parameter value to make sure it is valid. Refer to Appendix B, <i>Advanced Configuration</i> , in the <i>InfoMover File Transfer Installation Guide</i> for a discussion about configuration file parameters.
IM035E: Error: Master device parameter value does not match configuration file parameter value.	The parameter value read from the master device is not the same as the value read from the configuration file.	Edit the parameter in the configuration file to match the value read from the master device. Stop all servers and use the 'Format' option in the IFT Management Utility to format the devices with the value specified in the configuration file.
IM036F: Fatal Error: Device is not properly formatted.	This error appears under the following circumstances: <ul style="list-style-type: none"> • The device has not been properly formatted for IFT use. • The device has been formatted with different translation files. • The configuration file has changed and the device has not been reformatted to match. 	Verify that the translation files in use were also used to format the transfer devices. Use the Format option in the IFT Management Utility to format the devices.
IM037E: Error: Host IP address either cannot be found or is invalid.	The host name given in the command line cannot be located by the client.	Check the spelling of the host name. Check the connectivity of the specified host to the current system.
IM038E: Error: Usage Error.	The arguments for this command are either missing or invalid.	Enter the help command to determine the necessary arguments for the command.

Table A-2 InfoMover File Transfer Event Messages (continued)

Message	Explanation	Suggested Remedy
IM039F: Fatal Error: Device ID must be 3 or 4 characters.	The supplied device number is either less than three characters or greater than four characters.	Verify that the device number you are using is correct and defined. Verify that the device is offline.
IM040S: Serious Error: Failed to archive old session log.	The system operation rename failed. The session log could not be moved to an archive file.	Use a different filename for the session log. Verify that the current IFT user has write permissions for the old session log.
IM041S: Serious Error: Unable to load translation file.	IFT cannot properly open or read the specified translation file.	Check the spelling of the filename. Check the permissions for the specified file and also check the accuracy of the directory path.
IM042E: Error: Neither imServer is capable of translation.	Only IFT servers running on open systems platforms are capable of translation.	Ensure that at least one IFT server involved in the transaction is running on an open systems platform.
IM043S: Serious Error: Reported device block size is incompatible with software.	The block size that is reported does not match the block size required by IFT.	Verify that you are specifying the correct transfer devices in your configuration file. Ensure that these are blocked devices. Refer to the Appendix B, <i>Advanced Configuration</i> , in the <i>InfoMover File Transfer Installation Guide</i> , for an explanation about how to specify configuration file parameters.
IM044S: Serious Error: File I/O failure. A system read of the file or write to the file was unsuccessful.	There are two possible causes: <ul style="list-style-type: none"> • There is an error in z/OS that is caused by an Sx37 type abend. This results from various end-of-volume or other out-of-space conditions. • There is an error caused by an incorrect logical record length (LRECL). 	If you receive an Sx37 type abend, adjust the blksize configuration parameter value if necessary. Issue the imcylinder command to increase the primary and/or secondary allocation parameters as necessary to ensure adequate space. If you did not receive an Sx37 type abend, the problem is an incorrect logical records. Increase the length specified in the lrecl configuration parameter. or use the lrecl command. Another remedy is to use the imwrap ON command. Refer to the <i>InfoMover File Transfer User's Guide</i> for more details.
IM045S: Serious Error: Failure locking InfoMover File Transfer master device tables.	A lock of the master device tables failed or yielded unexpected result. The file transfer is incomplete.	Verify the functionality of your IFT master device and your host connections. Resubmit any incomplete file transfers.

Table A-2 InfoMover File Transfer Event Messages (continued)

Message	Explanation	Suggested Remedy
IM046F: Fatal Error InfoMover File Transfer failed to properly select a Solutions Enabler gatekeeper device.	A gatekeeper device could not be selected for the specified reason.	Ensure that there is an available GateKeeper. If a GateKeeper is present, ensure that you have run imManager option (7) to Update Symmetrix Database. Refer to Chapter 2, <i>Installation Requirements</i> , in the <i>InfoMover File Transfer Installation Guide</i> for information about the GateKeeper. Refer to the <i>Installation Checklist</i> for your platform. In addition, ensure that the device that you selected as a GateKeeper is not listed in the GKAVOID file. Refer to the EMC Solutions Enabler documentation or more information about the GKAVOID file.
IM047F: Fatal Error: The client is not running the same build type as the local server.	The client was started with a different build type than that of the local server.	Use the same build type to run the client as was used to start the server.
IM048F: Fatal Error: An unidentified error was found in the process segment flag.	The IFT server received an unrecognized error code.	Verify that your transfer devices and host-Symmetrix connection are functioning properly. One way to verify the transfer devices is to run the Query option in the IFT Management Utility. Use the List Connections option in the IFT Management Utility to view current connections. Clear any unresponsive connections. You may need to use the Format option in the IFT Management Utility to format the devices.
IM049S: Serious Error: Failed to create socket connection to local server.	IFT was unable to create the specified socket. The socket may already be in use, or the server may be listening on a different socket.	Ensure that you are requesting a connection to the correct socket (as designated by the port number in the IFT configuration file). Use a different socket or wait for the socket to be freed.
IM050E: Error: Failed to obtain catalog information for data set.	The system was unable to obtain the requested information about the specified dataset.	The dataset may have too many members.

Table A-2 InfoMover File Transfer Event Messages (continued)

Message	Explanation	Suggested Remedy
IM051F: Fatal Error: EMC Solutions Enabler failed to initialize. Aborting program.	The system was unable to initialize the EMC Solutions Enabler used by IFT. The version of EMC Solutions Enabler installed on your system is incompatible with this release of IFT.	Please upgrade EMC Solutions Enabler before proceeding.
IM052E: Serious Error: Unsupported MVS DSORG (dataset organization).	An attempt was made to access a dataset whose dataset type is not supported by IFT.	Refer to the <i>InfoMover File Transfer User's Guide</i> for a description of supported dataset types.
IM053E: Error: GDG base does not exist.	IFT cannot allocate a GDG base name that does not exist.	Either use IDCAMS to create a GDG base name, or use an existing GDG base name.
IM054S: Serious Error: Dynamic Allocation error.	Dynamic allocation error (SVC 99) for mainframe dataset.	Check log for error messages and return codes. Check OS/390 and z/OS documentation for information about Dynamic Allocation (SVC 99) return codes.
IM055E: Error: LIKEDSN dataset error.	The dataset name supplied for the LIKEDSN parameter is not valid or does not exist.	Reissue the likedsn command specifying a valid cataloged dataset name. Do not specify a VSAM dataset, a GDG relative value, or a dataset on tape.
IM056E: Error: No space left on file system.	There is not enough space left in the file system to create the file.	Remove other files in the file system to make room for this file or change the location where the new file is being created.
IM057E: Error: Broken pipe.	The named pipe was broken. The file system may be out of space, or the pipe may have been killed manually.	Verify that the file system has enough space for the file you are transferring. Verify that the pipe was not closed manually.
IM058E: Error: Imfilenumber is greater than 1 and volume not specified for new dsn.	OS/390 and z/OS dynalloc fails allocation when file sequence number is greater than one and volume is unspecified.	Specify a volume to create a tape dataset or reset Imfilenumber to allow allocation to DASD or tape (file number 1).
IM059E: Error: Sequential dataset with member specified.	Invalid sequential dataset specified.	Specify dataset name without member.

Table A-2 InfoMover File Transfer Event Messages (continued)

Message	Explanation	Suggested Remedy
IM060E: Error: The absolute path of the file, including the initiator's current working directory, cannot exceed 150 characters.	The absolute path of file, including the initiator's current working directory, is more than 150 characters.	This path is specified by the imSwDir configuration parameter in the IFT configuration file. Edit the path to comply with this restriction. Note: The absolute pathname is restricted to 150 characters for IFT releases 3.0 and higher.
IM061E: Error: The MVS server name contains invalid characters.	The mainframe server name cannot contain lowercase letters or the symbols ';' or '/'.	Verify that the mainframe server name specified in your startup procedure does not contain lowercase letters or the symbols ';' or '/'.
IM062E: Error: PDS error.	The PDS supplied cannot be used. The problem can be: <ul style="list-style-type: none"> • A member name is missing • There is no more space • An I/O error occurred 	Specify the PDS member name in parenthesis after the PDS name. The PDS may need to be compressed or reallocated with more directory blocks, or another user may already be allocated to the PDS. Also, you might not have authority to access the dataset.
IM063I: Informational: <Data_Set_Name> WRITTEN.	The specified dataset was written successfully to the mainframe system on which the Write To Operator (WTO) command was issued. This message can only be generated when a data access transfer has been executed.	None required.
IM064E: Error: InfoMover File Transfer Internal Error.	IFT encountered an internal error.	Check log files for an associated error.
IM065I: Informational: Client logging.	Indicates a client has opened or closed a connection to a server.	None required.
IM067W: Warning: Unable to issue WTO RC=nn.	IFT is unable to issue a Write To Operator (WTO) command after the successful transfer of a mainframe dataset. The value of <i>nn</i> is the return code.	If the return code (RC) is 5, an internal error has occurred. If the return code is a value other than 5, you should refer to the <i>MVS Assembler Services Reference Manual</i> discussion of WTO.

Table A-2 InfoMover File Transfer Event Messages (continued)

Message	Explanation	Suggested Remedy
IM068E: Error: InfoMover supports PAM on SunOS 5.6 and greater.	PAM (Pluggable Authentication Modules) are supported in SunOS releases 5.6 (Solaris 2.6) and greater.	Check your OS type. If the type is SunOS, check the version and upgrade if necessary.
IM69S: Serious Error: Getmain failure for ECSA storage needed for FBA IO.	There is an internal mainframe error.	Contact your mainframe system administrator regarding an ECSA storage shortage.
IM070W: Warning: Failure setting actualHomeDir. Using imSwDir as current working directory instead.	IFT was unable to determine the value of the <i>HOME</i> or <i>home</i> environment variable.	Verify that the <i>HOME</i> or <i>home</i> environment variable is set for your user login.
IM071I: Informational: Delete request successfully completed.	The file was deleted successfully.	None required.
IM72E: Error: Delete request failed.	This error occurs in the following circumstances: <ul style="list-style-type: none"> • The requested file does not exist in the specified location. • The current IFT user does not have permission to delete the requested file. • The specified file is already in use and locked by another IFT server (or another software process). • The specified file was a directory. 	Check the use of commands and the spelling of the filename. <ul style="list-style-type: none"> • Use a different username or work with your system administrator to set up the proper permissions for the file. • Resubmit the delete request after the target location resource has been freed or unlocked. • Make sure you have the correct filename. • Make sure the file you are attempting to delete is not a directory.
IM073W: Warning: Unable to resolve IP address.	The system was unable to determine the IP address of a specified host due to network related problems.	If the client is running on open systems, ensure that the host is connected to the network and the network environment on the host is setup properly. If the client is running on OS/390 and z/OS software, ensure that the host has access to a name server, TCP networking is on, and a proper TCP networking environment is setup on the host.

Table A-2 InfoMover File Transfer Event Messages (continued)

Message	Explanation	Suggested Remedy
IM074E: Error: Unable to locate empty connection slot.	All of the connection slots allocated for the transfers are currently full.	Use the List Connections option in the IFT Management Utility to view the connection slots. You may want to clear unresponsive connections. If all of the slots are currently active, you may want to increase the value of the maxcon configuration parameter.
IM075E: Error: Wildcarding error.	There was an error in a wildcard specification used with the mput and mget CLI commands.	The error can be caused by one of the following conditions: <ul style="list-style-type: none"> • No matches were found • The files found exceeded the maximum allowed for wildcard matches • There is insufficient space for the match list • There was an internal error, possible invalid parameters To solve the problem, check for: <ul style="list-style-type: none"> • No returned matches • The current value of the max_pattern_matches configuration parameter (and raise it if necessary) • Current space in your working directory • The parameters you used with the mput or mget command
IM076W: Warning: Connection still active.	A request has been made to terminate the IFT service. However, the connection referenced in the warning message is still active.	The IFT server terminates when all active connections are completed. If the connection involves a mainframe, you can stop the server immediately by issuing the MODIFY,FORCE mainframe system command.
IM077W: Warning: Connection appears unresponsive.	A request has been made to terminate the IFT server. However, the connection referenced in the warning message appears to be unresponsive. The shutdown of the server proceeds without waiting for the connection to be cleared.	None required. However, if you want, use the IFT Management Utility to clear the connection.
IM078W: Warning: Active connection(s) found - shutdown will wait until connections end.	A request has been made to terminate the IFT server. However, at least one active connection to the server was found. These connections are identified in messages issued to the IFT server log. Shutdown of the server waits until these connections terminate.	Wait for the active connections to terminate. If the connection involves a mainframe server, you can stop the server immediately by issuing the STOP,FORCE mainframe system command.

Table A-2 InfoMover File Transfer Event Messages (continued)

Message	Explanation	Suggested Remedy
IM079I: Informational: IFT Server Shutdown Complete.	The server has terminated.	None required.
IM080S: Serious Error: Error trying to perform normal server shutdown.	An error occurred when processing a request for normal server shutdown. Normal server shutdown cannot proceed.	To terminate the server it is necessary to issue the MODIFY,FORCE or CANCEL mainframe command to this server. To determine the source of the error, check the IFT server log.
IM081I: Informational: Connection has terminated.	The connection identified in the message has terminated. This message is issued if termination of the server has been suspended while active connections are being processed. If other connections are still active, then shutdown processing is still suspended.	None required.
IM082I: Informational: IFT Server Shutdown Command Received.	(Mainframe hosts only.) A STOP or MODIFY,STOP mainframe system command has been issued. No new connections are accepted for this server. The server terminates when all active connections end.	None required.
IM083I: Informational: IFT Server Forced Shutdown Command Received.	(Mainframe hosts only.) A MODIFY,FORCE mainframe system command has been issued. The server terminates immediately.	None required.
IM084W: Warning: IFT Server Shutdown already in progress.	(Mainframe hosts only.) A STOP or MODIFY,STOP mainframe system command has been issued. However, server shutdown is already in progress due to an earlier STOP or MODIFY,STOP command.	None required.
IM085E: Error: MODIFY command invalid.	(Mainframe hosts only.) An invalid MODIFY mainframe system command was issued to the server identified in the message. The error message contains a brief description of the error. The only valid parameters for the MODIFY command are STOP and FORCE .	Enter a valid MODIFY mainframe system command.

Table A-2 InfoMover File Transfer Event Messages (continued)

Message	Explanation	Suggested Remedy
IM086S: Serious Error: Unable to perform processing necessary to accept MVS system commands.	(Mainframe hosts only.) The error described in the message occurred during server initialization. The server terminates.	Check the IFT server's job log and the server log for additional information that would explain the source of the error.
IM087W: Warning: Unable to issue WTO message.	(Mainframe hosts only.) The error described in the message occurred when trying to issue a WTO (Write To Operator) command.	If the error description contains the words Internal Error, then an internal error has occurred, otherwise, the WTO return code is shown. Refer to the <i>MVS Assembler Services Reference Manual</i> for a discussion of WTO.
IM088E: Error: System Error.	An error occurred in the specified server while processing the dataset.	Check the mainframe job log and the IFT log of the specified server for information about the server.
IM089E: Error: MVS subtask still active for a terminated connection.	(Mainframe hosts only.) A TCB is still active on behalf of the PID noted in the error message. This situation may occur because of a communication error between the server and the connected client or server. A normal shutdown cannot occur while mainframe subtasks are still active.	You can force immediate termination of the server by issuing the MODIFY command with the FORCE option, or the CANCEL command. The syntax of the MODIFY FORCE command is F server-jobname,FORCE .
IM090E: Error: Server must be terminated via MVS CANCEL or MODIFY FORCE.	Mainframe only. Normal termination of the server cannot occur. Please refer to the IFT log messages in SYSPRINT to determine the reason the server cannot be terminated normally.	Use the CANCEL command to cancel the server, or issue the MODIFY FORCE command. The syntax of the MODIFY SOURCE command is F server-jobname,FORCE .
IM091E: Error: Unable to change locale character classification definition.	The setlocale (LC_TYPE, code-page-name) function failed.	Ensure that a valid code page name is specified in the locale= parameter in the configuration file.
IM097F: Fatal Error: EMCSCF address space is not running.	(Mainframe hosts only.) InfoMover requires ResourcePak Base. ResourcePak Base has an address space, typically named EMCSCF, running. This address space is not running.	Either: <ul style="list-style-type: none"> ResourcePak is not running or it is not installed. Install ResourcePak and/or start the EMCSCF address space. ResourcePak Base was customized and running under a different name.

Table A-2 InfoMover File Transfer Event Messages (continued)

Message	Explanation	Suggested Remedy
IM098E: Error: dir command has (*) wildcarding, but dsn is not a PDS or a GDG basenane.	(Mainframe hosts only.) User specified a (*) wildcard and dir did not find expected dataset type: a PDS or GDG basenane.	Use the (*) wildcard form only for PDSs or GDGs.
IM101S: Serious Error: A server with the same name has already been started, use -f option when starting the server if you want to start the server anyway.	You are trying to start a server with the same name as a currently running server.	Use a different name for the second server.
IM102S: Serious Error: The target server is inactive.	The server you are trying to open is not active.	Start the server before you attempt IFT operations.
IM103S: Serious Error: The server failed to have a heartbeat.	No heartbeat signal has been detected from the server in question.	Check to make sure you are using the current Solutions Enabler and microcode version.
IM104F: Fatal Error: The master device is in use by another server or client.	The task cannot proceed because the master device is not available.	Either: <ul style="list-style-type: none"> • Servers or client need to be stopped so that the IFT Management Utility can format the master devices. • The servers or client need to wait until the IFT Management Utility has completed formatting the transfer disks.
IM105F: Fatal Error: An initiator/connector is already running for the same server name/master device combination.	An operations error. A duplicate IFT started task was attempted.	Do not attempt to start a duplicate server task.
IM106W: Warning: The client detects that an initiator is not available.	A possible client timeout condition has occurred if the client and initiator are running in the same LPAR.	If the client and the initiator are running in the same LPAR, ensure that the initiator task is running before you run the client.
IM107E: Error: The server is not configured for character translation.	The MVS server has not been configured to perform data translation.	Configure the MVS server to perform file translation.

Table A-2 InfoMover File Transfer Event Messages (continued)

Message	Explanation	Suggested Remedy
IM108F: Fatal Error: MVS dataset or ddname specified in imXlateDir configuration parameter cannot be accessed or opened.	The mainframe server cannot allocate or open the translation dataset or ddname.	Verify the imXlateDir= parameter in the mainframe server configuration file. Check for proper security authorization. DSORG and DCB attributes.
IM109F: Fatal Error: Error parsing MVS character translation dataset or ddname.	The imXlateDir configuration parameter in the configuration file has a syntax error.	There is an insufficient number of entries in the translation table (<256), the hex value specified for an entry is invalid, or the table: parameter is not found.
IM110F: Fatal Error: IFT load lib is NOT APF authorized.	The IFT loadlib must be APF authorized.	Use the mainframe operator command set prog, apf, add to APF authorize loadlib.

Event Messages in Cancelled Mainframe Jobs

Under certain circumstances, you can lose event messages from IFT on a mainframe host. IFT uses the Language Environment (LE/370) Multi-Tasking Facility (MTF) to schedule work into IFT subtasks.

To handle standard output streams within these subtasks, MTF creates separate output files for each parallel function. These output files contain event messages issued from the subtask.

CANCEL Command and IFT Servers

When you issue a mainframe **CANCEL** command to stop an IFT server that is running as a started task, OS/390 and z/OS mainframe software considers that an abend has occurred. JES handles the output files from the cancelled IFT server as it would output from any other abnormally terminated job.

Abnormal Disposition OUTDISP Subparameter

The disposition of output for datasets created by started tasks is controlled by the OUTDISP parameter in the JES2 STCCCLASS initialization statement. If you set the abnormal disposition subparameter of OUTDISP to have these output files deleted from the JES output after you stop the server task with a mainframe **CANCEL** command, you lose any event messages in the output files.

To avoid losing potentially useful IFT messages, make sure that the abnormal disposition subparameter of OUTDISP does not delete this output file from the job's held output.

Using IFT on Mainframe Hosts

This appendix describes information you need to know to use IFT for data operations involving OS/390 and z/OS mainframe hosts as sources or targets. Topics include:

- ◆ Creating Batch CLI Client Jobs..... B-2
- ◆ Using Generation Data Groups..... B-7
- ◆ Specifying Allocation Attributes B-13
- ◆ Magnetic Tape Support B-24
- ◆ Managing Files with ISO Control Characters B-27
- ◆ Performing Translation on the Mainframe..... B-29

Creating Batch CLI Client Jobs

This section contains information about CLI client batch jobs.

Overview

As described in *CLI Batch or File Input Mode* on page 3-19, you can run a CLI client as a mainframe BATCH job if you supply a JCL procedure. The following sections describe some other points you need to keep in mind when you create CLI client batch jobs in OS/390 or z/OS systems.

Serializing Datasets

Before executing the steps of a job, the mainframe job entry subsystem (JES) obtains the proper serialization for all datasets identified in data definition statements within the job. This serialization is held until the last job step referencing the dataset has been completed.

If a disposition (DISP=) of NEW, OLD, or MOD is coded on any DD statement for a dataset, the system obtains exclusive access to that dataset. A statement specifying exclusive control overrides any statement that requests shared control.

JES also obtains exclusive access to a dataset if the normal or abnormal disposition of that dataset is DELETE. JES obtains only shared access to a dataset if the disposition of that dataset is SHR on every statement that references the dataset.

get and **put** Command Arguments

Serialization has implications for IFT when you invoke it from a multistep batch job. To ensure data integrity, IFT also serializes access to the local datasets identified in **get** and **put** CLI commands.

- ◆ The **put** command requires a *shared* access.
- ◆ The **get** command requires an *exclusive* access.

If you specify the local dataset in a **get** command and there is any subsequent job step that contains a data definition (DD) statement for that dataset, the **get** command fails because it requires exclusive access to the dataset.

If you specify the local dataset in a **put** command and there is a subsequent job step that contains a data definition statement for that dataset, the **put** command fails unless all dispositions are SHR.

SHR Disposition Status and put Command

The one exception occurs when a dataset in a **put** command argument is also specified in other DD statements with a disposition status of SHR. Because all steps only require shared access, the job succeeds.

Disposition Status Success and Failures

Table B-1 shows the success and failures with various disposition statuses:

Table B-1 Disposition Statuses for put and get Commands

FILE1 Disposition	JES Enqueue	put FILE1 remote_target (shared)	get remote_target FILE1 (exclusive)
		Success?	Success?
DISP=NEW	Exclusive	No	No
DISP=MOD	Exclusive	No	No
DISP=OLD	Exclusive	No	No
DISP=SHR	Shared	Yes ^a	No
DISP=(,DELETE)	Exclusive	No	No
DISP=(,DELETE)	Exclusive	No	No

a. True only if all references to the file specify DISP=SHR.

Example Consider the following example:

Suppose a job issues a **get** command to transfer a dataset from another host. Perhaps that the same job issues a DISP=(MOD) for that identical dataset at a later point in the job.

This other step causes JES to hold the exclusive enqueue on the dataset until the completion of the last job step that referenced this dataset. The IFT server is unable to allocate the dataset because it is held exclusively by the CLI client batch job.

One solution is to move allocations to a step prior to the CLI commands, or to another job altogether.

Another solution is to change any job steps that require exclusive control of the dataset so that the dataset is not named in a DD statement to JES. (JES serializes only on DD statements.) For example, you can use the IDCAMS **DELETE** command to delete a dataset or run TSO in batch to allocate a dataset.

Example B-1 is a sample JCL procedure:

Example B-1 Sample JCL Procedure

```

/* dynamic allocation via background TSO
//TSO EXEC PGM=IKJEFT01,DYNAMNBR=50
//SYSTSIN DD *
  alloc dsn('USERA.NEW1') cyl space(1) unit(sysda) new .....
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSIN DD DUMMY
/* step to populate USERA.NEW1
//UPDATE EXEC PGM=UPDATPGM
...
/* IFT transfer
//IM EXEC PGM=INFOMOVR
//SYSIN DD *
usera
passa
...
...
put 'USERA.NEW1' /unix/target_file
quit
/*
//SYSOUT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
...
/* IDCAMS delete
//DELETE EXEC PGM=IDCAMS
//SYSIN DD *
DELETE USERA.NEW1 NONVSAM
/*
//SYSPRINT DD SYSOUT=*

```

Continuation in SYSIN Control Statements

When you run the CLI client from a mainframe batch environment, you specify IFT control statements through the SYSIN DD statements in the job's JCL. In this environment, IFT ignores data in control statements past column 73 automatically if the first SYSIN statement contains an eight-digit sequence number.

Because of the maximum, permitted, dataset or filename length on OS/390, z/OS, UNIX, and Windows systems, any of the CLI commands that accepts names as variables can overflow the 72-character input area.

For that reason, you can continue CLI commands that you specify with OS/390 and z/OS SYSIN DD statements across multiple lines. This is especially useful with CLI commands that accept names as arguments, such as:

- ◆ **get**
- ◆ **put**
- ◆ **dir**
- ◆ **cd**
- ◆ **imrecvx**
- ◆ **imsendx**

A minus sign (dash) character at the end of an input statement indicates that the statement continues on the following line.

When you append the minus sign to the end of one or more blanks, input on the next line is considered to be the start of a new argument or option.

Example B-2 shows an sample:

Example B-2 Continuation Between Arguments

```
//SYSIN          DD *
-----1-----2-----3-----4-----5-----6-----7
get /fully/qualified/unix/file/name -                00010000
'FULLY.QUALIFIED.OS390.DATASET.NAME.OF.44.BYTES'      00020000
```

When the minus sign immediately trails a nonblank character, input from the next line is assumed to be the continuation of the current argument or option.

Example B-3 shows an example:

Example B-3 Continuation Across Arguments

```
//SYSIN          DD *
-----1-----2-----3-----4-----5-----6-----7
get /fully/qualified/unix/file/name'FULLY.QUALIFIED.OS390.DATASET.NAME.- 00010000
OF.44.BYTES'                                           00020000
```

In both cases, the dash is removed and the next line is appended directly to the preceding character to form the completed argument. (Single control statements can span several lines.) This is similar to the minus (dash) continuation in TSO.

Using Columns 73 through 80

Under certain circumstances, IFT accepts input in columns 73 through 80 of control statements. If the first SYSIN statement does not contain an eight-digit sequence number, subsequent input statements can use columns 73 through 80 for input.

Example B-4 shows an example:

Example B-4 Use of Columns 72 Through 80

```
//SYSIN DD *
*---+---1---+---2---+---3---+---4---+---5---+---6---+---7---+---8
put 'FULLY.QUAL.OS390.DATASET.NAME.OF.44.BYTES'           00010000
/fully/qual/unix/filename/which.could.be.extremely/Large - 00020000
infact.up.to.1024.Bytes                                   00030000
```

Using Generation Data Groups

A *generation data group* (GDG) is a group of related, cataloged datasets. A GDG is a collection of physical sequential (PS) files that are logically related by their content, and usually represent a history of the data over some period of time. The group is created through use of administrative utilities, and individual datasets in the group, called *generation data sets* (GDS), are created by running user programs.

This section describes how to use GDGs as sources or targets in data operations.

Overview

IFT allows you to use generation data groups as both:

- ◆ Sources of shared access data transfers
- ◆ Sources and targets of nonshared access data transfers

IFT permits you to transfer only one generation at a time, not an entire GDG.

Important: Datasets within a GDG must be physical sequential datasets; they cannot be partitioned datasets.

GDG Names

When using generation data groups with IFT, you must keep in mind three name components:

- ◆ The base name
- ◆ The absolute name
- ◆ The relative name

The Base Name

The GDG base name allows applications to refer to all datasets in the group by a common name. It also allows mainframe hosts to keep generations in chronological order, and automatically delete obsolete generations.

More Information: Refer to the *DFSMS/MVS: Using Data Sets* ¹ publication for more information about GDG datasets.

1. P/N SC26-4922, International Business Machines Corporation

You can refer to individual generations with a GDG by either a:

- ◆ Absolute name
- ◆ Relative name

The Absolute Name

Absolute names are constructed using the GDG base name and the *GxxxxVyy* suffix.

where:

xxxx

A decimal *generation number* (1 through 0999)

yy

A decimal dataset *version number* (0 through 99)

OS/390 and z/OS software maintains the generation number, and limits the number of generations that exist, according to the definition specified in the base name.

The Relative Name

With relative names, you replace the fully-qualified name by a reference to the relative generation number attached to the GDG base name. You enclose a zero, a positive integer, or negative integer in parentheses following the GDG base name. For example:

```
AGTEST.GDG.DATA.SET(+1)
```

instead of:

```
AGTEST.GDG.DATA.SET.G0004V00
```

The integer value in parentheses refers to the last absolute generation number assigned:

- ◆ Zero (0) — Indicates the current generation
- ◆ Plus one (+1) — Indicates the next absolute generation to be created
- ◆ Minus one (-1) — Indicates the previous generation

The relative generation number reflects the most recently cataloged information.

For example, it selects GDG datasets created or deleted by other jobs in a multitasking environment. Table B-2 on page B-9 shows this information.

Absolute and Relative Name Relationship

In Table B-2, the GDG base name is AGTEST.GDG.DATA.SET. The absolute name of the current generation is AGTEST.GDG.DATA.SET.G0003V00:

Table B-2 Absolute and Relative Dataset Names

Absolute Dataset Name	Relative Dataset Name
AGTEST.GDG.DATA.SET.G0001V00	AGTEST.GDG.DATA.SET(-2)
AGTEST.GDG.DATA.SET.G0002V00	AGTEST.GDG.DATA.SET(-1)
AGTEST.GDG.DATA.SET.G0003V00	AGTEST.GDG.DATA.SET(0)
AGTEST.GDG.DATA.SET.G0004V00	AGTEST.GDG.DATA.SET(+1)

GDG Support

The following sections describe points you need to keep in mind when you use GDGs as sources and targets in normal data operations.

Prerequisite for Using GDGs

Before you allocate individual generation data sets as IFT targets, you must define the GDG *base name*, using IBM's IDCAMS utility.

Using GDG Names in Transfers

You can reference a GDS as the source or target of a **get** or **put** command request with a relative or absolute dataset name. IFT accepts the relative names and generations of existing GDGs. IFT also accepts the absolute names of existing GDGs.

When using GDGs, you should maintain some degree of control over the execution of multiple jobs that refer to the same generation data group.

IFT processes GDG relative datasets in a different manner from the typical JCL batch job. In a batch job, the first time that you use a relative generation number for a GDG within a job, the system establishes the relationship between the relative generation number and the absolute generation number. This system maintains this relationship throughout the batch job.

For example, if you create a GDS with a relative generation number of (+1), the system recognizes any subsequent reference to (+1) throughout the job as having the same absolute generation number.

In IFT, however, the relative generation number is based on the most recent catalog information, and reflects GDG datasets created or deleted by other jobs in a multitasking environment. Therefore, if you create a GDS with a relative generation of (+1), any subsequent

references to this particular dataset should specify a relative number of (0), or the absolute GDG name. The absolute GDG name may be preferable if there is a chance that another job may have created another generation of the same GDG.

For example, you can use the following format to specify a relative GDG name as the target in a **get** and **put** command:

get/put <source> '<base>(n)'

where:

get/put

An IFT **get** or **put** command

<source>

The source dataset

<base>

The GDG base name

n

The relative generation indicator, a zero or an integer preceded by a plus (+) or minus (-)

Format You can use the following format to use an absolute GDG name as the source in a **get** and **put** command:

get/put '<base>.GxxxxVyy' <target>

where:

get/put

An IFT **get** or **put** command

<base>

The GDG base name

GxxxxVyy

The suffix

where:

xxxx

The generation number

yy

The version number

<target>

The target dataset

Although IFT cannot define a new GDG, it can create new versions of an existing GDG when you specify a positive generation indicator as part of the relative dataset name.

Important: You fully qualify the GDG dataset by enclosing it and the relative generation indicator in single quotes.

Examples

The first example moves the contents of the AGTEST.GDG.DATA.SET active generation data set to the specified transfile target UNIX file:

```
put 'AGTEST.GDG.DATA.SET(0)' /usr/testbed/transfile
```

The second example creates a new generation of the AGTEST.GDG.DATA.SET GDG on the mainframe system before moving the contents of the source UNIX file:

```
get /user/testbed/labtest1 'AGTEST.GDG.DATA.SET (+1)
```

The dir and ldir CLI Commands and GDGs

You can use the **dir** and **ldir** CLI commands to obtain directories of GDGs, and thereby track GDG usage.

More Information: Refer to *dir* on page 4-14 and *ldir* on page 4-68 for more information about these commands.

Using Multiple Digit Relative Generation Numbers

IFT allows the use of GDG generation numbers of more than one digit. For example, both of the following are correct:

```
'AGTEST.GDG.DATA.SET(+10)'  
'AGTEST.GDG.DATA.SET(-50)'
```

Base Names in the Same Mainframe Sysplex or GRS Complex

IFT does not support the use of relative generation numbers for mainframe-to-mainframe GDG transfers that have identical GDG base names for the source and target datasets if the local and remote mainframe systems are in the same Sysplex or Global Resource Serialization (GRS) complex.

For example, the following **put** CLI command fails:

```
put PATREC.GDG.DATA.SET(0) PATREC.GDG.DATA.SET(+1)
```

Multiple Jobs Using the Same GDG

When using GDGs in IFT data operations, you should maintain some degree of control over the execution of multiple jobs that refer to the same GDG. IFT requires that you:

- ◆ Refer to a newly-created GDG as (+1). Subsequent steps within the same job must refer to the generation as (0).
- ◆ Refer to existing generations by the generation number (GxxxVyy) in subsequent jobs.

Even with these guidelines, be careful in environments where jobs may refer to the same generation group concurrently. One job may catalog a new version of a generation, or delete the generation wanted by a concurrent job.

More Information: Refer to IBM's *DFSMSF/MVS Using Data Sets¹ Manual* for recommendations about retrieving generation data groups, and descriptions of the pitfalls to avoid.

1. P/N SC26-4922, International Business Machines Corporation.

Specifying Allocation Attributes

The following sections discuss how to specify allocation attributes for new target mainframe datasets.

Overview

IFT has allocation rules that specify how it determines allocation attributes. When you are creating a new target dataset and specify a member, you can set the attributes for the target data through:

- ◆ Defaults
- ◆ Configuration parameters
- ◆ The **likedsn** CLI command
- ◆ Other CLI commands

More Information: This guide does not discuss the IFT configuration parameters. Refer to the *InfoMover File Transfer System Administrator's Guide* for a discussion of the configuration file and configuration parameters.

Default Values

At the bottom level are the preset defaults supplied with the IFT software. In the absence of any other specification, either through setting an IFT configuration parameter or issuing a CLI command, IFT uses the following preset default allocation attributes:

- ◆ Space attributes
(default primary cylinders = 2, default secondary cylinders = 5)
- ◆ Record length (lrecl) = 4092
- ◆ Record format (recfm) = VB
- ◆ DSORG=PS
- ◆ UNIT=SYSDA

Configuration Parameters

At the next level are the configuration parameters that specify allocation attributes. Any allocation attribute you specify as a configuration parameter (either through the optional configuration or through editing the IFT configuration file on mainframe systems, IMCONFIG member, as part of the advanced configuration) overrides any IFT defaults.

Table B-3 on page B-14 shows the configuration parameters that specify the default allocation attributes for a new target dataset.

Table B-3 lists the type of configuration you need to perform to set these values. Table B-3 also shows the form of these configuration parameters as they appear in the IFT configuration file:

Table B-3 Configuration Parameters for New Dataset Allocation

Configuration Parameter Description	Set in	Configuration File Parameter Syntax
Default blksize	Advanced configuration	default_blksize=
Default primary cylinder	Optional configuration	default_cylinders=
Default DCB buffer size	Advanced configuration	default_bufno=
Default directory blocks	Optional configuration	default_imdirblocks=
Default position of dataset on tape	Advanced configuration	default_imfilenumber=
Default lrecl	Optional configuration	default_lrecl=
Default recfm	Optional configuration	default_recfm=
Default release	Optional configuration	default_release=
Default secondary cylinder	Optional configuration	default_sec_cylinders=
Default unit	Optional configuration	default_unit=

More Information: Refer to the *InfoMover File Transfer System Administrator's Guide* for a description of the IFT configuration file and configuration parameters.

The **likedsn** Command

At the third level for many allocation attributes is the **likedsn** CLI command. The characteristics of the model dataset you use as the argument of a **likedsn** command overrides some of the configuration parameters that specify default allocation attributes.

Argument Characteristics

The **likedsn** command argument can be a:

- ◆ Physical sequential (PS) dataset
- ◆ Partitioned data set (PDS)
- ◆ Relative GDG

Attributes When SMS is Active

When SMS is active, IFT uses the same allocation attributes of the model dataset that would be copied if you had specified the LIKE parameter in a JCL data definition statement. These attributes include the following:

- ◆ Record format (RECFM)
- ◆ Record length (LRECL)
- ◆ Space allocation (SPACE)

The system determines the space to be allocated by adding up the space allocated in the first three extents of the model dataset. Therefore, the space allocated for the new dataset generally does not match the space specified for the model.

Important: To ensure proper space allocations, use the **imcylinder** command to specify the primary and secondary space allocation explicitly. Refer to *imcylinder (OS/390 and z/OS)* on page 4-40.

You can find the complete rules for allocating attributes of a new dataset when SMS is active in Table B-4 on page B-17 (for physical sequential and partitioned data sets) and Table B-5 on page B-19 (for GDG datasets).

Attributes When SMS is not Active

When SMS is not active, IFT uses the same attributes in the model dataset that would be copied if you had specified the DCB=<model> parameter in a JCL data definition statement. These attributes include the following:

- ◆ Record format (RECFM)
- ◆ Record length (LRECL)
- ◆ Block size (BLKSIZE)

The system determines the space allocation by the IFT default, or by the value you enter as an **imcylinder** argument.

You can find more information about the DCB LIKE parameter or the DCB=<dsname> option in IBM's *MVS JCL Reference Manual* and *MVS JCL User's Guide*.

You can find the complete rules for allocating attributes of a new dataset when SMS is not active in Table B-5 on page B-19.

Other CLI Commands

At the highest level are other CLI commands including:

- ◆ **blksize**
- ◆ **imcylinder**
- ◆ **imdirblocks**
- ◆ **imfilenumber**
- ◆ **imrelease**
- ◆ **lrecl**
- ◆ **recfm**
- ◆ **unit**
- ◆ **volcount**
- ◆ **volume**

All of these CLI commands temporarily override all other possible sources of allocation attributes. In other words, these CLI commands override the following:

- ◆ System defaults
- ◆ Configuration parameters
- ◆ Model dataset argument of a **likedsn** CLI command

For example, the primary and secondary cylinder extents you specify as a argument to **imcylinder** overrides the `default_cylinders` and `default_secondary_cylinders` configuration parameters just as those configuration parameters override the system defaults.

The effect of these CLI commands is temporary, as they are provided to deal with certain exceptions to the normal case.

More Information: Refer to descriptions of these CLI commands in Chapter 3, *Using the Command Line Interface*, for more information about how to reenble the configuration parameter values as the default.

Physical Sequential and Partitioned Data Sets

Table B-4 lists the rules that IFT follows to determine the allocation parameters for new physical sequential and partitioned data sets in an environment in which:

- ◆ SMS is active
- ◆ SMS is not active

The entries are listed in order of precedence. That is, by default, condition 1 applies first. If condition 1 is not present, condition 2 applies, and so forth.

If there are no numbered conditions, the single condition within the table cell apply under all circumstances.

Table B-4 New PS and PDS Allocation Rules

Dataset Attribute	SMS Active	SMS Not Active
space (Space allocation)	<ol style="list-style-type: none"> 1. Primary and secondary space specified with the imcylinder command. 2. Primary and secondary space calculated using the likedsn command argument as model (sum of first three extents). 3. Space specified through the default_cylinders and default_secondary_cylinders configuration parameters. 4. Uses Primary = 2 Secondary = 5. <p>1, 3, and 4 are allocated in cylinders. 2 is allocated in tracks.</p>	<ol style="list-style-type: none"> 1. Primary and secondary space specified with the imcylinder command. 2. Space specified through the default_cylinders and default_secondary_cylinders configuration parameters. 3. Uses Primary = 2 Secondary = 5. <p>Always allocated in cylinders.</p>
unit (Unit specification)	<ol style="list-style-type: none"> 1. Specified by the unit command. 2. Specified by the default_unit configuration parameter. 3. Uses SYSDA (system default). 	<ol style="list-style-type: none"> 1. Specified by the unit command. 2. Specified by the default_unit configuration parameter. 3. Uses SYSDA (system default).
volume (Volume)	<ol style="list-style-type: none"> 1. Specified with the volume command. 2. System determined. 	<ol style="list-style-type: none"> 1. Specified with the volume command. 2. System determined.
lrecl (Record length)	<ol style="list-style-type: none"> 1. Specified by the lrecl command. 2. Specified by the model dataset used with the likedsn command. 3. Specified by the default_lrecl configuration parameter. 4. Uses IFT default of 4092. 	<ol style="list-style-type: none"> 1. Specified by the lrecl command. 2. Specified by the model dataset used with the likedsn command. 3. Specified by the default_lrecl configuration parameter. 4. Uses IFT default of 4092.

Table B-4 New PS and PDS Allocation Rules (continued)

Dataset Attribute	SMS Active	SMS Not Active
recfm (Record format)	<ol style="list-style-type: none"> 1. Specified by the recfm command. 2. Specified by the model dataset used with the likedsn command. 3. Specified by the default_recfm configuration parameter. 4. Uses IFT default of VB. 	<ol style="list-style-type: none"> 1. Specified by the recfm command. 2. Specified by the model dataset used with the likedsn command. 3. Specified by the default_recfm configuration parameter. 4. Uses IFT default of VB.
blksize (Block size)	<ol style="list-style-type: none"> 1. Specified by the blksize command. 2. Specified in default_blksize configuration parameter. 3. System determined. 	<ol style="list-style-type: none"> 1. Specified by the blksize command. 2. Specified in default_blksize configuration parameter. 3. System determined.
dsorg=pds/directory blocks	<ol style="list-style-type: none"> 1. Specified by the imdirblock command argument. 2. Specified by the model dataset used with the likedsn command. 3. Specified by the default_imdirblocks configuration parameter. 4. Uses default_imdirblocks=20. 	<ol style="list-style-type: none"> 1. Specified by the imdirblock command argument. 2. Specified by the default_imdirblocks configuration parameter. 3. IFT default of 20 directory blocks.
dsorg=pdse	<ol style="list-style-type: none"> 1. Specified by imdirblocks pdse command/argument. 2. Specified by likedsn pds command/argument. 3. Specified by the default_imdirblocks configuration parameter set to pdse (default_dirblocks=pdse). 	Not applicable.
bufno (for tuning performance - applies to all transfers)	<ol style="list-style-type: none"> 1. Specified by the imbufno command argument. 2. Specified by the default_bufno configuration parameter. 3. Uses default_bufno=2. 	<ol style="list-style-type: none"> 1. Specified by the imbufno command argument. 2. Specified by the default_bufno configuration parameter. 3. Uses default_bufno=2.
Release space for new datasets	<ol style="list-style-type: none"> 1. Specified by the imrelease command. 2. Specified by the default_release configuration parameter. 3. Do not release space. 	<ol style="list-style-type: none"> 1. Specified by the imrelease command argument. 2. Specified by the default_release configuration parameter. 3. Do not release space.
Tape file sequence number	<ol style="list-style-type: none"> 1. Specified by the imfilenumber command. 2. Specified by the default_imfilenumber configuration parameter. 3. Uses default_imfilenumber=1. 	<ol style="list-style-type: none"> 1. Specified by the imfilenumber command. 2. Specified by the default_imfilenumber configuration parameter. 3. Uses default_imfilenumber=1.

Table B-4 New PS and PDS Allocation Rules (continued)

Dataset Attribute	SMS Active	SMS Not Active
volcount (maximum number of output tape volumes)	<ol style="list-style-type: none"> Specified by the volcount command. System determined. Refer to the <i>MVS/ESA SP V5 JCL Reference^a</i> for information about the volume count subparameter of the JCL DD parameter VOLUME.	<ol style="list-style-type: none"> Specified by the volcount command. System determined. Refer to the <i>MVS/ESA SP V5 JCL Reference^a</i> for information about the volume count subparameter of the JCL DD parameter VOLUME.

a. P/N GC28-1479. International Business Machines Corporation

SMS Parameters: GDG Datasets

Table B-5 lists rules IFT follows to determine the allocation parameters for GDG datasets in an environment in which:

- ◆ The GDG dataset is managed by SMS
- ◆ The GDG dataset is not managed by SMS, but SMS is active
- ◆ SMS is not active

The entries are listed in order of precedence. That is, by default, condition 1 applies first. If condition 1 is not present, condition 2 applies, and so forth. If there are no numbered conditions, the single condition within the table cell apply under all circumstances.

Table B-5 New GDG Datasets Allocation Rules

Dataset Attribute	GDG SMS Managed	SMS Active But GDG Not SMS Managed	SMS Not Active
blksize	<ol style="list-style-type: none"> Block size specified through blksize command. Specified by value in <code>default_blksize</code> configuration parameter. Block size specified by host system. 	<ol style="list-style-type: none"> Block size specified through blksize command. Specified by value in <code>default_blksize</code> configuration parameter. Block size specified by host system. 	<ol style="list-style-type: none"> Block size specified through blksize command. Specified by value in <code>default_blksize</code> configuration parameter. Block size specified by host system.

Table B-5 New GDG Datasets Allocation Rules (continued)

Dataset Attribute	GDG SMS Managed	SMS Active But GDG Not SMS Managed	SMS Not Active
space (Space allocation)	<ol style="list-style-type: none"> 1. Primary and secondary space specified through the imcylinder command. 2. Calculated using the likedsn command dataset as a model (sum of first three extents). 3. Primary space specified through the default_ primary_cylinders configuration file value. The secondary space specified through the default_ secondary_ cylinders configuration file value. 4. Primary =2 Secondary=5. 	<ol style="list-style-type: none"> 1. Primary and secondary space specified through the imcylinder command. 2. Calculated using the likedsn command dataset as model (sum of first three extents). 3. Primary space specified through the default_ primary_cylinders configuration file value. The secondary space specified through the default_ secondary cylinders configuration file value. 4. Primary = 2 Secondary = 5. 	<ol style="list-style-type: none"> 1. Primary and secondary space specified through the imcylinder command. 2. Primary space specified through the default_ primary_cylinders configuration file value. The secondary space specified through the for default_ secondary_cylinders configuration file value. 3. Primary = 2 Secondary = 5.
unit (Unit specification)	<ol style="list-style-type: none"> 1. Specified by the unit command. 2. Specified by the default_unit configuration value. 3. Uses SYSDA (system default). 	<ol style="list-style-type: none"> 1. Specified by the unit command 2. Specified by the default_unit configuration value. 3. Uses SYSDA (system default). 	<ol style="list-style-type: none"> 1. Specified by the unit command. 2. Specified by the default_unit configuration value. 3. Uses SYSDA (system default).
volume (Volume)	<ol style="list-style-type: none"> 1. Specified with the volume command. 2. System determined. 	<ol style="list-style-type: none"> 1. Specified with the volume command. 2. System determined. 	<ol style="list-style-type: none"> 1. Specified with the volume command. 2. System determined.
volume count	<ol style="list-style-type: none"> 1. Specified by the volcount command. 2. System determined. Refer to the <i>MVS/ESA SP V5 JCL Reference</i>^a for information about the volume count subparameter of the JCL DD parameter VOLUME. 	<ol style="list-style-type: none"> 1. Specified by the volcount command. 2. System determined. Refer to the <i>MVS/ESA SP V5 JCL Reference</i>^a for information about the volume count subparameter of the JCL DD parameter VOLUME. 	<ol style="list-style-type: none"> 1. Specified by the volcount command. 2. System determined. Refer to the <i>MVS/ESA SP V5 JCL Reference</i>^a for information about the volume count subparameter of the JCL DD parameter VOLUME.

Table B-5 New GDG Datasets Allocation Rules (continued)

Dataset Attribute	GDG SMS Managed	SMS Active But GDG Not SMS Managed	SMS Not Active
space (Space allocation)	<ol style="list-style-type: none"> 1. Primary and secondary space specified through the imcylinder command. 2. Calculated using the likedsn command dataset as a model (sum of first three extents). 3. Primary space specified through the default_primary_cylinders configuration file value. The secondary space specified through the default_secondary_cylinders configuration file value. 4. Primary =2 Secondary=5. 	<ol style="list-style-type: none"> 1. Primary and secondary space specified through the imcylinder command. 2. Calculated using the likedsn command dataset as model (sum of first three extents). 3. Primary space specified through the default_primary_cylinders configuration file value. The secondary space specified through the default_secondary_cylinders configuration file value. 4. Primary = 2 Secondary = 5. 	<ol style="list-style-type: none"> 1. Primary and secondary space specified through the imcylinder command. 2. Primary space specified through the default_primary_cylinders configuration file value. The secondary space specified through the for default_secondary_cylinders configuration file value. 3. Primary = 2 Secondary = 5.
unit (Unit specification)	<ol style="list-style-type: none"> 1. Specified by the unit command. 2. Specified by the default_unit configuration value. 3. Uses SYSDA (system default). 	<ol style="list-style-type: none"> 1. Specified by the unit command 2. Specified by the default_unit configuration value. 3. Uses SYSDA (system default). 	<ol style="list-style-type: none"> 1. Specified by the unit command. 2. Specified by the default_unit configuration value. 3. Uses SYSDA (system default).
volume (Volume)	<ol style="list-style-type: none"> 1. Specified with the volume command. 2. System determined. 	<ol style="list-style-type: none"> 1. Specified with the volume command. 2. System determined. 	<ol style="list-style-type: none"> 1. Specified with the volume command. 2. System determined.
volume count	<ol style="list-style-type: none"> 1. Specified by the volcount command. 2. System determined. Refer to the <i>MVS/ESA SP V5 JCL Reference</i>^a for information about the volume count subparameter of the JCL DD parameter VOLUME. 	<ol style="list-style-type: none"> 1. Specified by the volcount command. 2. System determined. Refer to the <i>MVS/ESA SP V5 JCL Reference</i>^a for information about the volume count subparameter of the JCL DD parameter VOLUME. 	<ol style="list-style-type: none"> 1. Specified by the volcount command. 2. System determined. Refer to the <i>MVS/ESA SP V5 JCL Reference</i>^a for information about the volume count subparameter of the JCL DD parameter VOLUME.

Table B-5 New GDG Datasets Allocation Rules (continued)

Dataset Attribute	GDG SMS Managed	SMS Active But GDG Not SMS Managed	SMS Not Active
lrecl (Record length)	<ol style="list-style-type: none"> 1. Specified through the lrecl command. 2. Specified by the model dataset used with the likedsn command. 3. Specified through the default_lrecl configuration value. 	<ol style="list-style-type: none"> 1. Specified through the lrecl command. 2. Specified by the model dataset used with the likedsn command. 3. Specified through the default_lrecl configuration value. 4. Specified by the model DCSB. 	<ol style="list-style-type: none"> 1. Specified through the lrecl command. 2. Specified by the model dataset used with the likedsn command. 3. Specified through the default_lrecl configuration value. 4. Specified by the model DCSB.
recfm (Record format)	<ol style="list-style-type: none"> 1. Specified by the recfm command. 2. Specified by the model dataset used with the likedsn command. 3. Specified through the default_recfm configuration value. 	<ol style="list-style-type: none"> 1. Specified by the recfm command. 2. Specified by the model dataset used with the likedsn command. 3. Specified through the default_recfm configuration value. 4. Specified by the model DCSB. 	<ol style="list-style-type: none"> 1. Specified by the recfm command. 2. Specified by the model dataset used with the likedsn command. 3. Specified through the default_recfm configuration value. 4. Specified by the model DCSB.
blksize (Block size)	<ol style="list-style-type: none"> 1. Specified by blksize command. 2. Specified by default_blksize configuration parameter. 3. Determined by system. 	<ol style="list-style-type: none"> 1. Specified by blksize command. 2. Specified by default_blksize configuration parameter. 3. Determined by system. 	<ol style="list-style-type: none"> 1. Specified by blksize command. 2. Specified by default_blksize configuration parameter. 3. Determined by system.
bufno (for performance)	<ol style="list-style-type: none"> 1. Specified by the imbufno command. 2. Specified by default_bufno configuration parameter. 3. Uses bufno (JCL parameter)=2. 	<ol style="list-style-type: none"> 1. Specified by the imbufno command. 2. Specified by default_bufno configuration parameter. 3. Uses bufno (JCL parameter)=2. 	<ol style="list-style-type: none"> 1. Specified by the imbufno command parameter. 2. Specified by default_bufno configuration parameter. 3. Uses bufno (JCL parameter)=2.
Release space for new datasets	<ol style="list-style-type: none"> 1. Specified by the imrelease command. 2. Specified by the default_imrelease configuration parameter. 3. Do not release space. 	<ol style="list-style-type: none"> 1. Specified by the imrelease command. 2. Specified by the default_imrelease configuration parameter. 3. Do not release space. 	<ol style="list-style-type: none"> 1. Specified by the imrelease command. 2. Specified by the default_imrelease configuration parameter. 3. Do not release space.

Table B-5 New GDG Datasets Allocation Rules (continued)

Dataset Attribute	GDG SMS Managed	SMS Active But GDG Not SMS Managed	SMS Not Active
Tape file sequence number	<ol style="list-style-type: none"> 1. Specified by the imfilenumber command. 2. Specified by the default _imfilenumber configuration parameter. 3. Uses first position (file no. 1). 	<ol style="list-style-type: none"> 1. Specified by the imfilenumber command. 2. Specified by the default _imfilenumber configuration parameter. 3. Uses first position (file no. 1). 	<ol style="list-style-type: none"> 1. Specified by the imfilenumber command. 2. Specified by the default _imfilenumber configuration parameter. 3. Uses first position (file no. 1).

a. P/N GC28-1479, International Business Machines Corporation

Magnetic Tape Support

The following sections discuss data transfers involving OS/390 and z/OS datasets on magnetic tape.

Overview

Magnetic tape support is available in transfers involving an OS/390 and z/OS host. IFT allows you to use a dataset on magnetic tape as either the source or the target.

You can use all currently-supported mainframe host dataset organizations and record formats as source files on magnetic tape.

More Information: Refer to *File Transfers* on page 1-7 for a list of supported types.

Target Datasets on Tape

IFT allows you to specify a cataloged dataset on tape as both the source and the target of a transfer operation. To ensure proper DCB attributes for target mainframe tape datasets, do not use existing tape datasets as targets. Specify only new datasets as targets.

GDGs and Magnetic Tape

You can use GDGs on magnetic tape for sources and targets.

Configuration Parameters for Tape Operations

If you are using magnetic tape, you must set the following values in your IFT configuration file:

```
client_timeout=2000
command_timeout=2000
```

More Information: Refer to the appendix about advanced configuration in the *InfoMover File Transfer System Administrator's Guide* for a description of the units in which these values are expressed.

Remember, the time-out values you set in the `command_timeout` and `client_timeout` configuration parameters are influenced by delays for:

- ◆ Tape mounts
- ◆ Tape rewinds
- ◆ Dataset locates on the tape

Depending on your facility's service-level requirements for magnetic tape, you may need to increase the `command_timeout` and `client_timeout` configuration parameter values beyond the default values of 2000.

Tape Mount Operations

At the beginning of a data copy, the IFT server may request a remount of the tape for source mainframe host tape datasets if:

- ◆ The copy is mainframe-to-mainframe, variable-to-variable (V, VB, and VBS), and the source dataset contains variable length records.
- ◆ The copy is mainframe-to-open systems, and the source dataset has enabled trailing blank retention. (You have issued the **imtrail ON** CLI command.)

Each additional tape volume for a multivolume tape copy, requests an additional tape mount. These tape mount requests must be either:

- ◆ Serviced (the supplied tape)
- ◆ Canceled manually

If the IFT client associated with the copy terminates for any cause when you are at a mount request, IFT does not cancel the pending mount request. In addition, IFT does not address any request from the current client or any subsequent IFT client on that system until you either service or cancel the mount request.

The only way you can cancel the pending mount request on the first or only volume is to cancel the mount request from a mainframe host by replying to the outstanding operator prompt.

If the pending mount request is for any volume in a multivolume dataset after the first volume, you can cancel the pending mount request by shutting down the IFT server.

Dataset Allocations

All dataset allocations cause a mainframe host to serialize on the Task Input Output Table (TIOT), using the `SYSZTIOT` enqueue. During allocations that involve tape mounts, a mainframe host holds exclusive access to the `SYSZTIOT` resource until the mount is serviced or canceled.

Outstanding tape mounts can cause contention on the `SYSZTIOT` resource, and affect all future allocations in the server address space. While a mount is outstanding, the IFT server can be blocked from servicing requests from IFT clients.

Configuration Parameters and CLI Commands

Keep in mind the following configuration file values and CLI commands listed in Table B-6 that effect magnetic tape source and target datasets:

Table B-6 Magnetic Tape Commands and Configuration Parameters

CLI Command	Configuration File Parameters
blksize	default_blksize
indirblocks	default_indirblocks
imfilenumber	default_imfilenumber
irecl	default_irecl
recfm	default_recfm
unit	default_unit

More Information: Refer to the *InfoMover File Transfer System Administrator's Guide* for a description of the configuration file parameters and to Chapter 3, *Using the Command Line Interface* for descriptions of the CLI commands.

Managing Files with ISO Control Characters

If you transfer a mainframe file containing ISO ¹ control characters to an open systems host and you do not override the default values of the **imaddlf** and **imtrail** client commands, the ISO control character at the start of each record is converted to one or more C control sequence characters, rather than to the ASCII equivalent of the EBCDIC ISO value.

More Information: Refer to *imaddlf (OS/390 and z/OS)* on page 4-32 and *imtrail (OS/390 and z/OS)* on page 4-57.

Fixed Format Files

If you need to transfer a fixed format file, containing ISO control characters that you want to retain in the target file, the procedure is to issue the **imtrail ON** CLI command before the transfer. This action bypasses the translation of the ISO character to a C control sequence character. However, using an **imtrail ON** command causes IFT to keep the trailing blanks of the mainframe source in the target file.

Variable Format Files

Keep in mind that the procedure does not apply if you want to transfer a variable format file containing ISO characters. Any specification of the **imtrail ON** command has no effect on variable format files containing ISO control characters. The only way to transfer a variable format file containing ISO control characters and retain the control characters in the target file is to specify the **imaddlf OFF** client command.

Be aware, however, that this action is not recommended since the command produces a target file that does not contain any line feeds.

Open Systems Files

If you want to transfer an open systems file that contains C control sequence characters to a mainframe host, remember the following:

- ◆ If the target mainframe file has a record format of FBA or VBA, the C control sequence characters is translated to equivalent ISO control characters.

1. International Standards Organization

- ◆ If you transfer a mainframe file containing ISO control characters to an open systems host, and then you transfer the resulting open systems file to a mainframe file in FBA or VBA format, the control characters of the target file may not match the control characters of the original mainframe file. This result is due to the way in which control characters are translated between ISO and C.

More Information: Refer to IBM's *OS.390 C/C++ Programming Guide* for information on how this translation occurs.

Miscellaneous Issues

If the target mainframe file does not have a record format of FBA or VBA, the C control /f (form feed) appears in the target file as a x'0C' byte.

If you transfer to a mainframe host, an open systems source file containing the ASCII equivalent of ISO control characters at the start of each line, these characters is not interpreted as ISO control characters. They simply translate as text.

If you are transferring a file from Windows to an OS/390 or z/OS system and back to a Windows host, or from an OS/390 or z/OS system to Windows host and back to the mainframe, ensure that your settings of IMCRLF are consistent.

Performing Translation on the Mainframe

As discussed in Chapter 4, IFT normally performs translation in mainframe-open systems transfers on the open systems host involved. In this way, IFT helps to remove compute cycles from the mainframe host.

However, as an option, you can configure IFT to perform the mainframe-open systems translations on the mainframe.

More Information: Refer to Chapter 4 in the *InfoMover File Transfer System Administrator's Guide* for more information.

This appendix reviews the EMC process for detecting and resolving software problems, and provides essential questions that you should answer before contacting the EMC Customer Support Center. Topics include:

- ◆ Overview of Detecting and Resolving Problems C-2
- ◆ Troubleshooting the Problem C-3
- ◆ Before Calling the Customer Support Center C-4
- ◆ Documenting the Problem C-4
- ◆ Reporting a New Problem C-5
- ◆ Sending Problem Documentation C-6

Overview of Detecting and Resolving Problems

EMC software products are supported directly by the EMC Customer Support Center in the United States. EMC uses the process shown in Figure C-1 to resolve customer problems with its software products:

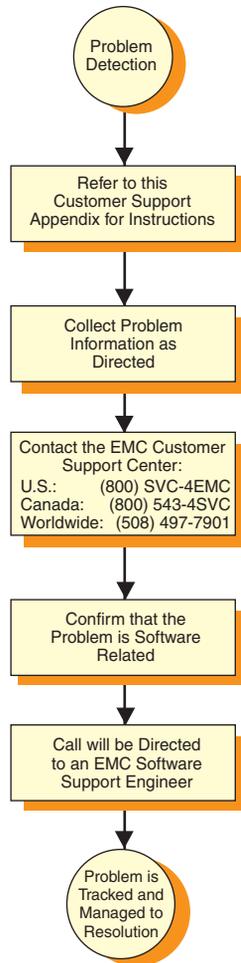


Figure C-1 Problem Detection and Resolution Process

Troubleshooting the Problem

Please perform the relevant diagnostic steps before you contact the EMC Customer Support Center:

1. Read the documentation carefully.
2. Reconstruct the events leading up to the problem and describe them in writing.
3. Run some test cases to reproduce the problem.

If you encounter a problem that requires technical programming or analysis, call the nearest EMC office or contact the EMC Customer Support Center at one of the following numbers:

United States: **(800) 782-4362 (SVC-4EMC)**

Canada: **(800) 543-4782 (543-4SVC)**

Worldwide: **(508) 497-7901**

Note: Please do not request a specific EMC Customer Support Representative unless one has already been assigned to your particular system problem.

For additional information about EMC products and services available to customers and partners, refer to the EMC Powerlink website at:

<http://powerlink.emc.com>

Before Calling the Customer Support Center

Have the following information available before calling the Customer Support Center or your support representative (if one has been assigned to you):

- Your company name
- Your name
- Your telephone number
- For an existing problem, the problem tracking system ID, if one was previously assigned to the problem by a support representative
- For an OS/390 or z/OS problem, the JESLOG, SYSPRINT, all STDOUT DD members of the server job output and similar output for the client, and the relevant portion of the SYSLOG

Documenting the Problem

If the EMC Customer Support Center requests information regarding the problem, please document it completely, making sure to include the following information:

- Your company name and address
- Your name
- Your telephone number
- The importance of the problem, so that it can be assigned a priority level

To expedite the processing of your support request, you can photocopy this list, and include it with the package.

Reporting a New Problem

For a new problem, please provide the following information:

- Release level of the software that you are running.
- Software installation parameters.
- Host type on which you are running.
- Operating system you are running and its release number.
- Functions of the software that you are running.
- Whether you can reproduce the problem.
- Previous occurrences of the problem.
- Whether the software has ever worked correctly.
- Time period that the software did work properly.
- Conditions under which the software worked properly.
- Changes to your system between the time the software worked properly and the problem began.
- Exact sequence of events that led to the system error.
- Message numbers and complete text of any messages that the system produced.
- A log file with the timestamp around the time of the error.
- Results from tests that you have run.
- Other related system output.
- Other information that may help solve the problem.

Sending Problem Documentation

Use one of the following methods to send documentation of the problem to the EMC Customer Support Center:

- ◆ e-mail
- ◆ FTP
- ◆ U.S. mail to the following address:

EMC Customer Support Center
171 South Street
Hopkinton, MA 01748-2208

If the problem was assigned a number or a specific support representative, please include that information in the address as well.

This glossary contains terms related to InfoMover File Transfer. Many of these terms are used in this guide.

A

alternate track

A track designated to contain data in place of a defective primary track. See also *primary track*.

actuator

A set of access arms and their attached read/write heads, which move as an independent component within a head and disk assembly (HDA).

adapter

Card that provides the physical interface between the director and disk devices (SCSI adapter), director and FICON director and serial channels (serial adapter).

API

Application programming interface. A set of routines that an application program uses to request and execute tasks performed by a computer's operating system.

B

bidirectional transfers

Also called duplex transfers. A bidirectional transfer is one in which data can be transferred simultaneously in both directions. If there is a single IFT client, that client can do both gets and puts. See also *dual initiated transfers* and *dual initiated transfers*.

C

cache	Random access electronic storage used to retain frequently used data for faster access by the channel.
cache slot	Unit of cache equivalent to one track.
catalog	An OS/390 or z/OS file containing dataset information.
CDAS	Acronym for Copper Directory Access Services. CDAS is the file browsing capability used by IFT. CDAS uses the FTP agent on the source host to retrieve a complete file listing on that host. A requirement for CDAS to operate is that a TCP/IP connection be present between the source and target hosts.
channel director	The component in the Symmetrix storage subsystem that interfaces between the host channels and data storage. It transfers data between the channel and cache.
command line arguments	<p>Command arguments are command line elements that provide information to the program, but do not change what the program does. For example: <code>telnet labnet9.1ss.emc.com</code>. The name <code>labnet9.1ss.emc.com</code> is an argument to the telnet program that specifies which computer the user wants to connect via telnet, but any name would be appropriate. The telnet program functions the same way regardless of what name is supplied to it.</p> <p>Arguments that have a fixed value are listed in the syntax description with that value. Arguments that can have a variable value are usually represented by a descriptive name in the syntax description.</p> <p>The name is usually enclosed in <code>< ></code> symbols, indicating that a meaningful argument value should be substituted for the name in the description.</p> <p>For example, <code>open <server></code> indicates that the open command takes a single argument whose value is the name of the server to which the user wants to connect.</p> <p>Conversely, the <code>imwrap ON OFF</code> command indicates that the ifsadmin command can accept either of two arguments, ON or OFF, but the value of those arguments cannot be changed.</p>
command line options	Command line options are switches that are designed to change the behavior of a command. On UNIX systems, command line options are usually provided as a single letter of the alphabet preceded by a

minus (-) sign. Multiple options can be strung together into a single option string.

For example, **ls -la** executes the **ls** command using the “**l**” and “**a**” options where “**l**” specifies that the user wants a long listing of data, and the “**a**” specifies that the user wants all files shown, including the “hidden” files.

In the syntax descriptions, options are usually represented by stringing them together, and proceeding them with a minus sign.

Some options require arguments to accompany them. In this case, the particular option and its associated arguments are listed separately from the remainder of the command line syntax.

**command line
optional elements**

Some command line elements are not required by the program that is being invoked. Both arguments and options can be optional with elements that are optional being enclosed in bracket [] characters in syntax descriptions.

**command line
interface**

An interface to IFT that allows you to set up and control shared access and nonshared access file transfer. The command line interface has a series of commands (and arguments) that perform various IFT tasks.

You can use the IFT command line interface either interactively or through scripts. When you use the command line interface interactively, you log on to the IFT client. You can then perform operations on that client, or open a IFT server on another host to perform file transfers or access data directly.

You can use the command line interface from any UNIX, Windows, OS/390, or z/OS system. See also *graphic user interface (GUI)*.

connector

A functional subtask in an IFT server. The connector accepts and processes any file transfer requests from an initiator subtask in the same server or in a different server. See also *initiator*.

controller ID

Controller identification number of the director the disks are channeled to for EREP usage. There is only one controller ID for Symmetrix storage subsystem.

count-key-data (CKD) A data recording format employing self-defining record formats in which each record is represented by a count area that identifies the record and specifies its format, an optional key area that may be used to identify the data area contents, and a data area that contains the user data for the record. CKD can also refer to a set of channel commands that are accepted by a device that employs the CKD recording format.

D

DASD Direct access storage device. A device that provides nonvolatile storage of computer data and random access to that data. It is analogous to an open systems hard disk.

data availability Access to any and all user data by the application.

data transfer An operation in which InfoMover File Transfer transfers data from one host to a file or pipe on another host through transfer devices on a Symmetrix storage subsystem common to both hosts.

dataset On a mainframe system, a collection of data in one of several prescribed arrangements and described by control information to which the system has access. See also *partitioned data set (PDS)* and *partitioned data set extended*.

delayed fast write There is no room in cache for the data presented by the write operation.

destage The asynchronous write of new or updated data from cache to disk device.

device A uniquely addressable part of the Symmetrix storage subsystem that consists of a set of access arms, the associated disk surfaces, and the electronic circuitry required to locate, read, and write data. See also *volume*.

device address The hexadecimal value that uniquely defines a physical I/O device on a channel path in an OS/390 and z/OS environment. See also *unit address*.

device number The value that logically identifies a disk device in a string.

device support facilities program (ICKDSF) A program used to initialize the Symmetrix storage subsystem at installation and provide media maintenance.

diagnostics System level tests or firmware designed to inspect, detect, and correct failing components. These tests are comprehensive and self-invoking.

director The component in the Symmetrix storage subsystem that allows the Symmetrix storage subsystem to transfer data between the host channels and disk devices. See also *channel director* and *disk director*.

disk director The component in the Symmetrix storage subsystem that interfaces between cache and the disk devices.

dual initiator A Symmetrix feature that automatically creates a backup data path to the disk devices serviced directly by a disk director, if that disk director or the disk management hardware for those devices fails.

dual initiated transfers A reference to the ability to have active clients on two or more hosts. For example, there could be a client on a mainframe system requesting data be transferred to a Sun system. At the same time, there could be a client on the Sun system transferring data to the mainframe system. See also *bidirectional transfers*.

duplex transfers A reference to the capability of having data transferred simultaneously in both directions. If there is a single IFT client, the clients can do both gets and puts. See also *bidirectional transfers*.

dynamic path reconnect (DPR) A function that allows disconnected I/O operations with Symmetrix storage subsystem to reconnect over any available channel path rather than be limited to the one on which the I/O operation was started.

dynamic sparing A Symmetrix feature that automatically transfers data from a failing disk device to an available spare disk device without affecting data availability. This feature supports all nonmirrored devices in the Symmetrix storage subsystem.

E

EBCDIC The acronym for extended binary-coded decimal interchange code. A byte encoding schema used on many mainframes. It is a different schema from ASCII.

EREP program The program that formats and prepares reports from the data contained in the Error Recording Data Set (ERDS).

ESCON Enterprise Systems Connection. A set of IBM and vendor products that interconnect S/390 computers with each other and with attached storage, locally attached workstations, and other devices using optical fiber technology and dynamically modifiable switches called ESCON directors.

In IBM mainframes, the local interconnection of hardware units is known as channel connection (and sometimes as local connection to distinguish it from remote or telecommunication connection).

ESCON fiber optic cabling can extend this local-to-the-mainframe network up to 60 kilometers (37.3 miles) with chained Directors.

The data rate on the link itself is up to 200 Mbps (million bits per second) and somewhat less when adapted to the channel interface. Vendor enhancements may provide additional distance and higher amounts of throughput.

ESCON director Device that provides a dynamic switching function and extended link path lengths (with XDF capability) when attaching an ESCON channel to a Symmetrix serial channel interface.

extent A continuous block of storage space on a disk that is occupied by or reserved by the operating system for a particular dataset, data space, file, or program.

F

fast write In a Symmetrix storage subsystem, a write operation at cache speed that does not require immediate transfer of data to disk. The data is written directly to cache and is available for later destaging.

FICON (fibre connectivity) A high-speed I/O interface for mainframe computer connections to storage devices. FICON channels increase I/O capacity to make them up to eight times as efficient as ESCON channels.

field replaceable unit (FRU) A component that is replaced or added by service personnel as a single entity.

file system Logical structures that allow easy access to data files residing within a disk drive, disk partition (slice), or logical volume.

fixed block architecture (FBA) A physical device format where data on disk is organized in physical blocks of fixed size, usually 512 bytes. The physical blocks are accessed by block numbers relative to block 0 on the device.

fixed blocked ASA (FBA) A data record consisting of fixed length records organized in fixed length blocks where the first byte of each record is reserved for and defines special printer control codes defined by the American National Standards Institute.

fixed length (Format-F) record formats Data record formats for mainframe datasets in which every record in a dataset is the same size. The records can be blocked or unblocked. If the records are blocked, the number of records in each block is the same for every block in the dataset. If the records are unblocked, every record is one block. The dataset can contain truncated (short) blocks. Format-F records can be used in any type of dataset.

fixed block (FB) record A fixed-length, blocked record in a mainframe dataset. The dataset can contain truncated (short) blocks. See also *fixed length (Format-F) record formats*.

fixed block standard (FBS) record A fixed-length, blocked, standard record in a mainframe dataset. Only the last block or track in the dataset can be a short block or have unfilled tracks. See also *fixed length (Format-F) record formats*.

flat file A file that consists of records of one type. A flat file has no embedded structure information governing relationships between the records.

frame Data packet format in an ESCON environment.

G

gatekeeper A small logical volume on a Symmetrix storage subsystem used to pass SCSI commands from a host to the Symmetrix storage subsystem. Gatekeeper devices are configured on standard Symmetrix disks, using six cylinders (6 MB).

GDG Generation data group. A group of related, cataloged datasets. A GDG is a collection of physical sequential files that are logically related by their content, and usually represent a history of the data over some period of time. The group is created through use of administrative utilities, and individual datasets in the group, called generation data sets (GDS), are created by running user programs.

GDS Generation data set. A member of a generation data group. A member of a generation data group is referred to by a GDG base name, followed by a number enclosed in parentheses. The number indicates the relative chronological position of the GDS.

gigabyte (GB) One gigabyte equals 1024 megabytes. Commonly interpreted as one billion bytes.

graphic user interface (GUI) An interface to InfoMover File Transfer that is available on Windows hosts. The IFT graphic user interface consists of a series of menus and screens.

You use the IFT graphic user interface to set up and control file transfers. You can also use the IFT graphic user interface to generate batch scripts that you can use to control operations with files you transfer regularly. See also *command line interface*.

H

head and disk assembly (HDA) A field replaceable unit in the Symmetrix storage subsystem containing the disk and actuator.

high-level qualifier The first qualifier in a dataset name. It usually identifies the owner of the dataset. It can be the user ID of an individual or the name of a group defined on a mainframe host.

home address (HA) The first field on a CKD track that identifies the track and defines its operational status. The home address is written after the index point on each track.

host A Used in this guide to describe the local host in a data transfer with a remote host, often referred to as host B. If the client is resident on host A, then host A is in fact a local host while the other host involved is a remote host. However, if the IFT client resides on a third host, then both hosts involved in the transfer is remote. Therefore, this guide uses the terms host A and host B instead of local and remote hosts. See also *host B*

host B Used in this guide to describe the remote host in a data transfer. See also *host A*.

hyper volume extension (HVE) The ability to define more than one logical volume on a single physical disk device making use of its full formatted capacity. These logical volumes are user-selectable in size. The minimum volume size is one cylinder and the maximum size depends on the disk device capacity and the emulation mode selected.

I

ICKDSF See Device Support Facilities program.

IDE Integrated drive electronics.

identifier (ID) A sequence of bits or characters that identifies a program, device, controller, or system.

IFT configuration file A file that contains information about devices on the system, default paths, dataset allocation parameters (OS/390 and z/OS), and time-out values for IFT. You can normally set any values you need to change in the configuration file by responding to either:

- The basic configuration questions through the installation procedure
- The custom configuration questions either through the installation procedure or through the IFT Management Utility

However, if you need to set certain configuration values, you must manually change those value by editing the configuration file. On UNIX or Windows hosts, the default name for the configuration file is `imConfig.ini`. On OS/390 and z/OS hosts, the default name for the configuration file is `IMCONFIG`.

IML Initial microcode program loading.

index marker Indicates the physical beginning and end of a track.

index point The reference point on a disk surface that determines the start of a track.

initiator A functional subtask of an IFT server. The initiator accepts and processes any file transfer request from an IFT client and connects to the connector subtask of the appropriate server to carry out the transfer. See also *connector*.

INLINES An EMC-provided host-based Cache Reporter utility for viewing short and long term cache statistics at the system console.

instance One execution or configuration of InfoMover File Transfer. When the performance requirements exceed what one IFT instance can provide, you can start a second instance with its own servers, configuration file, transfer devices, and most likely its own channels.

Additionally, you may want a second IFT instance running for operational or redundancy purposes.

I/O device An addressable input/output unit, such as a disk device.

IPL Initial program load. The process that loads the system programs, checks the system hardware, and prepares the system for user operations.

J

JCL Job control language. A scripting language used to identify the operating system requirements of a job.

JES Job entry subsystem. The primary job scheduling and printer queueing subsystem on a mainframe host.

K

kernel In a UNIX system, an operating system resident in memory.

kilobyte (K) 1024 bytes.

L

least recently used algorithm (LRU) The algorithm used to identify and make available the cache space by removing the least recently used data.

logical volume A user-defined storage device. In the Model 5200, the user can define a physical disk device as one or two logical volumes.

logical volume manager Host software that is responsible for creating host level logical storage and mapping logical storage to physical storage.

logical unit number (LUN) An encoded 3-bit identifier for the logical unit.

M

mainframe	Refers generically to an OS/390 or z/OS hardware system.
master device	Symmetrix shared logical volume used for communication between the IFT processes running on the source and the target hosts.
media	The disk surface on which data is stored.
megabyte (MB)	10 ⁶ bytes.
metavolume	Consists of multiple striped or concatenated logical volumes.
mirroring	The Symmetrix storage subsystem maintains two identical copies of a designated volume on separate disks. Each volume automatically updates during a write operation. If one disk device fails, the Symmetrix storage subsystem automatically uses the other disk device.
mirrored pair	A logical volume with all data recorded twice, once on each of two different physical devices.
mount point	The location a file system is attached (or mounted) to, usually an empty directory.

O

open system	A computer system that meets ISO Open Systems Interconnection specifications. By using the same communication protocols, open systems can communicate through a network regardless of manufacturer or model. More generally, it refers to any UNIX or Windows system.
OS/390	Pertaining to the IBM operating system that includes and integrates functions previously provided by many IBM software products (including the MVS operating system) and (a) is an open, secure operating system for the IBM S/390 family of enterprise servers, (b) complies with industry standards, (c) is Year 2000 ready and enabled for network computing and e-business, and (d) supports technology advances in networking server capability, parallel processing, and object-oriented programming.

OSRTI Open System Run Time Interface. It is a library written by EMC to mask the differences between the various versions of UNIX and Windows behind a common set of routines available on all the platforms.

P

partitioned data set (PDS) In a mainframe system, a dataset that is divided into partitions, called members, each of which can contain a program, part of a program, or data. See also *dataset*.

partitioned data set extended In a mainframe system, similar to a partitioned dataset but without a limited number of directory blocks. See also *dataset*

physical ID Physical identification number of the Symmetrix director for EREP usage. This value automatically increments by one for each director installed in the Symmetrix storage subsystem. This number must be unique in the mainframe system. It should be an even number. This number is referred to as the SCU_ID.

primary track The original track on which data is stored. See also *alternate track*.

promotion The process of moving data from a track on the disk device to cache slot.

Q

qualifier Part of a dataset name.

R

RACF Resource access control facility. The IBM-licensed external security manager, which is primarily intended to work with OS/390 or z/OS system software by way of the SAF (System Authorization Facility) interface.

read hit Data requested by the read operation is in cache.

read miss Data requested by the read operation is not in cache.

record zero The first record after the home address.

remote client An IFT client residing on a host that has no IFT server. A remote IFT client connects to IFT servers running remotely, over TCP/IP.

S

SAF	System Authorization Facility. A generic security interface in an OS/390 or z/OS system that works with an independent external security manager such as RACF.
sequential dataset	A dataset whose records are organized on the basis of their successive physical positions.
scalability	The ability of a system to do more work proportionate to the amount of additional resources provided to accomplish the task.
scrubbing	The process of reading, checking the error correction bits, and writing corrected data back to the source.
SCSI adapter	Card in the Symmetrix storage subsystem that provides the physical interface between the disk director and the disk devices. The SCSI (small computer systems interface) standard defines hardware interface and a command and data transmission protocol.
SCU_ID	For 3880 storage control emulations, this value uniquely identifies the storage director without respect to its selection address. It identifies to the host system, using the EREP, the director detecting the failing subsystem component. This value automatically increments by one for each director installed. The SCU_ID must be a unique number in the host system. It should be an even number and start on a zero boundary.
segment size	IFT's I/O transfer size specified as the number in 512 byte blocks. This parameter defines the device I/O size. The default 128 block segment size equates to 64 K-byte I/O operations to the transfer and data devices.
shared symmetrix	A term given to a Symmetrix storage subsystem that contains volumes that are shared by two or more hosts connected to the Symmetrix storage subsystem.
shell	In a UNIX system, the command and script interpreter.
short miss	The requested data is not in cache, but is in the process of being fetched.
source	A file or dataset that IFT copies to another location.

SSID	For 3990 storage control emulations, this value identifies the physical components of a logical DASD subsystem. The SSID must be a unique number in the host system. It should be an even number and start on a zero boundary.
stage	The process of writing data from a disk device to cache.
storage control unit	The component in the Symmetrix storage subsystem that connects the Symmetrix storage subsystem to the host channels. It performs channel commands and communicates with the disk directors and cache. See also <i>channel director</i> .
string	A series of connected disk devices sharing the same disk director.
SMS	System managed storage.
T	
target	SCSI device that performs an operation that was originated by an initiator SCSI device. The target is usually a peripheral device or controller (disk, tape, CD, and so on). The initiator is usually a host computer. Each device has a unique SCSI address (ID). A target may have a maximum of eight logical units (LUs) in SCSI-2 and 64 LUs in SCSI-3.
TID	Target ID.
thin client	An IFT client that resides on a system that does not have a resident server. A thin client is possible only on an open systems (UNIX or Windows) host that is connected to the rest of the IFT network by TCP/IP.
throughput	The amount of work accomplished in a given amount of time. Throughput for IFT generally refers to the total amount of data transferred between two or more systems in a given period of time (usually the beginning of the first transfer to the end of the last).
transfer devices	Symmetrix logical volumes that are shared between the source and target systems as an intermediary holding area for data.
translation files	A file on an open systems host that contains a single set of characters that IFT uses to make a character-by-character translation of text data when transferring that data from one host to another.

Each translation table contains hexadecimal representations of each character in the character set to be translated. Each can also contain protocol information that IFT needs to perform data operations.

You specify the default translation files through the custom configuration process.

translation tables

Sets of characters included in a single dataset on a mainframe host that IFT uses to make a character-by-character translation of text data when transferring that data from one host to another.

Each translation table contains hexadecimal representations of each character in the character set to be translated. You specify the default translation tables through the custom configuration process.

TSO Time-sharing option. Used in OS/390 and z/OS systems.

U**unit address**

The hexadecimal value that uniquely defines a physical I/O device on a channel path in a mainframe environment. See also *device address*.

V**verbose mode**

A mode available in the CLI client in which the CLI client echoes any command you enter and displays detailed information about the results of any CLI command you enter. You can enable or disable verbose mode with the **verbose** CLI command.

volume

A general term referring to a storage device. In the Symmetrix storage subsystem, a volume corresponds to a single device visible to the host. In the context of host-based, volume manager software, a volume is a logical unit of disk storage that may comprise storage on one to many Symmetrix devices.

W**wrap mode**

A mode used in open systems to mainframe transfers. In wrap mode, IFT wraps any open systems records longer than 72 characters to meet line length requirements on a mainframe system.

write hit

There is room in cache for the data presented by the write operation.

write miss There is no room in cache for the data presented by the write operation.

Z

z/OS The computer operating system for IBM's zSeries 900 (z900) line of large mainframe servers. It is a renamed and upgraded version of OS/390, which in turn evolved from the MVS operating system. IBM's renamed servers and operating systems reflect a strategy to realign its products more closely with the Internet and its own e-business initiatives.

It is described as an extremely scalable and secure high-performance operating system based on the 64-bit z/Architecture. Like its predecessor, OS/390, z/OS lays claim to being highly reliable for running mission-critical applications. The operating system supports Web- and Java-based applications.

A

- absolute names
 - datasets 4-75, 4-77
 - GDGs B-8, B-9
- additional instances, supported with UNIX 1-9
- additional instances, unsupported with Windows 1-9
- advanced switches
 - blocksize 2-19
 - current unit switch 2-18
 - end-of-line conversion 2-18
 - overwrite target 2-18
 - rec LF appending 2-18
 - record format 2-19
 - record length 2-19
 - release unused space on deallocation 2-18
 - save trailing blanks 2-18
 - target mainframe volume 2-18
 - template data set 2-18
 - text data uppercasing 2-18
 - transfer RDWs 2-18
 - wrap mode 2-18
- architecture
 - client 1-5
 - IFT server 1-3
 - IFT servers 1-3
 - overview 1-3
- automatic space deallocation (OS/390 and z/OS) 4-54

B

- base names, for GDGs B-7
- batch files 3-19

- batch files, automating CLI commands 3-19
- batch job, relative generation number B-9
- batch jobs in IFT 1-2
- batch mode, CLI 3-2
- BCVs, access to open systems during mainframe operations 1-10
- bidirectional transfers, definition g-1
- binary command 4-4, 4-63
- binary mode
 - with binary command 4-4
 - with imrdw command 4-50
- blanks, imtrail (OS/390 and z/OS) command 4-57
- BLKSIZE attribute B-15
- blksize command 4-5, B-16
- blocksize switch 2-19
- browsing 2-24
 - session name menu 2-9
 - workspace tab 2-6, 2-8, 2-9, 2-11, 2-12, 2-13, 2-15, 2-17
- bundling log files, with IM Management Utility 1-6

C

- cd command
 - default directory 4-7, 4-64
 - parameter values 4-7, 4-64
 - purpose 4-7
 - SYSIN control statements B-5
- CLI
 - capabilities 1-6
 - command reference 4-3
 - command syntax 3-2

- establishing a connection 3-10
- interactive mode 3-3
- methods of access 3-2
- options 3-10
- OS/390 and z/OS starting procedure on
 - client 3-4
- overview 3-2
- starting client 3-4
- stopping client 3-17
- summary of commands 3-27 to 3-32
- supported hosts 3-2
- verbose mode 3-10
- CLI commands, status switches 2-18, 2-19
- CLI commands, using 3-1
- client
 - exiting from 4-98
 - location for transfers 3-11, 3-12
 - log file 4-48
 - server connection 3-11
 - starting 3-4
 - starting from IFT Command window 3-8, 3-9
 - starting from Start menu 3-8
 - starting from UNIX using CLI 3-7
 - starting using CLI 3-4
 - starting using TSO CALL command 3-5, 3-6
 - starting with INFOMOVR CLIST 3-6
 - stopping using CLI 3-17
 - thin 3-12, g-14
- client dialog logging 4-48, 4-49
- client log file 4-48
- client logging
 - disabling 4-48
 - enabling 4-48
- client passwords in scripts 2-24
- client security 1-11
- client, characteristics 1-5
- client, starting from Windows using GUI 2-3
- client_timeout, configuration parameters 3-18
- client-server connections 3-11
- close command 3-17, 3-20, 4-11, 4-87
- commands
 - blksize B-16
 - cd 4-7, B-5
 - close 3-20, 4-11, 4-87
 - del 4-12
 - dir 4-14 to 4-22, 4-73, B-5
 - get 3-15, 4-23 to 4-28, 4-75, 4-101, 4-103, B-2, B-3, B-5, B-9
 - help 4-30
 - imaddlf 4-32
 - imbufno 4-35
 - imCrLf 4-37, 4-38
 - imcylinder 4-40, 4-41, 4-42, B-15
 - imdirblocks 4-44, B-16, B-26
 - imfilenumber 4-46, B-16, B-26
 - imlog 3-28, 4-48, 4-49
 - imrdw 3-29, 4-50, 4-51
 - imrecvx 3-29, 4-52, 4-55, B-6
 - imrelease 4-54, B-16
 - imsendx 4-55, B-5
 - imtrail 3-29, 4-57
 - imwrap 3-29, 4-59
 - imxlate 3-29, 4-57
 - lcd 4-64
 - likesdn 3-30, 4-75, 4-76, 4-77, 4-104, B-14
 - lrecl 4-79, B-16, B-26
 - lsp 3-30, 4-80
 - mainframe CANCEL A-27
 - mget 3-15, 4-81
 - mput 3-15, 4-84
 - open 3-31, 4-87, 4-88
 - prompt 4-89
 - put 3-15, 3-31, 4-27, 4-75, 4-90 to 4-95, 4-101, 4-103, B-2, B-5, B-9
 - pwd 3-31, 4-97
 - quit 3-17, 3-31, 4-31, 4-98
 - recfm 3-31, 4-99, B-16, B-26
 - replace 3-31, 4-26, 4-101
 - replace ON 4-92, 4-94
 - status 3-31, 4-4, 4-44, 4-46, 4-51, 4-54, 4-57, 4-63, 4-77, 4-103, 4-109, 4-110, 4-113, 4-115
 - summary list 3-27 to 3-32
 - unit 3-32, 4-107, 4-115, B-16, B-26
 - upper 4-110
 - verbose 3-32, 4-112
 - verbose OFF 3-10
 - volcount B-16
 - volume 3-32, 4-102, 4-115, B-16
- configuration files
 - creating with Management Utility 1-6
 - modifying 1-6
- configuration parameters

- client_timeout 3-18
- default_bufno B-14
- default_cylinders 4-40, B-14
- default_imbufno 3-28, 4-35
- default_imdirblocks B-14
- default_imfilenumber B-14
- default_imrelease B-14
- default_lrecl 3-30, 4-79, B-14
- default_recfm B-14
- default_secondary_cylinders B-14
- default_unit B-14
- global_disabled command 4-12, 4-27, 4-94
- imfilenumber 4-46
- magnetic tape B-24
- magnetic tape files B-26
- OS/390 and z/OS datasets B-13
- recfm B-14
- release B-14
- user_disabled_command 4-12, 4-27, 4-94
- verbose_server_messages A-3
- connections
 - client-server 3-11
 - closing 3-17
 - concurrent 1-9, 3-16
 - establishing using CLI 3-10
- connections, establishing 3-10
- connector, IFT server component 1-3
- copying data, IFT capabilities 1-2
- current unit switch 2-18
- Customer Support Center
 - sending documentation C-6
 - troubleshooting a problem C-3
- cylinder allocation 4-103
- D**
- data
 - control block 3-28, 4-35
 - conversions. See character translation.
 - incomplete or invalid event A-10
 - transferring datasets and files 3-15
- dataset types supported for transfer 2-19
- datasets
 - allocation attributes B-13
 - default attributes 4-26, 4-92
 - default position on tape B-14
 - magnetic tape B-24
 - magnetic tape allocations B-25
 - names (relative and absolute) B-9
 - serialization B-2
 - SMS B-17
 - supported types (OS/390 and z/OS) 2-19
 - transferring (OS/390 and z/OS) 2-19
 - transferring from mainframe 3-14
 - types supported 1-7
- DCB
 - attributes 4-77
 - buffer size, default B-14
 - dsname parameter 4-76
- default
 - block size of target datasets 4-92
- default translation files 2-16
- default_blksize configuration parameter 4-5
- default_bufno configuration parameter B-14
- default_cylinders configuration parameter 4-40, B-14
- default_imbufno configuration parameter 3-28, 4-35
- default_imdirblocks configuration parameter B-14
- default_imfilenumber configuration parameter 4-46, B-14
- default_imrelease configuration parameter B-14
- default_lrecl configuration parameter 3-30, 4-79, B-14
- default_recfm configuration parameter B-14
- default_secondary_cylinder configuration parameter 4-40
- default_secondary_cylinders configuration parameter 4-40, B-14
- default_unit configuration parameter B-14
- del command 4-12
 - synonyms 4-12
- devices, transfer 1-4
- diagnostic information, viewing with
 - Management Utility 1-6
- dialog client logging 4-48
- dir command 4-14 to 4-22, B-5
 - memory usage 4-18
 - redirection symbol 4-14
 - with GDGs B-11
- directory
 - block defaults B-14
 - limitations on deleting 4-12

disabled get commands 4-27
 disabled put commands 4-12, 4-94
 disabling the get command on some hosts 4-12,
 4-27, 4-94
 disposition status, MOD B-2, B-3
 dual initiated transfers, definition g-5

E

edit pulldown menu 2-9
 embedded spaces, in Windows directory names
 4-8, 4-18, 4-65, 4-71
 EMC technical publications v-xiv
 EMC TimeFinder/Mirror support 1-10
 enable translation check box 2-15
 end-of-line conversion switch 2-18
 EOL conversion mode 4-37
 error
 message severity codes A-3
 messages A-2
 messages of incomplete transfer 3-16
 severity codes A-3
 ESCON channels (IFS) 1-5
 ESCON channels, definition g-6
 establishing a connection 3-10
 event messages
 sending data to IFT client A-10
 termination in middle of transfer A-11
 events
 logging to file A-6
 message format A-2
 summary of messages A-7
 existing sessions list 2-8
 extents
 primary 4-42
 secondary 4-42

F

Fast Wide SCSI channels (IFS) 1-5
 fatal severity code A-3
 Fibre Channels (IFS) 1-5
 FICON channels (IFS) 1-5
 file
 replacement mode 4-101, 4-102, 4-103
 transferring using transfer confirmation
 window 2-14
 transferring using workspace tab 2-13

file input mode, CLI 3-2
 file pulldown menu 2-9
 file structure, OS/390 and z/OS 3-15
 file transfer
 incomplete 3-16
 saving as scripts 2-23
 transfer devices 1-3
 file transfers
 concurrent operations 1-9
 OS/390 and z/OS flat files 1-7
 file translation
 options 2-15
 performing 2-15
 performing on mainframe systems 3-13
 file translations
 performing on mainframe systems 1-10
 files
 batch 3-19
 flat 1-7
 transfers 3-20
 fixed block architecture, transfers 1-3
 flat files
 on disk 1-7
 on magnetic tape 1-7
 physical sequential data sets (PSDS) formats
 1-7

G

GDG

absolute names B-9
 base names B-7
 datasets, relative generations B-9
 magnetic tape B-24
 relative names B-8
 SMS B-19
 sources and targets 4-26
 usage with dir command B-11
 generating scripts 2-23
 generating scripts tab 2-23
 generation
 data groups 4-26, 4-75, 4-92, B-7
 data groups (GDG) g-7
 dataset, (GDS) types, definition g-8
 generation datasets (GDS) flat files 1-8
 get command 3-15, 4-23 to 4-29, 4-75, 4-92, 4-101,
 B-3, B-5

- disabling from hosts 4-12, 4-27, 4-94
 - source specifications 4-25
 - target specifications 4-24
 - with GDGs B-9
 - get commands
 - disabled for hosts 4-27
 - disabled for users 4-27
 - global_disabled_command configuration
 - parameter 4-12, 4-27, 4-94
 - go to text box 2-9
 - graphical user interface, IFT capabilities 1-2
 - GUI
 - definition g-8
 - error and event messages A-5
 - generate script tab 2-23
 - generating scripts 2-23
 - platform supported scripts 3-22
 - steps to create script 3-26
 - transfer table tab 2-21
 - workspace tab 2-4
 - GUI capabilities 1-6
 - GUI purpose 2-23
 - GUI, supported platforms 1-6
- H**
- help command 4-30
 - Hierarchical File System (HFS), OS/390 and z/OS
 - flat files 1-8
 - high-level qualifier (HLQ), definition g-8
 - host A definition, valid client locations 3-12
 - host A, definition 3-12
 - host B definition, valid client locations 3-12
 - host b, definition 3-12
 - host IP address
 - GUI script 3-26
 - session detail parameter 2-8
 - host IP address parameter 2-8
 - host type parameter 2-7
 - hosts, supported using CLI 3-2
- I**
- IDCAMS utility B-9
 - IFT
 - capabilities 1-2
 - CLI commands 3-1, 4-1
 - command window 3-9
 - configuration file 4-48
 - default directory
 - using the cd command 4-7
 - using the lcd command 4-64
 - definition of configuration file g-9
 - GUI error display window A-5
 - language environment multi-tasking facility A-27
 - migration path 1-2
 - servers 1-3, 3-7, 3-9, 3-26
 - starting interactive CLI operations 3-3
 - transfers 1-3
 - IFT capabilities, CLI 1-6
 - IFT client
 - source file 1-3
 - target file 1-3
 - IFT Management Utility
 - failure A-11
 - IFT Management Utility main menu
 - querying Symmetrix devices 3-23, 3-24
 - IFT servers, starting 2-6, 2-8, 2-9, 2-11, 2-12, 2-13, 2-15, 2-17
 - IFT with TimeFinder 1-10
 - imaddlf command 4-32
 - synonyms 4-32
 - imbufno command 4-35
 - IMCONNR member
 - error message A-15
 - imcrlf command 4-33, 4-37, 4-38
 - imcylinder command 4-40, 4-41, 4-42, 4-76, B-15, B-16
 - imdirblocks command 4-44, B-16
 - imfilenumber command 4-46, B-16, B-26
 - IMINTR member
 - error message A-15
 - imlog command 3-28, 4-48, 4-49
 - imrdw command 3-29, 4-50, 4-51
 - imrecvx command 3-29, 4-52, B-5
 - imrelease command 4-54, B-16
 - imsendx command 3-29, 4-56, B-5
 - imtrail command 3-29, 4-57
 - imwrap command 3-29, 4-59
 - imxlate command 3-29, 4-4, 4-57, 4-61
 - imXlate_local_remote configuration parameter 4-61
 - INFOMOVR CLIST, IFT client 3-6
 - information severity code A-3

initiator member, definition g-9
 initiator, IFT server component 1-3
 instance
 definition g-10
 instances
 concurrent connections 1-9
 instances, supported hosts 1-9
 interactive mode, CLI 3-2, 3-3
 interactive prompting
 controlling 4-89
 using prompt command 4-89
 with OS/390 and z/OS 4-89
 interface
 CLI 3-2
 GUI 2-2
 invalid target names 4-25
 ISO characters, OS/390 and z/OS datasets 4-33, 4-57

J

Java runtime environment
 as GUI prerequisite 2-2
 installing on Windows 2-2
 Java runtime environment, as GUI prerequisite 2-2
 JCL continuation character B-4

L

language environment multi-tasking facility
 event messages A-27
 ldir command 4-68
 likedsn command 4-75
 client command 3-30
 configuration parameters B-14
 model datasets 4-75
 with status command 4-104
 linefeeds
 appending to OS/390 and z/OS sources 4-32
 using imaddlf command 4-32
 local server, using close command 4-11
 log
 events A-6
 files 4-103
 LRECL (record length) 4-76, B-15
 lrecl command
 configuration parameter B-14

magnetic tape source datasets B-26
 overriding allocation attributes B-16
 purpose 4-79
 lsp command 3-30, 4-80

M

magnetic tape files
 as source or target files 4-25
 configuration parameters B-24, B-26
 dataset allocations B-25
 datasets B-24
 mounting B-25
 OS/390 and z/OS flat files 1-8
 OS/390 and z/OS systems 4-93
 source and target datasets 1-4
 mainframe CANCEL command A-27
 mainframe GUI sessions
 correct text box 2-5
 mainframe sessions 2-9, 2-10
 Management Utility, functionality 1-6
 Management Utility, IFT 1-6
 memory, using dir command 4-18
 mget command 3-15, 4-81, 4-89
 mirrored volume VOLSERs, changing VOLSERs
 on mirrored volumes 1-10
 MOD disposition status B-3
 mput command 3-15
 pattern 4-84
 patterns 4-81
 purpose 4-84

N

named pipes
 as source or target files 4-25, 4-93
 transferring 1-7
 nonshared access transfers
 definition g-4
 with CLI g-3

O

open command
 client command 3-31
 purpose 4-87
 using 4-88
 open systems

- definition g-11
- imlog command 4-48
- imrecvx command 4-52, 4-55
- imsendx command 4-55
- imxlate command 4-61
- open systems GUI sessions
 - correct text box 2-5
- opening sessions, sequence 2-9
- OS/390 and z/OS
 - batch interactive prompting 4-89
 - batch transfers 3-20
 - cancelled jobs A-27
 - connector log 4-19
 - imaddlf command 4-32
 - imbufno command 4-35
 - imcylinder command 4-40
 - imdirblocks command 4-44
 - imrdw command 4-50
 - likedsn command 4-75, 4-76, 4-77
 - lrecl command 4-79
 - lsp command 4-80
 - recfm command 4-99
 - source files 4-32
 - SYSIN DD statements B-5
 - target dataset allocation attributes B-13
 - target files 4-32
 - transferring mainframe datasets 2-19
- OS/390 and z/OS datasets
 - configuration parameters B-13
 - containing ISO characters 4-33, 4-57
 - IFT default B-13
- OUTDISP, abnormal disposition subparameter A-27
- overwrite target switch 2-18

P

- packed decimal transfers, OS/390 and z/OS flat files 1-8
- parallel data operations, concurrent operations 1-9
- parameters
 - host IP address 2-8
 - primary server port 2-8
 - server name 2-7
 - session details 2-7
 - session name 2-7

- user login 2-7
- partitioned data set (PDS)
 - definition g-12
 - extended 4-44
 - using imdirblocks command 4-44
 - within GDG B-7
- partitioned datasets (PDS)
 - OS/390 and z/OS flat files 1-7
- password characters, starting IFT client 3-6
- passwords, in script files 1-6
- platforms
 - GUI 1-6
 - starting CLI client 3-4
- platforms, supported 1-2
- port numbers 3-26
- primary extents, imcylinder command 4-42
- primary server port parameter 2-8
- primary server port, session detail parameter 2-8
- prompt command 4-89
- prompting
 - controlling 4-81, 4-89
 - with mget and mput 4-89
- put command 3-15, 3-31, 4-75, 4-90 to 4-95, 4-101, 4-103, B-5
 - with GDG B-9
- put commands
 - disabled for hosts 4-12, 4-94
 - disabled for users 4-12, 4-94
- pwd command 3-31, 4-97

Q

- query Symmetrix devices menu 3-23, 3-24
- quit command 3-17, 3-31, 4-31, 4-98

R

- RACF, definition g-12
- raw devices, in file transfers 1-7
- RDW retention
 - disabling 4-50
 - enabling 4-50
 - using imrdw command 4-50
- rec LF appending switch 2-18
- RECFM (record format)
 - allocation attributes B-15
 - SMS inactive B-15
 - using likedsn command 4-76

- recfm command 3-31, 4-99, B-16, B-26
- RECFM=U datasets, transferring 4-100
- record
 - descriptor words 4-50
 - line feeds 4-103
- record format switch 2-19
- record length switch 2-19
- record/span descriptor words 4-50
- recovery from transfer failure 3-16
- relative
 - dataset names 4-77
 - generation indicator (GDG) B-10
 - names in GDGs B-8
- relative generation number, batch job B-9
- release unused space on deallocation switch 2-18
- remote servers, with close command 4-11
- replace command 3-31, 4-26, 4-101
- replace ON command 4-92, 4-94
- replacement mode
 - replace command 4-101
 - status command 4-103

S

- save trailing blanks switch 2-18
- save windows
 - for logging script output 2-24
- saving scripts to a file 2-24
- script files
 - creating with Management Utility 1-6
 - encrypted passwords 1-6
- scripts
 - automating CLI commands 3-19
 - client passwords 2-24
 - creating manually 3-19
 - GUI 3-22
 - security 3-20
 - UNIX 3-20
 - user IDs 2-24
 - Windows systems 3-22
- scripts, saving to a file 2-24
- SDW (span descriptor words) 4-50
- secondary cylinder default value B-14
- secondary extents, imcylinder command 4-42
- security
 - IFT features 1-11
 - manually created scripts 3-20
- segment size, definition g-13
- serious severity code, error messages A-3
- server
 - client connection 3-11
 - components 1-3
 - event log file 4-48
 - stopping A-27
- server name parameter 2-7
- server names, with open command 4-87
- server shutdown messages
 - managing A-6
 - on AIX, HP-UX, and Solaris A-6
 - on Windows A-6
- server startup messages
 - managing A-6
 - on AIX, HP-UX, and Solaris A-6
 - on Windows A-6
- servers, starting 2-6, 2-8, 2-9, 2-11, 2-12, 2-13, 2-15, 2-17
- session name menu, browsing sessions 2-9
- session name parameter 2-7
- session name, session detail parameter 2-7
- sessionLogFile 4-48
- sessions
 - detail parameters 2-7
 - workspace tab 2-6
- severity code
 - error A-3
 - fatal A-3
 - information A-3
 - serious error A-3
 - warning A-3
- SMS
 - environments 4-76
 - GDG datasets B-19
 - sequential and PDS datasets B-17
 - using likedsn command B-15
- source files
 - IFT client 1-3
 - using get command 4-23
 - using put command 4-90
- source sessions 2-21
- SPACE (space allocation) 4-76, B-15
- spaces
 - embedded 4-8, 4-15, 4-17, 4-65, 4-68, 4-70
 - in Windows directory names 4-8, 4-9, 4-14, 4-18, 4-65, 4-66, 4-71

- in Windows filenames 4-8, 4-9, 4-18, 4-27, 4-65, 4-66, 4-71, 4-93
 - start menu, starting a client 3-9
 - starting
 - CLI client 3-4
 - client from IFT Command window 3-8
 - clients on UNIX 3-7
 - UNIX client 3-7
 - starting IFT servers 2-6, 2-8, 2-9, 2-11, 2-12, 2-13, 2-15, 2-17
 - status command 3-31, 4-54, 4-57, 4-63, 4-103, 4-110, 4-115
 - status switches, CLI commands 2-18, 2-19
 - striped datasets, OS/390 and z/OS flat files 1-8
 - supported datasets 2-19
 - supported platforms 1-2
 - switches
 - status 2-18
 - Symmetrix device databases
 - creating 1-6
 - Management Utility 1-6
 - updating 1-6
 - Symmetrix devices, initializing through
 - Management Utility 1-6
 - syntax, CLI 3-2
 - SYSIN DD statements B-5
- T**
- target
 - dataset default block size 4-26, 4-92
 - names 4-25
 - permissions 4-92
 - specifications with get command 4-24, 4-91
 - target file
 - get and put commands 4-101
 - get command 4-23
 - IFT client 1-3
 - permission settings 4-26, 4-92
 - using put command 4-90
 - target MVS volume switch 2-18
 - target sessions 2-21
 - technical support C-3
 - template data switch 2-18
 - text data uppercasing switch 2-18
 - text files, imsendx command 4-55
 - thin client, valid client locations 3-12
 - TimeFinder support, IFT capabilities 1-10
 - to save scripts to a file 2-24
 - tools pulldown menu 2-9
 - trailing blanks
 - intrail (OS/390 and z/OS) command 4-57
 - status command 4-103
 - with intrail command 4-57
 - transfer confirmation window 2-14
 - transfer devices, characteristics 1-4
 - transfer devices, definition 1-4, g-14
 - transfer failure recovery 3-16
 - transfer RDWs switch 2-18
 - transfer settings window 2-17
 - transfer states
 - failure 2-21
 - success 2-21
 - transfer status settings, setting advanced 2-17
 - transfer table tab
 - displayed information 2-21
 - transfer tags 2-14, 2-21
 - transfers
 - checking and setting conditions 3-13
 - concurrent 3-16
 - data 3-14
 - description 1-3
 - incomplete 3-16
 - of file types 1-7
 - times 4-27
 - using wildcards 4-81
 - valid client locations 3-11, 3-12
 - verification messages 4-27
 - translating data, IFT capabilities 1-2
 - translation
 - performing on mainframe systems 3-13
 - translation file
 - creating with Management Utility 1-6
 - extensions 4-52, 4-55
 - names 4-52, 4-55
 - tables 3-29
 - using status command 4-103
 - valid 2-16
 - translations
 - performing on mainframe systems 1-10
 - TSO CALL command
 - starting client 3-5, 3-6

U

- Ultra SCSI channels (IFS) 1-5
- unit command 3-32, 4-107, 4-115, B-16, B-26
- unit default value B-14
- universal naming conventions (UNC) 4-24, 4-93
- UNIX
 - manually created scripts 3-20
 - starting client 3-7
- UNIX System Services files, invalid OS/390 and z/OS dataset 1-8
- upper CLI command 4-110
- upper command 3-32, 4-110
- uppercase
 - conversion using upper command 4-110
 - mode with status command 4-103
 - mode with upper command 4-110
- uppercasing text data, as a performance issue 4-110
- user IDs in scripts 2-24
- user login parameter 2-7
- user login, session detail parameter 2-7
- user password parameter 2-7
- user_disabled_command configuration parameter 4-12, 4-27, 4-94

V

- valid translation files, in GUI transfers 2-16
- variable format datasets 4-50
- verbose
 - messaging 4-103, 4-116
 - mode 4-4
- verbose command 3-32, 4-112
- verbose mode
 - CLI interface 3-10
 - definition A-5, g-15
 - disabling 3-10
 - meaning 3-10
- verbose_server_messages, configuration parameter A-3, A-5
- volcount command 4-113, B-16
- volsers
 - lsp (OS/390 and z/OS) command 4-80
 - volcount (OS/390 and z/OS) command 4-108, 4-113
 - volume (OS/390 and z/OS) command 4-115
- volume command 3-32, 4-46, 4-102, 4-115, B-16

W

- warning severity code A-3
- wildcards in transfers
 - mget command 4-81
 - mput command 4-84
- Windows Event Viewer
 - logged event files A-6
 - seeing messages A-6
 - with IFT messages A-6
- working file directory
 - editing 4-7, 4-64
 - with pwd command 4-97
- workspace tab
 - browsing sessions 2-8
 - functions 2-4
 - overview 2-4
 - performing file translation 2-15
 - searching sessions 2-9
 - transferring files 2-13
- wrap mode
 - definition g-15
 - imwrap command 4-59
- wrap mode switch 2-18

Z

- z/OS, definition g-16