

IBM® DB2 Universal Database™



# Data Links Manager Administration Guide and Reference

*Version 8*



IBM® DB2 Universal Database™



# Data Links Manager Administration Guide and Reference

*Version 8*

Before using this information and the product it supports, be sure to read the general information under *Notices*.

This document contains proprietary information of IBM. It is provided under a license agreement and is protected by copyright law. The information contained in this publication does not include any product warranties, and any statements provided in this manual should not be interpreted as such.

You can order IBM publications online or through your local IBM representative.

- To order publications online, go to the IBM Publications Center at [www.ibm.com/shop/publications/order](http://www.ibm.com/shop/publications/order)
- To find your local IBM representative, go to the IBM Directory of Worldwide Contacts at [www.ibm.com/planetwide](http://www.ibm.com/planetwide)

To order DB2 publications from DB2 Marketing and Sales in the United States or Canada, call 1-800-IBM-4YOU (426-4968).

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© **Copyright International Business Machines Corporation 1998 - 2002. All rights reserved.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

---

# Contents

<b>Tables</b> . . . . .	<b>vii</b>	Authorizing users to perform actions . . . . .	33
<b>About this book</b> . . . . .	<b>ix</b>	Basic DLFM configuration tasks . . . . .	37
Who should read this book . . . . .	ix	Basic DLFM configuration tasks: details . . . . .	38
How this book is structured . . . . .	x	Creating additional Data Links File Systems on a Data Links server . . . . .	38
Conventions . . . . .	x	Creating additional Data Links File Systems on a Data Links server: steps . . . . .	40
<b>Part 1. Overview</b> . . . . .	<b>1</b>	Modifying a Data Links File System . . . . .	45
<b>Chapter 1. Introduction to DB2 Data Links Manager</b> . . . . .	<b>3</b>	Changing the size of a Data Links File System: tasks. . . . .	46
What's new in DB2 Data Links Manager Version 8 . . . . .	3	Adding DB2 databases to a DLFM on a Data Links server . . . . .	48
DB2 Data Links Manager . . . . .	4	Adding DLFMs to a specific DB2 database on the DB2 host. . . . .	50
DB2 Data Links environment. . . . .	5	Dropping Data Links Manager from a specific DB2 database on the DB2 host . . . . .	53
Data Links server components . . . . .	6	Enabling the Data Links Manager Replication daemon . . . . .	54
DB2 clients in the Data Links Manager environment . . . . .	8	Configuration tuning . . . . .	56
DB2 servers in the Data Links Manager environment . . . . .	8	Configuration tuning: details . . . . .	56
File system paradigm . . . . .	9	Synchronizing system clocks . . . . .	57
DB2 Data Links Manager and your applications . . . . .	11	Ensuring sufficient DB2 log space for the DLFM_DB . . . . .	57
DATALINK data type. . . . .	12	Setting the number of copy processes . . . . .	58
DB2 Data Links Manager typical setups. . . . .	15	<b>Chapter 3. Working with the Data Links File Manager</b> . . . . .	<b>61</b>
<b>Part 2. Administration Guide and Reference</b> . . . . .	<b>19</b>	Basic operations: starting, stopping, and restarting the DLFM . . . . .	61
<b>Chapter 2. System management options</b> . . . . .	<b>21</b>	Restarting the DLFM after an abnormal termination . . . . .	62
Data Links server file backups . . . . .	21	Monitoring the DLFM background processes (AIX, Solaris Operating Environment) . . . . .	63
Data Links server file backups: details . . . . .	22	Monitoring the DLFM background processes (Windows) . . . . .	64
Backup storage location . . . . .	22	Enabling and registering file systems with DLFM (AIX, Solaris Operating Environment) . . . . .	65
Size considerations. . . . .	23	Registering a drive with DLFF (Windows operating systems). . . . .	66
Using Tivoli Storage Manager as an archive server (AIX) . . . . .	24	Listing registered file systems (AIX, Solaris Operating Environment) . . . . .	68
Using Tivoli Storage Manager as an archive server (Solaris Operating Environment) . . . . .	26	Listing registered drives (Windows operating systems) . . . . .	68
Using Tivoli Storage Manager as an archive server (Windows) . . . . .	28	Loading, querying, and unloading a DLFF (AIX) . . . . .	69
Using a local directory for the archive area	30		
Using an XBSA-compliant storage manager	31		

Loading, querying, and unloading a DLFF (Solaris Operating Environment) . . . . .	70
Registering, querying and de-registering a DLFF (Windows operating system) . . . . .	71
Increasing a DLFF-controlled file system size (AIX, Solaris Operating Environment) . . . . .	72
Registering DB2 Universal Databases with the Data Links server . . . . .	73
Listing databases registered with the Data Links server . . . . .	74
Removing information about a DB2 database from Data Links Manager . . . . .	75
Changing the diagnostic level for error message log files . . . . .	76
Modifying logging for Data Links Filesystem Filter (DLFF) processing (AIX) . . . . .	77
Logging messages after the DLFF driver has been loaded (AIX) . . . . .	79
Minimizing logging for Data Links Filesystem Filter (DLFF) processing (Solaris Operating Environment) . . . . .	79
Modifying logging for Data Links Filesystem Filter (DLFF) processing (Windows operating systems) . . . . .	80
Migrating a DLFF-enabled file system to a different hard disk—overview . . . . .	80
Migrating a DLFS to a different hard disk: details . . . . .	82
Migrating a DLFS to a different hard disk (AIX) . . . . .	82
Migrating a DLFS to a different hard disk (Solaris Operating Environment) . . . . .	86
Migrating a DLFS to a different drive (Windows) . . . . .	89
Using SCOPY instead of COPY . . . . .	91
Retrieving archive server backup file information . . . . .	93
<b>Chapter 4. Security . . . . .</b>	<b>95</b>
Introduction to Data Links Manager security	95
Built-in security features . . . . .	95
Data access security features . . . . .	96
Data access security features: details . . . . .	97
Basic linked file security controls . . . . .	97
Advanced file management security features . . . . .	98
Advanced file management security controls: detail . . . . .	101
Read operation security . . . . .	105
Write operation security . . . . .	105

Securing linked files during updates: issues . . . . .	107
--------------------------------------------------------	-----

---

## Part 3. Application Programmer's Reference . . . . . 109

<b>Chapter 5. Updating linked files . . . . .</b>	<b>111</b>
Updating approaches overview . . . . .	111
Approach 1: Unlink/update/relink files . . . . .	113
Approach 1: procedure . . . . .	114
Unlinking and relinking files . . . . .	114
Approach 2: Update linked files/native file system defines access . . . . .	115
Approach 3: Update linked files/customize file access privileges . . . . .	116
Approach 3: details . . . . .	118
Approach 3 configuration tasks . . . . .	118
Approach 3 maintenance tasks . . . . .	120
Approach 3 application program flow . . . . .	122
Backing out updates in progress . . . . .	126
Usage considerations for write tokens . . . . .	128
Recovering missing write tokens . . . . .	129
Approach 4: Replace a linked file's contents	131
Approach 4: details . . . . .	132
Replacing a linked file's contents using approach 4 . . . . .	132
Approach 4 Usage Scenario . . . . .	133
Updating approaches summary . . . . .	134

---

## Part 4. Troubleshooting . . . . . 139

<b>Chapter 6. Server recovery . . . . .</b>	<b>141</b>
Data Links server machine failure . . . . .	141
Failure and recovery overview . . . . .	141
DB2 Data Links Manager system setup and backup recommendations . . . . .	143
File system backup and restore recommendations . . . . .	143
Bringing the file system directory hierarchy to the current point in time . . . . .	144
db2_recon_aid utility. . . . .	146
DB2 Data Links Manager recovery scenarios	148
<b>Chapter 7. Data Links File Manager error message reference . . . . .</b>	<b>155</b>
DLFM001I . . . . .	157
DLFM101E . . . . .	157
DLFM121E . . . . .	158
DLFM122E . . . . .	158

DLFM124E . . . . .	158
DLFM126E . . . . .	158
DLFM128E . . . . .	159
DLFM129I . . . . .	159
DLFM151I . . . . .	159
DLFM201E . . . . .	160
DLFM202E . . . . .	160
DLFM203E . . . . .	160
DLFM204E . . . . .	161
DLFM205E . . . . .	161
DLFM206E . . . . .	161
DLFM207E . . . . .	161
DLFM208I . . . . .	161
DLFM209E . . . . .	162
DLFM210E . . . . .	162
DLFM211E . . . . .	162
DLFM221E . . . . .	163
DLFM222I . . . . .	163
DLFM223E . . . . .	163
DLFM252E . . . . .	164
DLFM254I . . . . .	164
DLFM341E . . . . .	164
DLFM402E . . . . .	164
DLFM501E . . . . .	165
DLFM701E . . . . .	171
DLFM703E . . . . .	171
DLFM704E . . . . .	172
DLFM706W . . . . .	172
DLFM707I . . . . .	172
DLFM721E . . . . .	172
DLFM741E . . . . .	173
DLFM742E . . . . .	173
DLFM743E . . . . .	173
DLFM744E . . . . .	174
DLFM746E . . . . .	174
DLFM747E . . . . .	174
DLFM748I . . . . .	175
DLFM749E . . . . .	175
DLFM750I . . . . .	175
DLFM751E . . . . .	175
DLFM801I . . . . .	176
DLFM802I . . . . .	176
DLFM803I . . . . .	176
DLFM804I . . . . .	177
DLFM805I . . . . .	177
DLFM806I . . . . .	177
DLFM807I . . . . .	178
DLFM808I . . . . .	178
DLFM809I . . . . .	178
DLFM810I . . . . .	179

DLFM811E . . . . .	179
DLFM812I . . . . .	179
DLFM813I . . . . .	180
DLFM814I . . . . .	180
DLFM900I . . . . .	180
DLFM901E . . . . .	180
DLFM908E . . . . .	181
DLFM909E . . . . .	181

## Appendix A. Data Links File Manager

<b>commands . . . . .</b>	<b>183</b>
dlfm command . . . . .	183
dlfm add_db command . . . . .	183
dlfm add_prefix command . . . . .	184
dlfm bind command . . . . .	185
dlfm create command . . . . .	185
dlfm create_db command . . . . .	186
dlfm deny command . . . . .	186
dlfm drop_db command . . . . .	188
dlfm drop_dlm command . . . . .	189
dlfm grant command . . . . .	189
dlfm grant replication read command . . . . .	191
dlfm grant replication write command . . . . .	193
dlfm help command . . . . .	194
dlfm list registered databases command . . . . .	194
dlfm list registered directories command . . . . .	194
dlfm list registered prefixes command . . . . .	196
dlfm list registered replication access control command . . . . .	196
dlfm list registered users command . . . . .	197
dlfm list upd_in_progress files for db command . . . . .	198
dlfm list upd_in_progress files for prefix command . . . . .	199
dlfm refresh key command . . . . .	201
dlfm restart command . . . . .	202
dlfm retrieve command . . . . .	202
dlfm revoke command . . . . .	203
dlfm revoke replication command . . . . .	205
dlfm see command (AIX and Solaris Operating Environment) . . . . .	206
dlfm set link security command . . . . .	206
dlfm setup command . . . . .	207
dlfm shutdown command . . . . .	208
dlfm start command . . . . .	208
dlfm startdbm command . . . . .	209
dlfm stop command . . . . .	209
dlfm stopdbm command . . . . .	210
dlfm ? command . . . . .	210

<b>Appendix B. Data Links Filesystem Filter commands</b>	<b>213</b>
dlff add command (Windows operating system)	213
Error messages.	213
dlff get dlfmaccount command (Windows operating system).	214
dlff get loglevel command (Windows operating system).	215
dlff list command (Windows operating system)	215
dlff refreshtrace command (Windows operating system).	216
dlff remove command (Windows operating system)	217
Error messages.	217
dlff set dlfm_write_group command (Windows operating system)	218
dlff set dlfmaccount command (Windows operating system).	219
Error messages.	220
dlff set loglevel command (Windows operating system).	221

<b>Appendix C. Data Links File System (DLFS) errors.</b>	<b>223</b>
DLFS errors from commands (AIX)	223
DLFS errors from commands (Solaris Operating Environment)	229
DLFS errors from commands (Windows)	236
DLFS errors from files referenced from columns with WRITE PERMISSION ADMIN.	240

**DB2 Data Links Manager Glossary . . . 243**

<b>Notices</b>	<b>247</b>
Trademarks.	250

**Index . . . 253**

<b>Contacting IBM</b>	<b>257</b>
Product information	257



---

## Tables

1. Database configuration parameters that impact the Data Links Manager archive area . . . . . 23
2. Prohibited linked file operations . . . . . 98
3. DATALINK column attributes and applicable DLM security privileges . . . . . 99
4. Required DATALINK column attributes for Approach 3 . . . . . 119
5. Linked file updating approaches summary. . . . . 135
6. DATALINK column attributes and applicable DLM security privileges . . . . . 187
7. DATALINK column attributes and applicable DLM security privileges . . . . . 190
8. Partial list of AIX commands that you might use to open a file without a token and the associated error message. . . . . 227
9. Partial list of AIX commands that you might use to open a file with an expired token and the associated error message. . . . . 227
10. Partial list of AIX commands that you might use to open a file with an invalid token and the associated error message. . . . . 229
11. Partial list of commands that you might use in the Solaris Operating Environment to open a file without a token and the associated error message. . . . . 233
12. Partial list of commands that you might use in the Solaris Operating Environment to open a file with an expired token and the associated error message. . . . . 234
13. Partial list of commands that you might use in the Solaris Operating Environment to open a file with an invalid token and the associated error message. . . . . 235
14. Partial list of Windows commands that you might use to open a file without a token and the associated error message. . . . . 239
15. Partial list of Windows commands that you might use to open a file with an expired token and the associated error message. . . . . 239
16. Partial list of Windows commands that you might use to open a file with an invalid token and the associated error message. . . . . 240
17. Possible causes and suggested actions for errors that result from opening a file for read operations. . . . . 241
18. Possible causes and suggested actions for errors that result from opening a file for write operations. . . . . 242



---

## About this book

This book provides detailed information about, and shows you how to use, administer, and write applications for the DB2 Data Links Manager on Windows NT, Windows 2000, AIX, and the Sun Solaris Operating Environment.

This manual assumes that you have already installed, configured, and verified DB2 Data Links Manager as described in *Quick Beginnings for DB2 Data Links Manager*, GC09-4829-00.

---

## Who should read this book

This manual is for database administrators, system administrators, IT specialists, application programmers, and anyone else who must work with DB2 Data Links Manager for Windows NT, Windows 2000, AIX, and the Solaris Operating environment.

Before using this book, you should understand:

- DB2 Universal Database
- Structured Query Language (SQL)
- The operating system environment in which DB2 and Data Links Manager is running

For general information about DB2, see the *DB2 Administration Guide: Planning* (SC09-4822-00), *DB2 Administration Guide: Implementation* (SC09-4820-00), and the *DB2 Administration Guide: Performance* (SC09-4821-00) manuals. For information about DB2 commands, see the *DB2 Command Reference* (SC09-4828-00) manual. For information about SQL, see the 2-volume *DB2 SQL Reference* manual (SC09-4844-00 and SC09-4845-00).

**Recommendation:** You should also have access to system and network support experts who have experience in establishing connectivity to various data sources.

**Attention:** This manual does not contain instructions for installing DB2 Data Links Manager. The *Quick Beginnings for DB2 Data Links Manager* manual describes how to install, configure, and validate your DB2 Data Links Manager installation.

The DB2 Data Links Manager Web site at <http://www.ibm.com/software/data/db2/datalinks> contains additional information about Data Links Manager, such as case studies, articles, and presentations.

---

## How this book is structured

This book covers the following topics:

- Chapter 1, “Introduction to DB2 Data Links Manager” on page 3 presents an introduction to, and overview of, DB2 Data Links Manager.
- Chapter 2, “System management options” on page 21 describes the Data Links Manager system options, and how to configure them.
- Chapter 3, “Working with the Data Links File Manager” on page 61 discusses how to use the Data Links File Manager (DLFM) component.
- Chapter 4, “Security” on page 95 lists the different types of security features that Data Links Manager provides and describes how to use them.
- Chapter 5, “Updating linked files” on page 111 explains the various ways in which you can update linked files.
- Chapter 6, “Server recovery” on page 141 explains how to recover a Data Links server in case of failure.
- Chapter 7, “Data Links File Manager error message reference” on page 155 provides a list of Data Links File Manager (DLFM) error codes, messages, and suggested actions.
- Appendix A, “Data Links File Manager commands” on page 183 lists all Data Links File Manager (DLFM) commands.
- Appendix B, “Data Links Filesystem Filter commands” on page 213 lists all Data Links Filesystem Filter (DLFF) commands for the Windows NT and Windows 2000 operating systems.
- Appendix C, “Data Links File System (DLFS) errors” on page 223 lists operations and potential error messages for a Data Links File System (DLFS).
- “DB2 Data Links Manager Glossary” on page 243 presents a list of Data Links Manager terms and acronyms.

---

## Conventions

Unless otherwise stated, the term *Windows* means both the Windows NT and Windows 2000 operating systems.

This book uses the following highlighting conventions:

- **Boldface** indicates commands or graphical user interface (GUI) controls such as names of fields, folders, icons, or menu choices.

- *Italics* indicate variables that you should replace with your own value. They are also used to indicate book titles and to emphasize words.
- Monospace indicates file names, directory paths, commands, and examples of text you enter exactly as shown.



---

# Part 1. Overview

## Chapter 1. Introduction to DB2 Data Links Manager

<b>Manager</b> . . . . .	3
What's new in DB2 Data Links Manager	
Version 8 . . . . .	3
DB2 Data Links Manager . . . . .	4
DB2 Data Links environment. . . . .	5
Data Links server components . . . . .	6
DB2 clients in the Data Links Manager	
environment . . . . .	8
DB2 servers in the Data Links Manager	
environment . . . . .	8
File system paradigm . . . . .	9
DB2 Data Links Manager and your	
applications . . . . .	11
DATALINK data type. . . . .	12
DB2 Data Links Manager typical setups. . . . .	15





---

# Chapter 1. Introduction to DB2 Data Links Manager

---

## What's new in DB2 Data Links Manager Version 8

Version 8 includes the following enhancements:

- The new *DB2 Data Links Manager Administration Guide and Reference* is a reference for system administrators, database administrators, and application programmers. Its contents include:
  - Link and unlink files
  - Update linked files
  - Replicate linked files with DB2 DataPropagator
  - Administer and secure both data and access on a Data Links server
  - Recover a Data Links server
  - Tune your Data Links Manager configuration
  - Troubleshoot Data Links Manager
  - Integrate various file systems to work with Data Links Manager
- DB2 Data Links Manager is now available on the Windows 2000 operating system, in addition to Windows NT, AIX, and the Solaris Operating Environment.
- DB2 Data Links Manager now supports both the Version 8 and Version 7 Solaris Operating Environments.
- You can now update files while they remain under the control of DB2 Data Links Manager. "Linked files" are files under the control of the Data Links Manager. Before DB2 Version 8, you had to unlink a file, make changes to the file, then relink the file.
- Performance improvements have been made for archive, data recovery, and replication operations on linked files. These improvements:
  - Reduce the time for beginning backups involving linked files
  - Reduce the time for running the Reconcile utility
  - Improve the throughput on replication of linked files using DB2 DataPropagator
- Improved security features for restricting the linking of files to authorized users.

---

## DB2 Data Links Manager

Working with DB2, Data Links Manager provides four key elements of external data control:

- Referential integrity
- Access control
- Recovery capabilities
- Transaction consistency

Data Links Manager allows you to take advantage of the data storage features of both a Relational Database Management System (RDBMS) and a file system. You can continue to store unstructured data, such as images, engineering drawings, and x-rays, in a file system and also use an RDBMS to manage data stored within a database.

Data Links Manager offers a unique opportunity for e-commerce and internet applications, which are based on a file system paradigm, to be managed along with RDBMS data from a single point of administration. Other benefits of Data Links Manager include:

- Minimal or no changes to existing applications
- Maximize application performance and reduced network traffic by strategically putting external files close to applications
- Direct access to linked files through native filesystem APIs
- Works with UNIX<sup>®</sup> and Windows<sup>®</sup> file systems and takes advantage of the inherent benefits of the file system

Data Links Manager product extends the relational DBMS capabilities of referential integrity, value-based security, transactional consistency and coordinated backup and recovery to files that exist outside of your database. Data Links Manager manages these files as though they are logically within the database by extending database functionality to external file systems.

Data Links Manager simplifies and reduces system administration costs and complexities by providing a single administration point for file and database data. Data Links Manager guarantees:

- Referential integrity to external files
- RDBMS data value based access control to external files as an option
- Automatic and coordinated backup-and-restore capability within transactional environments

### **Related concepts:**

- “File system paradigm” on page 9

- “DB2 Data Links Manager and your applications” on page 11
- “DATALINK data type” on page 12
- “DB2 Data Links environment” on page 5
- “DB2 Data Links Manager typical setups” on page 15
- “Data Links server components” on page 6
- “DB2 clients in the Data Links Manager environment” on page 8
- “DB2 servers in the Data Links Manager environment” on page 8

**Related tasks:**

- “Installing DB2 Data Links manually using the db2\_install command (AIX)” in the *Quick Beginnings for Data Links Manager*
- “Installing DB2 Data Links Manager using the DB2 Setup wizard (AIX)” in the *Quick Beginnings for Data Links Manager*
- “Installing DB2 Data Links Manager using the DB2 Setup wizard (Solaris Operating Environment)” in the *Quick Beginnings for Data Links Manager*
- “Installing DB2 Data Links Manager (Windows)” in the *Quick Beginnings for Data Links Manager*
- “Installing DB2 Data Links Manager manually using the db2\_install command (Solaris Operating Environment)” in the *Quick Beginnings for Data Links Manager*

---

## DB2 Data Links environment

A DB2<sup>®</sup> Data Links environment is composed of:

- A Data Links server, running DB2 Data Links Manager
- A DB2 Universal Database<sup>™</sup> server
- A DB2 client
- (Optional) File archive server — Tivoli<sup>®</sup> Storage Manager or any product supporting the XBSA standard interface.

You can install DB2 Data Links Manager on:

- AIX<sup>®</sup> systems to manage files stored in a Journaled File System (JFS) or Transarc’s DCE Distributed File Service (DCE-DFS).
- Solaris Operating Environments to manage files stored in a UNIX<sup>®</sup> File System (UFS).
- Windows<sup>®</sup> NT or Windows 2000 systems with an NTFS-formatted drive.
- Tivoli Space Manager managed file system on AIX.

**Related concepts:**

- “DB2 Data Links Manager typical setups” on page 15

- “Data Links server components” on page 6
- “DB2 clients in the Data Links Manager environment” on page 8
- “DB2 servers in the Data Links Manager environment” on page 8

---

## Data Links server components

A Data Links server has the following components:

- Data Links File Manager (DLFM)
- Data Links Filesystem Filter (DLFF) controlling a Data Links File System (DLFS)
- DB2<sup>®</sup> (Logging Manager)

### Data Links File Manager (DLFM)

The DLFM tracks all the files on a Data Links server, or all the files in a DCE cell, that are linked to one or more DB2 databases. The DLFM receives and processes *link-file* and *unlink-file* messages arising from SQL INSERT, UPDATE, and DELETE statements that reference a DATALINK column. For each linked file, the DLFM logically tracks the database instance, the fully qualified table name, and the column name referred to in the SQL statement in which the file is referenced.

The DLFM also tracks previously linked files, if they were linked to a DATALINK column for which the *RECOVERY YES* option was specified. This option allows DB2 to provide point-in-time roll-forward recovery for any file that is specified by a DATALINK column.

### Data Links Filesystem Filter (DLFF)

The DLFF filters operations to ensure that linked files are not deleted or renamed, and that the file’s attributes are not changed. Optionally, it also filters commands to ensure that proper access authority exists for READ PERMISSION DB and WRITE PERMISSION ADMIN files. AIX and Solaris operating environment file systems under the control of a DLFF can be NFS exported. Windows NT<sup>®</sup> and Windows 2000 file systems under DLFF control can be net shared.

A file system under the control of DLFF is called a DLFS file system.

### Data Links File System (DLFS) DMAPP (Data Manager Application)

DLFS file systems include FSM (File System Migrator; the filter file system component of Tivoli<sup>®</sup> Space Manager), JFS, NTFS, or UFS environments; or a Data Links Filesystem Filter (DLFF) DMAPP (Data Manager Application) in DCE-DFS environments.

In DCE-DFS environments, the DMAPP component filters commands to ensure that linked files are not deleted, or renamed, or that the file’s attributes are not changed. The DMAPP component monitors

filesets residing in DMLFS aggregates that are Data Links Manager enabled. Once an aggregate has been Data Links Manager enabled, it can contain filesets that may be brought under Data Links control. The DMAPP can then manage the data within these filesets after the aggregate is exported into the namespace. Data Links Manager enabling an LFS aggregate is part of the Storage Management Toolkit (SMT) provided by Transarc.

### **DB2 (Logging Manager)**

A DB2 database, called DLFM\_DB, acts as a logging manager for the Data Links server. This database contains registration information about databases that can connect to a Data Links server. It also contains information about the mount points of the file systems on AIX® or Solaris operating environment, or the sharename of the drives on Windows NT or Windows 2000, that are managed by a DLFF.

**Note:** It is recommended that you **do not** interact directly with the DLFM\_DB database (for example, connecting to it directly, and either querying or *especially* updating any information) except to take backups and perform restore and rollforward operations as necessary for recovery purposes, or to set certain database configuration parameters to ensure adequate space for logging, etc.

The contents of this database are not documented, and this product is not supported if the user has performed undocumented actions against the DLFM\_DB, since the contents of this database are considered a product internal.

The DLFM\_DB database also contains information about files that have been linked, unlinked, or backed up on a Data Links server or in a DCE cell. The DB2 Data Links Manager installation program creates this database during the installation.

DB2 can provide point-in-time rollforward recovery on the Data Links server for any linked file that is specified by a DATALINK column with *RECOVERY YES*. You can backup these files on disk, using Tivoli Storage Manager, or using an XBSA-conformant backup and archive utility.

The database backup makes sure that all the files on the file server that are linked to this database using a DATALINK column are also backed up. File backup is asynchronous.

To minimize tuning for log file capacity issues in the future, it is recommended that you ensure that there is sufficient disk space

available for the following database configuration variables for DLFM\_DB: LOGFILSIZ, LOGPRIMARY, and LOGSECOND.

**Related concepts:**

- “DB2 Data Links environment” on page 5
- “DB2 Data Links Manager typical setups” on page 15
- “DB2 clients in the Data Links Manager environment” on page 8
- “DB2 servers in the Data Links Manager environment” on page 8

**Related reference:**

- “Configuration Parameters for Database Logging” in the *Data Recovery and High Availability Guide and Reference*

---

## DB2 clients in the Data Links Manager environment

A DB2® client will connect as usual to a remote DB2 server and access the database.

The remote client can NFS mount a file system on AIX® or Solaris operating environment, or share a drive on Windows, under the control of a Data Links Filesystem Filter that is installed on a Data Links server. This way, the client can access the files on the Data Links server directly.

In DCE-DFS environments on AIX systems, the client uses the DFS™ client. A DB2 Data Links DFS Client Enabler, also known as the DLFS cache manager or DLFS-CM, is also required to access linked files in DATALINK columns created with READ PERMISSION DB specified.

**Related concepts:**

- “DB2 Data Links environment” on page 5
- “DB2 Data Links Manager typical setups” on page 15
- “Data Links server components” on page 6
- “DB2 servers in the Data Links Manager environment” on page 8

---

## DB2 servers in the Data Links Manager environment

The Data Links Manager can be registered on a DB2® database running on a DB2 UDB server. For Data Links servers using FSM, NTFS, JFS, and UFS file systems, you can register more than one Data Links Manager with a given DB2 database.

**Note:** Data Links Manager does not support a partitioned DB2 Enterprise Server Edition database.

The database, with tables that include DATALINK columns referencing files stored in a DFS™ filesystem, must register the DCE cell using the ADD DATALINKS MANAGER command. In DCE-DFS environments, the DB2 server can register only one DCE cell. Also, you must install the DFS client on the DB2 server to allow access to configuration information which is stored in DFS.

The AIX® and Solaris operating environments do not require DLFS to be mounted on the DB2 server, and the Windows® environment does not require DLFS to be net shared on the DB2 server. In each of these environments, the DB2 server communicates with the Data Links File Manager using a reserved TCP/IP port on the file server.

**Related concepts:**

- “DB2 Data Links environment” on page 5
- “DB2 Data Links Manager typical setups” on page 15
- “Data Links server components” on page 6
- “DB2 clients in the Data Links Manager environment” on page 8

---

## **File system paradigm**

Current e-business trends, such as e-commerce, supply chain management, and customer relationship management require applications that can integrate structured data from various DBMS and unstructured data that reside on file systems. Examples of such unstructured data are audio, video, and images traditionally stored in an assortment of file formats.

DB2® Data Links Manager allows you to use your existing and emerging file system based applications by leaving them untouched. DB2 Data Links Manager enables these same files to be integrated with database systems to meet the rigorous integrity, security, and transactional demands of e-business.

This advantage allows you the freedom to decide what setup best fits your organization. You do not have to disrupt your business or your customers' access to critical data in order to move it all over to an RDBMS. Moving this data away from the file system paradigm would also require an extensive rewrite of the application you use to access this data.

With Data Links Manager, you can continue using a file paradigm to store, access, and modify files instead of moving these files into a traditional database repository. Continuing to use a file system as a data storage option can provide many benefits.

### **Performance**

The *store and forward* model of data is unacceptable for performance reasons. For example, it may be unacceptable for the database manager to materialize a Binary Large Object (BLOB) into a file, and the converse, each time the data needs to be accessed as a file. Also, data may be captured in high volumes, and you may not want to store this data in the database.

Data Links Manager does not add any overhead to the file READ and WRITE operations. Data Links ensures referential integrity of file references from the database by intercepting file operations such as OPEN, DELETE and RENAME that could compromise the integrity of the file reference. File READ and WRITE operations do not compromise referential integrity and are not part of the Data Links' scope of operations. Also, given that the number of file READ and WRITE operations generally outnumber the occurrences of OPEN, DELETE or RENAME on a file, the performance degradation, if any, is marginal.

### **Network considerations**

You may be accessing data directly from a file server that is physically close to a workstation. For example, you can configure the file server so that the network distance is much shorter to the user, compared to the database where all the BLOBs are stored. The number of bytes that flow for a large object are much larger than the number of bytes for an answer from a typical SQL query. Network distance between resources is therefore a significant consideration.

### **Isochronous delivery**

You may be using an application that uses a stream server because it has real-time requirements for delivery and capture. This is known as isochronous delivery. An example of isochronous delivery might be a video server that delivers high-quality (or "jitter-free") video to a client workstation in real time. In these kinds of applications, it is likely that such data will not be moved into or out of the database as a BLOB, but rather stay on the file server for immediate access.

**Cost** If you are considering using a database as a repository, you should consider first the expense of rewriting applications that currently use standard file I/O semantics. Your applications may use existing tools that work with the file paradigm. Replacing these tools can also be expensive. Using Data Links Manager requires little if any modification of existing file system applications.



The cost of administration of a database is reduced as the large objects are outside the database and the size of the database is more manageable. As the database contains only pointers to files (as DATALINK column values) it takes less time to backup the database. During the backup, DB2 checks with DLFM to ensure that the files linked to this database are backed up. The DLFM starts asynchronous backup of the files after they are linked to the database.

**Related concepts:**

- “DB2 Data Links Manager” on page 4
- “DB2 Data Links Manager and your applications” on page 11

---

## **DB2 Data Links Manager and your applications**

DB2<sup>®</sup> Data Links Manager is a candidate for any application that involves processing information from multiple heterogeneous sources that include databases and file systems, where it is required that this information be consistent between the different sources, secure, accurate, and timely.

In addition to e-commerce, customer relationship management, and supply chain management e-business, Data Links Manager can be used effectively in:

- Medical applications, in which a file server stores X-rays and a database stores their attributes.
- Entertainment industry applications that perform asset management of video clips. A file server stores video clips, but a database stores the clips’ attributes. Access control is required for accessing the video clips based on database privileges for accessing the meta-information.
- World Wide Web applications that manage millions of files, and allow access control that is based on database privileges.
- Financial applications, which require distributed capture of check images and a central location for those images.
- CAD applications, where engineering drawings are kept as files and a database stores their attributes. Queries can be run against the drawing attributes.

Many of these applications need search capabilities to find the data in the files. These search capabilities, however, do not require physically bringing the data into the database system because their raw content is not needed for the query. Typically, you would extract features of an image or a video and store them in the database for performing a search on the extracted features. The ability to store a *reference* to such files and parametric data that describes their contents is the approach these applications use to combine the search capabilities of SQL with direct manipulation of raw data.

The DB2 relational extenders for text, voice, image and so on provide this functionality. The extenders allow you to create special indexes on the data and to search this data. This data can be stored either in DB2 or through DB2 Data Links in a file system.

Data Links Manager can provide all these capabilities even if the data is not in a database, and it can be used in conjunction with the DB2 relational extenders' search features to search on this data as well as perform parametric searches. Additionally, the data and the indexes created by the DB2 extenders can be kept synchronized.

**Central administration point:**

Data Links Manager simplifies many of the tasks associated with managing external files that are logically integrated with database information. For instance, Data Links Manager ensures that a backup of a database is coordinated with a backup of all the files referenced in that database, such as a point-in-time image which can be captured for the combination of database and file data.

Also, restoring the database from a backup results in Data Links Manager automatically restoring the corresponding content of the files, thereby guaranteeing consistency. Contrast this with other systems where the onus of keeping the database and file systems synchronized is left to the database administrator and can be error prone. That setup has a potential of jeopardizing the integrity of the information and the application.

**Related concepts:**

- "DB2 Data Links Manager" on page 4
- "File system paradigm" on page 9

---

## **DATALINK data type**

Data Links technology uses the DATALINK data type, implemented as an SQL data type in DB2<sup>®</sup> Universal Database, which references an object stored external to a database. You can use the DATALINK data type just like any other SQL data type to define columns in tables. The DATALINK type is part of the SQL standard and is applicable across standard-conformant database products.

In File System Migrator (FSM), NT File System (NTFS), Journaled File System (JFS) and UNIX<sup>®</sup> File System (UFS) environments the DATALINK values encode the name of a Data Links Manager server containing the file and the filename in terms of a Uniform Resource Locator (URL).

A URL is a text string of the general format:

```
http://www.ibm.com/datalinks/datalinks.txt
```

DB2 validates the DATALINK value (file reference), just as it does for any SQL data value stored in the database. You register a set of known Data Links Manager servers. The only Data Links Manager server names that you can specify in a DATALINK value are those that have been registered to a DB2 database. Although Data Links uses a URL syntax to reference a file, it does not mean that the Data Links Manager server also functions automatically as a web server.

In DCE-DFS environments, the Data Links Manager is registered for the entire cell. Linked files are referred to in terms of a URL with scheme, "dfs", and the DFS<sup>™</sup> path name of the file.

Even though the DATALINK value represents an object that is stored outside the database system, you can use SQL queries to search parametric data to obtain the file name that corresponds to the query result. You can store attributes in tables, such as indexes on files containing video, image, text, or other media formats, along with the DATALINK value. With a central repository of files on a file server and DATALINK data types in a database, you can obtain an inventory of its contents and devise strategies to get at the information you need.

An application designer can use Data Links to maintain a reference to an existing operating system file in a column of a DB2 table. This reference is stored through a the DATALINK data type using a URL syntax. Other columns of this table would generally maintain meta data about the linked files.

Typically, the application programmer would insert rows in this table with meta data about the file, and its file reference (URL syntax) in the DATALINK column. The application would then typically use an SQL query against this table on the meta data columns to locate the files of interest, retrieve the file reference in the DATALINK column, and then use that URL to directly access the file using the native APIs of the file system or through a browser.

For the application to update or delete a linked file, it must first unlink the file from the DB2 UDB table in which it is referenced, but only if you are not using the update-in-place methodology to perform the file update. You can choose a variety of methodologies for updating linked files: the unlink/update/relink, the update in place, or the replacement method.

For the application to update or delete a linked file, it must first unlink, assuming you have used the method to link a file that requires you to unlink

it, the file from the DB2 UDB table before the operation is allowed to proceed. Multiple DATALINK columns can be defined for a single DB2 UDB table.

Data Links Manager treats information residing in file systems as though it was logically within the database, so you are not required to make any changes to existing applications. DB2 UDB's object-relational capabilities allow an application designer to write UDFs that process files that are either stored within BLOBs, stored as a reference in a DATALINK column, or stored as a reference in a UDT which is a Distinct Type mapping to the DATALINK type.

If files are stored within BLOBs and are currently used in existing applications, then the potential exists for synchronization problems between the BLOB and native file information if the BLOB is simultaneously updated by different users *outside* of the database when the BLOB is materialized as an external file, and updated through some application which only operates on files. When objects are updated outside of the database *not* part of a single database unit of work, there is always the potential for one update overlaying another. With some of the serialization techniques built into Data Links (for example, using the update in place feature), this problem is avoided.

Therefore, while both the BLOB and DATALINK types support file processing through the database, they address different application requirements, and are complementary in nature. DB2 UDB is unique in the industry in offering the customer these two choices to best serve the custom demands of their application.

Data Links Manager has been designed to support a distributed computing environment, with capabilities that include the following:

- A DATALINK column in a DB2 UDB table can reference one or more file systems spread over one or more file system servers associated with different operating systems such as AIX® and Windows® 2000.
- A single Data Links Manager can be associated with DATALINK columns in one or more DB2 UDB databases.
- A DATALINK column can reference files residing in Transarc's distributed file system DCE-DFS.
- Uni-directional and bi-directional replication of linked files is supported in an atomic, automatic, and consistent way in conjunction with DB2 UDB's database replication capabilities through DB2 DataPropagator.

**Related concepts:**

- "DB2 Data Links Manager" on page 4
- "File system paradigm" on page 9
- "DB2 Data Links Manager and your applications" on page 11
- "Updating approaches overview" on page 111

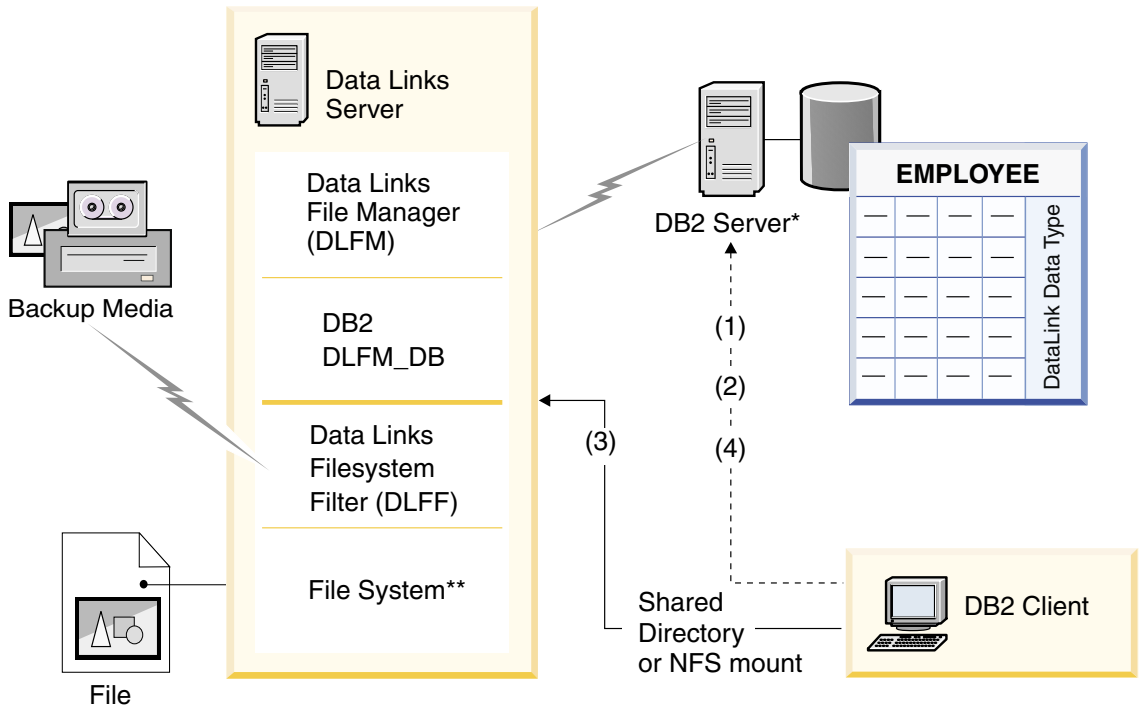
**Related reference:**

- “Updating approaches summary” on page 134

---

## DB2 Data Links Manager typical setups

Figure 1 shows an overview of the interaction between a DB2<sup>®</sup> server, the DB2 Data Links Manager components, the backup media, and a remote client application in FSM on AIX, NTFS, JFS, and UFS environments.



\*Single-partitioned database system

\*\*NTFS on Windows, JFS on AIX or UFS on Solaris

Figure 1. Overview of Data Links Manager Processing (FSM on AIX, NTFS, JFS, or UFS environments)

In this example, a client application connects to a database with a DATALINK data type, selects a DATALINK value from this database, and updates the data file as follows:

1. The client application issues a **CONNECT** statement to connect to a database on a DB2 server.
2. The application then issues a **SELECT** statement that contains a DATALINK column. For example:

```
select dlurlpath(d11) into :var_d11 from EMPLOYEE
```

3. The application then copies the :var\_d11 file to the new\_version file over a shared drive on Windows<sup>®</sup> or an NFS mount on AIX<sup>®</sup> or Solaris operating environment.
4. Then the application edits the new\_version file. To save the changes in the database, the application would issue an **UPDATE** statement. For example:

```
update EMPLOYEE set d11=dlvalue(:new_version)
commit
```

The following diagram shows a Data Links Manager implementation in a DCE-DFS environment on AIX. The numbers in the diagram correspond to the previous steps.

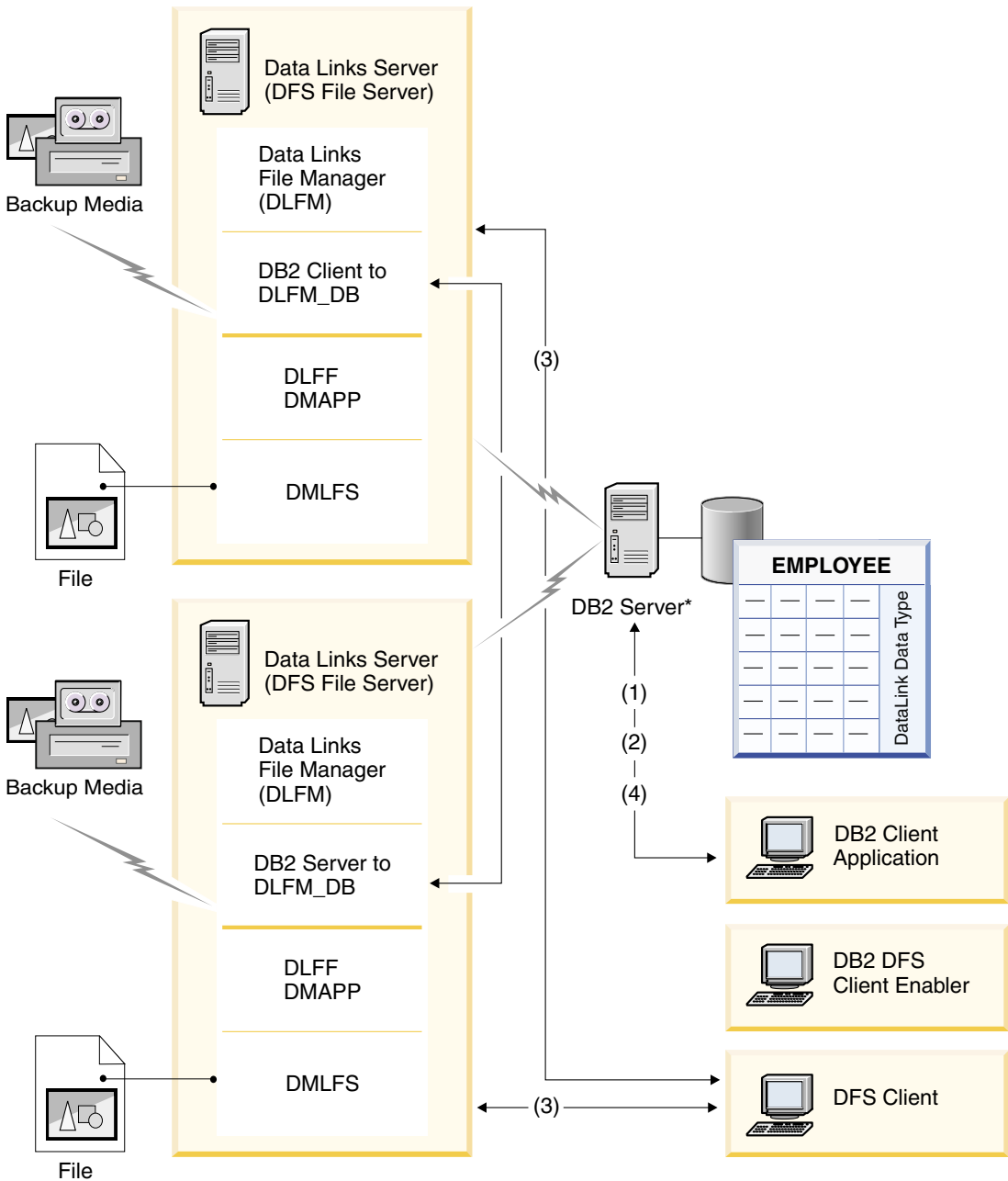


Figure 2. Overview of Data Links Manager Processing (DCE-DFS environments)

**Related concepts:**

- “DB2 Data Links Manager” on page 4

- “DB2 Data Links environment” on page 5



---

## Part 2. Administration Guide and Reference

<b>Chapter 2. System management options</b>	21
Data Links server file backups	21
Data Links server file backups: details	22
Backup storage location	22
Size considerations	23
Using Tivoli Storage Manager as an archive server (AIX)	24
Using Tivoli Storage Manager as an archive server (Solaris Operating Environment)	26
Using Tivoli Storage Manager as an archive server (Windows)	28
Using a local directory for the archive area	30
Using an XBSA-compliant storage manager	31
Authorizing users to perform actions	33
Basic DLFM configuration tasks	37
Basic DLFM configuration tasks: details	38
Creating additional Data Links File Systems on a Data Links server	38
Creating additional Data Links File Systems on a Data Links server: steps	40
Identifying DLFS paths to contain linked files	40
Creating a new UNIX file system (optional)	42
Enabling file system sharing (AIX, Solaris Operating Environment)	43
Enabling file system sharing (Windows)	43
Activating the Data Links Filesystem Filter	44
Modifying a Data Links File System	45
Changing the size of a Data Links File System: tasks	46
Changing the file system size	46
Adding DB2 databases to a DLFM on a Data Links server	48
Adding DLFMs to a specific DB2 database on the DB2 host	50
Dropping Data Links Manager from a specific DB2 database on the DB2 host	53
Enabling the Data Links Manager Replication daemon	54
Configuration tuning	56
Configuration tuning: details	56
Synchronizing system clocks	57

Ensuring sufficient DB2 log space for the DLFM_DB	57
Setting the number of copy processes	58

### Chapter 3. Working with the Data Links File Manager

Basic operations: starting, stopping, and restarting the DLFM	61
Restarting the DLFM after an abnormal termination	62
Monitoring the DLFM background processes (AIX, Solaris Operating Environment)	63
Monitoring the DLFM background processes (Windows)	64
Enabling and registering file systems with DLFM (AIX, Solaris Operating Environment)	65
Registering a drive with DLFF (Windows operating systems)	66
Listing registered file systems (AIX, Solaris Operating Environment)	68
Listing registered drives (Windows operating systems)	68
Loading, querying, and unloading a DLFF (AIX)	69
Loading, querying, and unloading a DLFF (Solaris Operating Environment)	70
Registering, querying and de-registering a DLFF (Windows operating system)	71
Increasing a DLFF-controlled file system size (AIX, Solaris Operating Environment)	72
Registering DB2 Universal Databases with the Data Links server	73
Listing databases registered with the Data Links server	74
Removing information about a DB2 database from Data Links Manager	75
Changing the diagnostic level for error message log files	76
Modifying logging for Data Links Filesystem Filter (DLFF) processing (AIX)	77
Logging messages after the DLFF driver has been loaded (AIX)	79
Minimizing logging for Data Links Filesystem Filter (DLFF) processing (Solaris Operating Environment)	79

Modifying logging for Data Links Filesystem Filter (DLFF) processing (Windows operating systems) . . . . .	80
Migrating a DLFF-enabled file system to a different hard disk—overview . . . . .	80
Migrating a DLFS to a different hard disk: details . . . . .	82
Migrating a DLFS to a different hard disk (AIX) . . . . .	82
Migrating a DLFS to a different hard disk (Solaris Operating Environment) . . . . .	86
Migrating a DLFS to a different drive (Windows) . . . . .	89
Using SCOPY instead of COPY . . . . .	91
SCOPY versus COPY . . . . .	92
Retrieving archive server backup file information . . . . .	93
<b>Chapter 4. Security . . . . .</b>	<b>95</b>
Introduction to Data Links Manager security	95
Built-in security features . . . . .	95
Data access security features . . . . .	96
Data access security features: details . . . . .	97
Basic linked file security controls . . . . .	97
Advanced file management security features . . . . .	98
Advanced file management security controls: detail . . . . .	101
Advanced file management security rules and guidelines . . . . .	101
Read operation security . . . . .	105
Write operation security . . . . .	105
Securing linked files during updates: issues . . . . .	107
Security issues for updating linked files . . . . .	107

---

## Chapter 2. System management options

This chapter describes various Data Links Manager system management options and how to configure and administer them.

---

### Data Links server file backups

The Data Links File Manager (DLFM) automatically takes backups of the following items on a Data Links server:

- Its own database, DLFM\_DB.  
This backup can either be a table space or a full database backup. A table space backup is triggered when certain DLFM\_DB tables are modified (for example, when the **dlfm add\_prefix** or **dlfm add\_db** commands are used on the Data Links server). A full database backup is triggered during significant events like DLFM\_DB creation or DLFM\_DB migration.
- The associated linked files for all table DATALINK columns defined with RECOVERY YES, during file update operations.  
The associated DB2<sup>®</sup> hosts automatically coordinate the linked file backup activity with host backup activity as needed.

When the DLFM initiates backups of Data Links server data, the process of backing up is referred to as *archiving*. The location where the backed up data is stored is called the *archive area*. An archive area can be on a local disk or on an *archive server*.

To retrieve a list of the archived files for a particular Data Links server, and the status of each of those files, you use the **dlfm retrieve** command.

If you accepted the Data Links Manager installation defaults, your archive area is the home directory of the Data Links Manager Administrator account, on a local disk. You can change archive area configuration characteristics, such as directory location and storage medium, as necessary. You can use a local disk, a Tivoli<sup>®</sup> Storage Manager archive server, or an XBSA-compliant archive server (for example, Legato NetWorker).

You can also customize the number of copy processes that get activated on a Data Links server. The number of copy processes affects the speed at which files get copied to the archive area.

**Related concepts:**

- “Backup storage location” on page 22
- “Size considerations” on page 23
- “Using an XBSA-compliant storage manager” on page 31

**Related tasks:**

- “Using Tivoli Storage Manager as an archive server (AIX)” on page 24
- “Using Tivoli Storage Manager as an archive server (Solaris Operating Environment)” on page 26
- “Using Tivoli Storage Manager as an archive server (Windows)” on page 28
- “Using a local directory for the archive area” on page 30
- “Setting the number of copy processes” on page 58
- “Retrieving archive server backup file information” on page 93

**Related reference:**

- “dlfm retrieve command” on page 202

---

## Data Links server file backups: details

The following sections provide more detail about how to plan for and set up a Data Links server file backup scheme.

### Backup storage location

A Data Links server archive area can be located in any of the following places:

- A directory on a local disk (the installation default is the Data Links Manager Administrator account home directory).
- Tivoli® Storage Manager archive server.
- Any archive server that uses the XBSA client APIs.

Benefits and trade-offs are associated with each potential archive area location. For example, the effort of administering a directory on a local disk is minimal, but the amount of storage space might be a limitation. Using a separate archive server requires a separate program to manage the server configuration and its size, and probably additional administrative effort. However, the additional administration effort might be outweighed by having greater storage capacity than a local disk can provide, and by having the ability to automate file storage migration. If you are already using a storage manager application such as IBM’s Tivoli Storage Manager, or any commercial XBSA-compliant program, such as Legato NetWorker, any additional administration efforts might be minimal.

You must choose an archive area location that is based on your organization’s needs.

## Size considerations

When you configure an archive area, you must ensure that the size of the area is large enough, or can be made large enough, to accommodate your organization's needs. If you accepted the Data Links Manager installation defaults, the archive area is the Data Links Manager Administrator account home directory, on a local disk.

The following potential issues can affect the size of an archive area:

- Each table DATALINK column defined with RECOVERY YES triggers a back up of the associated linked file. Each time the linked file gets updated, another backup is made.

**Requirement:** The storage location must have at least the same amount of space as all of the files to be linked in a particular database to a DATALINK column with the RECOVERY YES attribute.

- Multiple DB2® databases can store linked files on the same Data Links server.

You must ensure that the storage location has enough space to accommodate all backups for all databases that are storing files on a particular Data Links server.

- The values of certain DB2 host database configuration parameters can impact the storage space needed for an archive area. Table 1 lists those database configuration parameters.

**Important:** If one Data Links File Manager (DLFM) manages files for *multiple* DB2 databases, note that the value of each of these parameters can be different for each database.

*Table 1. Database configuration parameters that impact the Data Links Manager archive area*

Parameter	Description	Default value [Range]	Comments
num_db_backups	The number of the most recent database backups to retain.	12 [0-32,768]	When the specified number is reached, the oldest backup is deleted from the archive area.
dl_num_copies	The number of additional copies to be made in the archive area when a file is linked.	0 [0-15]	<b>Recommendation:</b> Use the default value of zero (0) to avoid the extra space required for multiple copies of the same file.

Table 1. Database configuration parameters that impact the Data Links Manager archive area (continued)

Parameter	Description	Default value [Range]	Comments
dl_time_drop	The interval of time, in days, files will be retained in the archive area after a <b>DROP DATABASE</b> command is issued.	1 [0-365]	A value of zero (0) means that the files are deleted immediately from the archive server when a <b>DROP DATABASE</b> command is issued.

**Related reference:**

- “Data Links Number of Copies configuration parameter - dl\_num\_copies” in the *Administration Guide: Performance*
- “Data Links Time After Drop configuration parameter - dl\_time\_drop” in the *Administration Guide: Performance*
- “UPDATE DATABASE MANAGER CONFIGURATION Command” in the *Command Reference*
- “Number of Database Backups configuration parameter - num\_db\_backups” in the *Administration Guide: Performance*
- “Configuration parameters summary” in the *Administration Guide: Performance*

**Using Tivoli Storage Manager as an archive server (AIX)**

You can use Tivoli Storage Manager (TSM) to back up files that reside on a Data Links server.

**Prerequisites:**

You must have both Data Links Manager Administrator *and* root authorities on the Data Links server machine that you will be using.

**Procedure:**

To use Tivoli Storage Manager as an archive server:

1. Install the Tivoli Storage Manager client on the Data Links server machine. For more information, see your Tivoli Storage Manager product documentation.
2. Register the Data Links server client application with the Tivoli Storage Manager server. For more information, see your Tivoli Storage Manager product documentation.

3. Add the following environment variables to the Data Links Manager Administrator's db2profile or db2cshrc script files:
 

```
(for Bash, Bourne, or Korn shell)
export DSMI_DIR=/usr/tivoli/tsm/client/api/bin
export DSMI_CONFIG=$HOME/tsm/dsm.opt
export DSMI_LOG=$HOME/d1dump
export PATH=$PATH:$DSMI_DIR

(for C shell)
setenv DSMI_DIR /usr/tivoli/tsm/client/api/bin
setenv DSMI_CONFIG ${HOME}/tsm/dsm.opt
setenv DSMI_LOG ${HOME}/d1dump
setenv PATH=${PATH}:$DSMI_DIR
```
4. Ensure that the dsm.sys TSM system options file is located in the \$DSMI\_DIR directory.
5. Ensure that the dsm.opt TSM user options file is located in the *INSTHOME*/tsm directory, where *INSTHOME* is the home directory of the Data Links Manager Administrator.
6. Set the *PASSWORDACCESS* option to generate in the /usr/tivoli/tsm/client/api/bin/dsm.sys Tivoli Storage Manager system options file.
7. Register the TSM password with the generate option *before* starting the Data Links File Manager (DLFM) using Tivoli Storage Manager as a backup target. This way, you will not need to provide a password when the DLFM initiates a connection to the TSM server. For more information, see your TSM product documentation.
8. Log on to the Data Links server as the Data Links Manager Administrator.
9. Use the **db2set** command to set the DLFM\_BACKUP\_TARGET registry variable to TSM. The value of DLFM\_BACKUP\_DIR\_NAME registry variable is ignored in this case. Changing the DLFM\_BACKUP\_TARGET registry variable to TSM will activate the Tivoli Storage Manager backup option.

**Notes:**

- a. Do not change the setting of the DLFM\_BACKUP\_TARGET registry variable on an active Data Links server, because:
  - The previously archived files are not moved to the newly specified archive location. For example, if you start the DLFM with the DLFM\_BACKUP\_TARGET registry value set to TSM, and change the registry value to a disk location, all newly archived files are stored in the new location on the disk. The files that were previously archived to TSM are not moved to the new disk location.

- You cannot do a full recovery or reconcile operation using files that were previously archived. The RESTORE utility will be unable to perform a full recovery, and the RECONCILE utility will show multiple exceptions.
  - b. You can override the default TSM management class with the DLFM\_TSM\_MGMTCLASS registry variable. If this registry variable is left unset, then the default TSM management class is used.
10. Restart the DLFM by invoking the **dlfm restart** command.

**Related tasks:**

- “Using Tivoli Storage Manager as an archive server (Solaris Operating Environment)” on page 26
- “Using Tivoli Storage Manager as an archive server (Windows)” on page 28
- “Using a local directory for the archive area” on page 30

**Related reference:**

- “db2set - DB2 Profile Registry Command” in the *Command Reference*
- “Tivoli Storage Manager” in the *Data Recovery and High Availability Guide and Reference*
- “Data-links variables” in the *Administration Guide: Performance*

## **Using Tivoli Storage Manager as an archive server (Solaris Operating Environment)**

You can use Tivoli Storage Manager (TSM) to back up files that reside on a Data Links server.

**Prerequisites:**

You must have both Data Links Manager Administrator *and* root authorities on the Data Links server machine that you will be using.

**Procedure:**

To use Tivoli Storage Manager as an archive server:

1. Install the Tivoli Storage Manager client on the Data Links server machine. For more information, see your Tivoli Storage Manager product documentation.
2. Register the Data Links server client application with the Tivoli Storage Manager server. For more information, see your Tivoli Storage Manager product documentation.
3. Add the following environment variables to the Data Links Manager Administrator’s db2profile or db2cshrc script files:



```
(for Bash, Bourne, or Korn shell)
export DSMI_DIR=/opt/tivoli/tsm/client/api/bin
export DSMI_CONFIG=$HOME/tsm/dsm.opt
export DSMI_LOG=$HOME/dldump
export PATH=$PATH:/opt/tivoli/tsm/client/api/bin
```

```
(for C shell)
setenv DSMI_DIR /opt/tivoli/tsm/client/api/bin
setenv DSMI_CONFIG ${HOME}/tsm/dsm.opt
setenv DSMI_LOG ${HOME}/dldump
setenv PATH=${PATH}:/opt/tivoli/tsm/client/api/bin
```

4. Ensure that the dsm.sys TSM system options file is located in the /opt/tivoli/tsm/client/api/bin directory.
5. Ensure that the dsm.opt TSM user options file is located in the *INSTHOME*/tsm directory, where *INSTHOME* is the home directory of the Data Links Manager Administrator.
6. Set the *PASSWORDACCESS* option to generate in the /opt/tivoli/tsm/client/api/bin/dsm.sys Tivoli Storage Manager system options file.
7. Register the TSM password with the generate option *before* starting the Data Links File Manager (DLFM) using Tivoli Storage Manager as a backup target. This way, you will not need to provide a password when the DLFM initiates a connection to the TSM server. For more information, see your TSM product documentation.
8. Log on to the Data Links server as the Data Links Manager Administrator.
9. Use the **db2set** command to set the DLFM\_BACKUP\_TARGET registry variable to TSM. The value of DLFM\_BACKUP\_DIR\_NAME registry variable is ignored in this case. Changing the DLFM\_BACKUP\_TARGET registry variable to TSM will activate the Tivoli Storage Manager backup option.

**Notes:**

- a. Do not change the setting of the DLFM\_BACKUP\_TARGET registry variable on an active Data Links server, because:
  - The previously archived files are not moved to the newly specified archive location. For example, if you start the DLFM with the DLFM\_BACKUP\_TARGET registry value set to TSM, and change the registry value to a disk location, all newly archived files are stored in the new location on the disk. The files that were previously archived to TSM are not moved to the new disk location.
  - You cannot do a full recovery or reconcile operation using files that were previously archived. The RESTORE utility will be unable to perform a full recovery, and the RECONCILE utility will show multiple exceptions.

- b. You can override the default TSM management class with the `DLFM_TSM_MGMTCLASS` registry variable. If this registry variable is left unset, then the default TSM management class is used.
10. Restart the DLFM by invoking the **`dlfm restart`** command.

**Related tasks:**

- “Using Tivoli Storage Manager as an archive server (AIX)” on page 24
- “Using Tivoli Storage Manager as an archive server (Windows)” on page 28
- “Using a local directory for the archive area” on page 30

**Related reference:**

- “db2set - DB2 Profile Registry Command” in the *Command Reference*
- “Tivoli Storage Manager” in the *Data Recovery and High Availability Guide and Reference*
- “Data-links variables” in the *Administration Guide: Performance*

## Using Tivoli Storage Manager as an archive server (Windows)

You can use Tivoli Storage Manager (TSM) to back up files that reside on a Data Links server.

**Prerequisites:**

You must have both Data Links Manager Administrator *and* Windows system administrator authorities on the Data Links server machine that you will be using.

**Procedure:**

To use Tivoli Storage Manager as an archive server:

1. Install the Tivoli Storage Manager client on the Data Links server machine. For more information, see your Tivoli Storage Manager product documentation.
2. Register the Data Links server client application with the Tivoli Storage Manager server. For more information, see your Tivoli Storage Manager product documentation.
3. Log on to the Data Links server as the Data Links Manager Administrator.
4. Click **Start** and select **Settings** —> **Control Panel** —> **System**. The System Properties window opens. Navigate to **Environment variables**, and enter the following system environment variables and corresponding values:

Variable	Value
DSMI_DIR	c:\tsm\baclient
DSMI_CONFIG	c:\tsm\baclient\dsm.opt
DSMI_LOG	c:\tsm\dlldump

5. Ensure that the dsm.sys TSM system options file is located in the c:\tsm\baclient directory.
6. Ensure that the dsm.opt TSM user options file is located in the c:\tsm\baclient directory.
7. Set the *PASSWORDACCESS* option to generate in the c:\tsm\baclient\dsm.sys Tivoli Storage Manager system options file.
8. Register the TSM password with the generate option *before* starting the Data Links File Manager (DLFM) using Tivoli Storage Manager as a backup target. This way, you will not need to provide a password when the Data Links File Manager initiates a connection to the TSM server. For more information, see your TSM product documentation.
9. Set the DLFM\_BACKUP\_TARGET environment variable to TSM using the following command:

```
db2set -g DLFM_BACKUP_TARGET=TSM
```

The value of the DLFM\_BACKUP\_DIR\_NAME environment variable is ignored in this case. Changing the DLFM\_BACKUP\_TARGET registry variable to TSM will activate the Tivoli Storage Manager backup option.

#### Notes:

- a. Do not change the setting of the DLFM\_BACKUP\_TARGET registry variable on an active Data Links server, because:
    - The previously archived files are not moved to the newly specified archive location. For example, if you start the DLFM with the DLFM\_BACKUP\_TARGET registry value set to TSM, and change the registry value to a disk location, all newly archived files are stored in the new location on the disk. The files that were previously archived to TSM are not moved to the new disk location.
    - You cannot do a full recovery or reconcile operation using files that were previously archived. The RESTORE utility will be unable to perform a full recovery, and the RECONCILE utility will show multiple exceptions.
  - b. You can override the default TSM management class with the DLFM\_TSM\_MGMTCLASS registry variable. If this registry variable is left unset, then the default TSM management class is used.
10. Restart the DLFM by invoking the **dlfm restart** command.

#### Related tasks:

- “Using Tivoli Storage Manager as an archive server (AIX)” on page 24
- “Using Tivoli Storage Manager as an archive server (Solaris Operating Environment)” on page 26
- “Using a local directory for the archive area” on page 30

**Related reference:**

- “db2set - DB2 Profile Registry Command” in the *Command Reference*
- “Tivoli Storage Manager” in the *Data Recovery and High Availability Guide and Reference*
- “Data-links variables” in the *Administration Guide: Performance*

## Using a local directory for the archive area

Use this procedure to change the backup directory location of an archive area on a local disk.

**Prerequisites:**

- The Data Links server must already be configured to use a local disk as an archive area.
- The new directory that you want to use must exist.
- The new directory that you want to use must have the required space. See the end of this topic for a link to more information about archive area space requirements.
- You must have DB2 Data Links Manager Administrator authority.
- UNIX systems only: To use a directory that is NFS mounted, you must ensure that any user name with root authority on that file server has read and write access to the mounted directory. The DB2 Data Links Manager Administrator’s user name must also exist on the machine where this directory was NFS mounted and exported.

**Restrictions:**

You cannot use a directory that is located on a file system using a Data Links Filesystem Filter (DLFF).

**Procedure:**

To change the directory location of a Data Links server archive area:

1. Log on to the specific Data Links server machine where you want change the directory location.
2. Stop the Data Links File Manager (DLFM) with the **dlfm stop** command.

3. Copy the entire contents of the current backup directory to the new backup directory, preserving all original file names, directory names, timestamps, and the original directory structure.  
**Attention:** You can remove or clean out the previous backup directory only after successfully completing this entire procedure.
4. Using the **db2set** command, set the registry variable `DLFM_BACKUP_DIR_NAME` as an absolute path to the directory that you want to use. For example:  

```
db2set DLFM_BACKUP_DIR_NAME=/home/dl_fm/dl_fm_backup_dir
```
5. Restart the DLFM with the **dlfm start** command.

**Related concepts:**

- “Data Links server file backups” on page 21

**Related reference:**

- “db2set - DB2 Profile Registry Command” in the *Command Reference*
- “Data-links variables” in the *Administration Guide: Performance*

## Using an XBSA-compliant storage manager

You can use an XBSA-compliant storage application to back up files that reside on a Data Links server. An XBSA-compliant storage application is any program that uses industry-standard XBSA APIs for backup and restore operations.

**Prerequisites:**

You must have both Data Links Manager Administrator *and* superuser authorities on the Data Links server machine that you will be using.

UNIX<sup>®</sup> systems: You must have root authority.

Windows: You must use a system administrator account (for example, the `dldadmin` account).

**Procedure:**

To use an XBSA-compliant storage application as an archive server:

1. Install an XBSA client on the Data Links server. For more information, refer to your XBSA vendor product documentation.
2. Register the Data Links server client application with the XBSA server. For more information, refer to your XBSA vendor server documentation.
3. Add the required environment variables to the DB2<sup>®</sup> Data Links Manager Administrator account configuration.

- On UNIX systems, you add the required environment variables to the Data Links Manager Administrator account's db2profile or db2cshrc script files. For example, Legato NetWorker requires the following variables:

```
(for bash, Bourne, or Korn shell)
export NSR_SERVER=fully_qualified_server_hostname
```

```
(for C shell)
setenv NSR_SERVER=fully_qualified_server_hostname
```

*fully\_qualified\_server\_hostname* is the fully qualified hostname of the Legato NetWorker server machine.

- On Windows, you add the required environment variables to the Data Links Manager Administrator account configuration by using the Windows® System Properties **Environment Variables**.
  - a. Click Start and select **Settings** → **Control Panel** → **System**. The System Properties window opens.
  - b. Select the **Environment** tab, then enter the required *system* environment variables.

For example, Legato NetWorker requires a system environment variable called NSR\_SERVER to be set with the fully qualified hostname of the Legato NetWorker server machine.

4. Log on to the Data Links server as the Data Links Manager Administrator.
5. Use the **db2set** command to set the DLFM\_BACKUP\_TARGET registry variable to XBSA.
6. Use the **db2set** command to set the DLFM\_BACKUP\_TARGET\_LIBRARY registry variable to the fully-qualified shared library supplied by the vendor. On UNIX systems, the library name must also specify which shared object in the library is to be used. The name of the shared object is available from the vendor supplying the XBSA-compliant shared library.

In the following examples, which illustrate setting the variable for use with the Legato NetWorker application, the name of the shared object is bsashr10.o:

```
db2set DLFM_BACKUP_TARGET_LIBRARY=/usr/lpp/Legato/libxdb2.a(bsashr10.o) //AIX
db2set DLFM_BACKUP_TARGET_LIBRARY=/opt/IBM/db2/v8.1/Legato/libxdb2.so(bsashr10.o) //Solaris
db2set DLFM_BACKUP_TARGET_LIBRARY=c:\Program Files\nsr\bin\libbsa.dll //Windows
```

7. Restart the Data Links File Manager (DLFM) with the **dlfm restart** command.

#### Related tasks:

- “Using Tivoli Storage Manager as an archive server (AIX)” on page 24
- “Using Tivoli Storage Manager as an archive server (Solaris Operating Environment)” on page 26
- “Using Tivoli Storage Manager as an archive server (Windows)” on page 28

- “Using a local directory for the archive area” on page 30

---

## Authorizing users to perform actions

This topic lists the common actions performed by Data Links Manager users and the corresponding authorizations that you need to assign to your users who perform these actions. This topic also lists other items that you need to consider when assigning authorizations and when customizing authorization levels.

The common authorizations are:

- “Authorizations for linking files”
- “Authorizations for updating linked files” on page 34
- “Authorizations for replicating linked files” on page 35

### Authorizations for linking files

Data Links Manager provides customizable security features so that you can authorize users to link to the files that reside on a particular Data Links server. Users are prohibited from performing file link operations unless you explicitly authorize such operations to your users.

To authorize file linking operations to your users, use the following Data Links File Manager (DLFM) commands.

#### **dlfm set link security**

This command activates (and deactivates) the file link security feature for a specific Data Links server. File link security must be activated (on) before any authorizations for linking privileges to files for specific users take effect.

By default, file link security is *on* when you install Data Links Manager.

#### **dlfm grant**

This command grants link privileges to specific DB2<sup>®</sup> database users for access to the files stored on a Data Links server. You can grant these privileges to a specific user, to a user group, or to all users of a specific DB2 database.

Because the file link security control feature is turned on (activated) by default at installation time, *no one* can link to files on a Data Links server until you explicitly grant them the link privilege. Therefore, after you install Data Link Manager, you *must* use the **dlfm grant** command to explicitly grant file link privileges to users.

You can prohibit linking privileges or remove the file linking authorizations by using the following commands.

**dlfm deny**

This command prohibits users from linking to files that are stored on a Data Links server. You can deny privileges to individual users or groups of users on specific Data Link server directories.

**dlfm revoke**

This command removes the effects of the **dlfm grant** or **dlfm deny** commands.

To view a list of users who are authorized to link to files, use the following DLFM commands.

**dlfm list registered users**

This command lists the users with link privileges for a specific Data Links server directory.

**dlfm list registered directories**

This command lists all Data Links server directories and the access privileges for a specific user, user group, or all users on a particular DB2 database, database instance, and database server node.

**Authorizations for updating linked files**

When table DATALINK columns are defined with the WRITE PERMISSION ADMIN attribute, you can authorize users to make changes to the contents of the linked files stored on a Data Links server.

To enable users to update linked files, use the **dlfm grant** command. This command allows you to grant the write (update) privilege so that specified users can generate and obtain write tokens. Without valid write tokens, users *cannot* update linked files.

You need to verify the following settings and authorizations if you want to use a linked file updating approach that performs updates-in-place.

- Verify that the users are authorized to access the DB2 database in which the linked file reference is stored, and to access and update the linked file on the Data Links server.
- Verify that the DATALINK columns of the DB2 table are defined with the correct READ PERMISSION and RECOVERY attribute settings.

You can prohibit updating privileges or reverse these authorizations by using the following commands.

**dlfm deny**

This command prohibits users from updating linked files that are stored on a Data Links server.



### **dlfm revoke**

This command enables you to remove the effects of the **dlfm grant** or **dlfm deny** commands.

To view a list of users who are authorized to update linked files, use the following DLFM commands.

### **dlfm list registered users**

This command enables you to list the users with update privileges for a specific Data Links server directory.

### **dlfm list registered directories**

This command enables you to list all Data Links server directories and view the access privileges for a specific user, user group, or all users on a particular DB2 database, database instance, and database server node.

## **Authorizations for replicating linked files**

Data Links Manager provides authorization commands to support DB2 replication users who need access to linked files to replicate linked data.

**Important:** During the Data Links Manager installation process, you are prompted to enable the Data Links Manager replication file-copy daemon (DLFM\_ASNCOPYD). If you plan to use Data Links Manager for replication, enable the DLFM\_ASNCOPYD daemon during the installation process. You can also enable this daemon after installation, if necessary.

To allow DB2 replication users to access linked files for data replication, use the following DLFM commands.

### **dlfm grant replication read**

This command grants specific Data Links server system users the authority to read all of the linked files in a specific directory on the Data Links server. You can grant the authority to a specific user, to a user group, or to all registered Data Links server system users.

### **dlfm grant replication write request**

This command grants specific Data Links server system users the authority to issue a file write request to the DLFM\_ASNCOPYD daemon. DLFM\_ASNCOPYD communicates with the ASNDLCOPY exit routine, which is part of DB2 replication, to replicate linked files.

**Important:** This command does not grant the authority to create and modify files. This command also does not grant the authority to create directories. All of those operations are controlled by a user's file system access permissions.

To remove the read privilege or the write request privilege from user accounts that no longer need to access linked files for replication, use the **dlfm revoke**

**replication** command. This command enables you to remove read access to linked files or to remove the authority to issue a file write request to the DLFM\_ASNCOPYD daemon for authorized users.

To view the current read and write requests that Data Links server users can perform, use the **dlfm list registered replication access control** command. This command displays a list of read and write requests privileges defined to the DLFM\_ASNCOPYD daemon for a specific system user, for a system group, or for all Data Links server users.

**Related concepts:**

- “Basic linked file security controls” on page 97
- “Advanced file management security features” on page 98
- “Advanced file management security rules and guidelines” on page 101
- “Approach 1: Unlink/update/relink files” on page 113
- “Approach 2: Update linked files/native file system defines access” on page 115
- “Approach 3: Update linked files/customize file access privileges” on page 116
- “Approach 4: Replace a linked file's contents” on page 131

**Related tasks:**

- “Replicating special data types” in the *Replication Guide and Reference*
- “Enabling the Data Links Manager Replication daemon” on page 54

**Related reference:**

- “dlfm deny command” on page 186
- “dlfm grant replication read command” on page 191
- “dlfm grant replication write command” on page 193
- “dlfm grant command” on page 189
- “dlfm list registered replication access control command” on page 196
- “dlfm list registered users command” on page 197
- “dlfm list registered directories command” on page 194
- “dlfm revoke command” on page 203
- “dlfm revoke replication command” on page 205
- “dlfm set link security command” on page 206

---

## Basic DLFM configuration tasks

The Data Links File Manager (DLFM) is one of the main components of a Data Links server. Most of the tasks involved in administering a Data Links Manager system require interacting with the DLFM.

The DLFM tracks all of the files on a Data Links server that are linked to one or more DB2 databases. The DLFM receives and processes link-file and unlink-file messages that result from SQL INSERT, UPDATE, and DELETE statements that reference a DATALINK column. For each linked file, the DLFM logically tracks the database instance, the fully qualified table name, and the column name referred to in the SQL statement in which the file is referenced.

The DLFM also tracks previously linked files if they were linked to a table DATALINK column for which the RECOVERY YES option was specified. This option enables DB2 to provide point-in-time roll-forward recovery for any file that is specified in a DATALINK column.

When you first install Data Links Manager, communications are enabled between a single DLFM and a single DB2 database. However, Data Links Manager enables you to establish relationships between *multiple* DB2 databases and *multiple* DLFMs. A single DB2 database can communicate with up to 16 DLFMs.

**Important:** Administrative complexity increases in proportion to the number of relationships between multiple DB2 databases and multiple DLFMs. Try to keep your system configuration as simple as possible within the context of your requirements.

You can *expand* your Data Links Manager system at any time by doing either or both of the following tasks:

- Adding new Data Links server machines (this task requires installing Data Links Manager on the new machine).
- Creating additional Data Links File Systems (DLFSs) on existing Data Links server machines.

You can *modify* your existing system configuration at any time by doing any or all of the following tasks:

- Changing the size of a DLFS on existing Data Links servers
- Adding DLFMs to DB2 databases
- Adding DB2 databases to DLFMs
- Dropping Data Links Manager from DB2 databases
- Adding Tivoli Space Manager to a DLFS (AIX systems only)

- Enabling DB2 replication on a Data Links server

**Related concepts:**

- “Before you install DB2 Data Links Manager (AIX)” in the *Quick Beginnings for Data Links Manager*
- “Before you install DB2 Data Links Manager (Solaris Operating Environment)” in the *Quick Beginnings for Data Links Manager*
- “Before you install DB2 Data Links Manager (Windows)” in the *Quick Beginnings for Data Links Manager*
- “Dropping Data Links Manager from a specific DB2 database on the DB2 host” on page 53

**Related tasks:**

- “Creating additional Data Links File Systems on a Data Links server” on page 38
- “Modifying a Data Links File System” on page 45
- “Adding DLFMs to a specific DB2 database on the DB2 host” on page 50
- “Adding DB2 databases to a DLFM on a Data Links server” on page 48
- “Enabling the Data Links Manager Replication daemon” on page 54

---

## **Basic DLFM configuration tasks: details**

The following sections provide specific details and procedures about how to:

- Create additional Data Links File Systems (DLFSs) on existing Data Links servers
- Change the size of a DLFS on existing Data Links servers
- Add DLFMs to DB2 databases
- Add DB2 databases to DLFMs
- Drop Data Links Manager from DB2 databases
- Enable DB2 replication on a Data Links server

Additional DLFM-related tasks are described in Chapter 3, “Working with the Data Links File Manager” on page 61.

### **Creating additional Data Links File Systems on a Data Links server**

This procedure provides an overview of the process for creating new DLFSs on a Data Links server. See the end of this procedure for detailed links to most of the procedure steps.

**Prerequisites:**

You must have both Data Links Manager Administrator *and* superuser authorities on the Data Links server machine that you will be using.

UNIX systems: You must have root authority.

Windows: You must use a system administrator account (for example, the dlmadmin account).

### Procedure:

To create a new DLFS on an existing Data Links server machine:

1. Identify the DLFS paths that will contain linked files.
2. Create a new file system (optional).

You can use an existing file system as long as there is no conflict with having it be controlled by the Data Links Filesystem Filter (DLFF).

On Windows, you create a new NTFS partition using the Windows NT Disk Administrator tool or Windows 2000 Disk Management

3. Enable the chosen file system to be shared.
4. Turn the chosen file system into a DLFS by activating the Data Links Filesystem Filter (DLFF).

As part of this task, you define DLFS paths to the DLFM with the **dlfm add\_prefix** command.

5. Enable client connectivity to the DLFS. You can enable client connectivity as part of this procedure, or at a later time.

- On UNIX systems, clients initiate a local file system mount with the **mount** command. Mount command parameters can either be specified directly on the mount command, or in the client's /etc/filesystems file (AIX) or /etc/vfstab file (Solaris Operating Environment). Then, the client can perform the mount by just specifying the directory to be mounted, and the mount parameters will automatically be taken from the appropriate file.

You can also specify things like the file systems to be automatically mounted at boot-up time, and NFS caching characteristics for the mount in the appropriate client files. Refer to your AIX or Solaris Operating Environment documentation for more information about automating mounts.

**AIX only:** You can use the SMIT admin tool to compose the **mount** command more easily.

- On Windows, clients can initiate access to a shared drive by using the **net use** command, or by using the **Map Network Drive** function in Windows Explorer.
6. Authorize users to perform actions on the DLFS.

You can enable user authorizations as part of this procedure, or at a later time.

**Related concepts:**

- “Authorizing users to perform actions” on page 33

**Related tasks:**

- “Identifying DLFS paths to contain linked files” on page 40
- “Creating a new UNIX file system (optional)” on page 42
- “Activating the Data Links Filesystem Filter” on page 44
- “Enabling file system sharing (AIX, Solaris Operating Environment)” on page 43
- “Enabling file system sharing (Windows)” on page 43

**Related reference:**

- “Basic DLFM configuration tasks” on page 37

## **Creating additional Data Links File Systems on a Data Links server: steps**

The following sections provide detailed instructions for creating new DLFSs on a Data Links server.

### **Identifying DLFS paths to contain linked files**

*Identifying Data Links File System (DLFS) paths to contain linked files* is part of the larger task of creating a new DLFS on an existing Data Links server.

Before you begin the task of adding a DLFS to an existing Data Links server, you must determine the directory, or the directory path, where you want the linked files to be stored. Consider the kinds of data you need to store, and how you might need to organize that data over time as the amount increases.

After you have determined the storage directory of the linked files, you will register a *prefix* corresponding to that location with the Data Links File Manager (DLFM) on a Data Links server. In Data Links Manager, a prefix is the *absolute path* of a DLFS mount point (on UNIX systems) or a drive sharename (on Windows) under which linked files are stored. An absolute path is the full path name of an object. Absolute path names begin at the highest level, or “root” directory, which is identified by the forward slash (/) or backward slash (\) character.

You can specify only one prefix per DLFS, and, once you have specified it, you cannot change it. However, you can create subdirectories of a prefix at any time, as necessary. You do not have to register or define an existing prefix’s subdirectories to DLFF or DLFM.

An exercise to help you in determining prefix values is to think about what the uniform resource locator (URL) values might look like in a DATALINK column for your linked files.

**Important:** The following example and discussion use UNIX file path naming conventions. However, the directory and prefix naming considerations that are described apply to *all* file and operating systems.

**Example:**

Suppose you want to add employee resumes to an employee database. You want to store the resume files in their native format. You already have a Data Links server machine named mercury. A URL to one of your resume files might look like this: `http://mercury/resumes/jdoe1.doc`, where `resumes` is a server directory, and `jdoe1.doc` is the resume file that will be linked.

The example illustrates a valid prefix location, but not a practical one, because it does not allow for the expansion of data over time. If you wanted to add other types of data at a later time, such as employee photographs, you might have a problem. *You can only specify one prefix per DLFS.* You could work around this limitation, but the potential solutions might be confusing to end-users and other system users, such as machine administrators. For example, you could store all employee data, regardless of its type, in the `resumes` directory. However, no one might think to look for employee photographs in a directory called `resumes`. Another potential solution: create a subdirectory of the `resumes` directory called `photos`. However, not only would your directory structure look strange (`/resumes/photos/`), but it might not occur to anyone to look for a subdirectory at all.

To serve both the present need (to store employee resume files) and any future needs (for example, to store employee photographs), it might be better to create a generic directory on mercury named `empdata`. The DLFS prefix would be `/empdata`. Your directory structure could easily be expanded to include paths `/empdata/resumes/` and `/empdata/photos/`, if and when they become needed. You would not *ever* have to do any extra administrative work in Data Links Manager to accommodate those paths — you would only need to create the subdirectories in the file system.

Creating and using generic prefixes allows you the flexibility to expand your directory structure in a way that will make sense over time, with a minimum amount of administrative work in Data Links Manager.

**Related tasks:**

- “Creating additional Data Links File Systems on a Data Links server” on page 38

- “Activating the Data Links Filesystem Filter” on page 44
- “Enabling file system sharing (AIX, Solaris Operating Environment)” on page 43
- “Enabling file system sharing (Windows)” on page 43

**Related reference:**

- “Basic DLFM configuration tasks” on page 37

**Creating a new UNIX file system (optional)**

*Creating a new UNIX file system* is part of the larger task of creating a new DLFS on an existing Data Links server.

This procedure is optional. You can use an existing JFS (AIX) or UFS (Solaris Operating Environment) file system as long as there is no conflict with having it be controlled by the Data Links Filesystem Filter (DLFF).

**Prerequisites:**

A user ID with root authority.

**Procedure:**

To create a new JFS or UFS:

1. Log on to the Data Links server as a user with root authority.
2. Create a new file system using the appropriate command.
  - On AIX systems, create a JFS using the **smitty manfs** command, and set the *Mount AUTOMATICALLY at system restart?* option to *no*. For more information about creating a new JFS, refer to your AIX product documentation or “man” pages.
  - On Solaris Operating Environments, create a UFS with the **newfs** command. For more information about the **newfs** command, refer to your Solaris product documentation or “man” pages.
3. Log out.

**Related tasks:**

- “Creating additional Data Links File Systems on a Data Links server” on page 38
- “Activating the Data Links Filesystem Filter” on page 44
- “Enabling file system sharing (AIX, Solaris Operating Environment)” on page 43
- “Enabling file system sharing (Windows)” on page 43

**Related reference:**



- “Basic DLFM configuration tasks” on page 37

### **Enabling file system sharing (AIX, Solaris Operating Environment)**

*Enabling file system sharing* is part of the larger task of creating a new DLFS on an existing Data Links server.

#### **Prerequisites:**

A user ID with root authority.

#### **Procedure:**

To enable sharing on a UNIX file system:

1. Log on to the Data Links server as a user with root authority.
2. Use the **exportfs** command to define the file system you want to convert into a DLFS as shareable (readable) by clients.

You can mark the exported file system to be read-only, to be read-write, or to be read-write only to specific client machines.

There are additional system sharing controls that you might want to specify. Refer to your AIX or Solaris Operating Environment system documentation for more details about the **exportfs** command and how to use it.

**AIX only:** You can also use SMIT to generate the **exportfs** command.

#### **Related tasks:**

- “Creating additional Data Links File Systems on a Data Links server” on page 38
- “Identifying DLFS paths to contain linked files” on page 40
- “Creating a new UNIX file system (optional)” on page 42
- “Activating the Data Links Filesystem Filter” on page 44
- “Enabling file system sharing (Windows)” on page 43

#### **Related reference:**

- “Basic DLFM configuration tasks” on page 37

### **Enabling file system sharing (Windows)**

*Enabling file system sharing* is part of the larger task of creating a new DLFS on an existing Data Links server.

#### **Prerequisites:**

A system administrator account (for example, the dlmadmin account).

#### **Procedure:**

To enable sharing on a Windows NTFS:

1. Log on to the Data Links server as Windows system administrator.
2. Click Start and select **Programs** —> **Windows Explorer**.
3. Right-click on the drive that you want to share and select **Sharing**. If you are already sharing the drive that you want to be controlled by a DLFS, skip to step 6.
4. Select **Shared As**.
5. Click **New Share**.
6. Enter a share name for this drive in the **Share Name** field and click **OK**.
7. Click **Permissions**.
8. Select the **Everyone** option.
9. Click **Type of Access** and select the **Full Control Option**.
10. Click **OK** to register the new share name.

**Related tasks:**

- “Creating additional Data Links File Systems on a Data Links server” on page 38
- “Identifying DLFS paths to contain linked files” on page 40
- “Creating a new UNIX file system (optional)” on page 42
- “Activating the Data Links Filesystem Filter” on page 44
- “Enabling and registering file systems with DLFM (AIX, Solaris Operating Environment)” on page 65

**Related reference:**

- “Basic DLFM configuration tasks” on page 37

**Activating the Data Links Filesystem Filter**

*Activating the Data Links Filesystem Filter (DLFF)* is part of the larger task of creating a new DLFS on an existing Data Links server. Activating the DLFF consists of two main tasks:

- Enabling the file system to use a DLFF
- Registering that file system with the Data Links File Manager (DLFM) on a Data Links server

**Prerequisites:**

- DB2 Data Links Manager Administrator authority.
- Additionally on UNIX systems, a user ID with root authority.
- Additionally on Windows, a superuser ID (for example, dlmadmin).

**Procedure:**

To activate the DLFF for a file system:

1. Log on to the machine where the file system is located, either as a user with root authority (UNIX) or as an Administrator (Windows).
2. The remaining steps depend on the specific type of file system you are using.
  - For both JFS and UFS, see Enabling and registering file systems with DLFM for the remaining procedures.
  - For Windows NTFS, see Registering a drive with DLFF for the remaining procedures.

**Related tasks:**

- “Enabling and registering file systems with DLFM (AIX, Solaris Operating Environment)” on page 65
- “Registering a drive with DLFF (Windows operating systems)” on page 66
- “Creating additional Data Links File Systems on a Data Links server” on page 38
- “Identifying DLFS paths to contain linked files” on page 40
- “Creating a new UNIX file system (optional)” on page 42
- “Enabling file system sharing (AIX, Solaris Operating Environment)” on page 43
- “Enabling file system sharing (Windows)” on page 43

**Related reference:**

- “Basic DLFM configuration tasks” on page 37

## **Modifying a Data Links File System**

After you create a Data Links File System (DLFS), you can change its size as your system needs dictate.

On the AIX JFS, you can take advantage of the functionality of the Tivoli Space Manager. The Tivoli Space Manager Hierarchical Storage Manager (HSM) client program automatically migrates eligible files to secondary storage to maintain specific levels of free space on local file systems. The prerequisite for this functionality is Tivoli Space Manager Version 4.2 or later.

You can enable any AIX-based DLFS to use the Tivoli Space Manager at any time. Conversely, you can also remove Tivoli Space Manager from a DLFS at any time.

**Related concepts:**

- “Before you install DB2 Data Links Manager (AIX)” in the *Quick Beginnings for Data Links Manager*

- “Tivoli Space Manager Hierarchical Storage Manager (AIX)” in the *Quick Beginnings for Data Links Manager*

**Related tasks:**

- “Changing the file system size” on page 46

**Related reference:**

- “Tivoli Storage Manager” in the *Data Recovery and High Availability Guide and Reference*
- “Basic DLFM configuration tasks” on page 37

## **Changing the size of a Data Links File System: tasks**

The following section describes how to change the size of a Data Links File system.

### **Changing the file system size**

Use these procedures to increase or decrease the size of a Data Links File System (DLFS) on Windows, or to decrease the size of a DLFS on UNIX systems.

The task of increasing the size of a DLFS on UNIX systems is explained in another topic, which is referenced at the end of this topic.

### **Prerequisites:**

You must shut down the Data Links File Manager (DLFM) on the Data Links server.

You must have superuser authority on the system that you will be using. On UNIX systems, you must have root authority. On Windows, you must use an Administrator account (for example, the dladmin account).

On Windows, you must also have access to a disk management program.

### **Restrictions:**

You can *increase* the size of a DLFS only if there is free space available on the physical hard disk (or hard disks) where the DLFS is located.

### **Procedure:**

To decrease space from an existing UNIX file system that is using a DLFF:

1. Log on to the system that contains the DLFS you want to modify as a user with root authority.

2. Unload the DLFF driver from the file system. The required steps are described in Loading, querying, and unloading a DLFF (AIX) and Loading, querying, and unloading a DLFF (Solaris Operating Environment).

3. Unmount the file system by entering one of the following commands:

```
/usr/opt/db2_08_01/instance/dlffmsmd -j filesystem_name //AIX  
/opt/IBM/db2/V8.1/instance/dlffmsmd -j filesystem_name //Solaris
```

where *filesystem\_name* specifies the name of the mounted file system that is using the DLFF.

4. Decrease the space allocation:
  - On AIX systems, refer to your AIX documentation for the specific steps required to decrease the space allocation of the base file system.
  - On Solaris Operating Environments, use a utility program, such as Solstice DiskSuite, to decrease the size of the file system. Refer to your utility program documentation for the the specific steps required to decrease the space allocation of the base file system.
5. Load the DLFF driver again. The required steps are described in Loading, querying, and unloading a DLFF (AIX) and Loading, querying, and unloading a DLFF (Solaris Operating Environment).

6. Mount the file system by entering one of the following commands:

```
/usr/opt/db2_08_01/instance/dlffmsmd dlfm_mountpoint //AIX  
/opt/IBM/db2/V8.1/instance/dlffmsmd dlfm_mountpoint //Solaris
```

where *dlfm\_mountpoint* is the mount point of the file system that you modified.

7. Log off.

To increase *or* decrease the size of a DLFS on Windows systems:

1. Log on to the computer that contains the DLFS that you want to modify as an Administrator user.
2. Launch a disk management program.  
You can either use the standard disk management program that comes with the Windows Administrative Tools (for example, Disk Management on Windows 2000), or a commercial program like Partition Magic.
3. Increase or decrease the size of the DLFS as necessary according to the disk management program instructions.

**Attention:** Regardless of whether you increase or decrease the DLFS, you must not change the drive letter and the share drive name of the DLFS drive. Making either of those changes will remove the file system from Data Links Manager control.

To decrease the size of a DLFS drive partition, you must divide the original DLFS drive into two partitions. You can clear space on the new partition as necessary.

**Important:** Any new drive partition created when splitting a DLFS drive will not be under control of the Data Links Filesystem Filter (DLFF), so it will not be a DLFS drive. To define a non-DLFS drive as a DLFS drive, use the **dlff add** command.

4. Reboot the computer.

The original modified DLFS drive will continue to be under the control of the Data Links Filesystem Filter (DLFF).

**Related tasks:**

- “Loading, querying, and unloading a DLFF (AIX)” on page 69
- “Loading, querying, and unloading a DLFF (Solaris Operating Environment)” on page 70
- “Increasing a DLFF-controlled file system size (AIX, Solaris Operating Environment)” on page 72

## **Adding DB2 databases to a DLFM on a Data Links server**

This procedure describes how to add a specific DB2 database to an existing Data Links server. As part of this procedure, you will associate a specific Data Links File Manager (DLFM) to a specific DB2 database.

**Prerequisites:**

- Data Links Manager Administrator authority.
- A valid DB2 user ID that has System Administrative (SYSADM) authority on the database instance that contains the database you want to use.

**Procedure:**

To add a specific DB2 database to an existing Data Links server:

1. Ensure that the specific DB2 database that you want to add already exists.  
See the Related tasks and Related concepts links for links to detailed information and instructions about installing DB2 and creating databases, if necessary.
2. Enable the specific DB2 database to use DB2 Data Links Manager.
  - a. Log on to the DB2 host system with a valid DB2 user ID that has System Administrative (SYSADM) authority on the instance that contains the database you want to use. By default, any user ID that belongs to the primary group of the instance owner has SYSADM authority on an instance.
  - b. Set the name of the DB2 instance containing the database to enable to use Data Links Manager by using one of the following commands:

```
export DB2INSTANCE=instance_name // UNIX Bash, Bourne, or Korn shell
setenv DB2INSTANCE=instance_name // UNIX C shell
set DB2INSTANCE=instance_name //Windows
```

c. Issue the following additional commands:

```
db2 get instance //optional, to ensure the database instance name set correctly
db2 update dbm cfg using datalinks yes
db2start
```

d. Log off.

3. Register the database you chose in the previous step with the Data Links File Manager (DLFM) on the Data Links server.

a. Log on to the Data Links server as the Data Links Manager Administrator.

b. Issue the following command: `dlfm add_db database instance hostname`

- *database* represents the database alias name of the remote database.
- *instance* represents the instance where database resides. If you are registering a Windows instance on a UNIX Data Links Manager, instance must be in uppercase.
- *hostname* represents the hostname of the DB2 UDB server where database resides.

c. Log off.

4. Register the Data Links server with the DB2 database.

a. Log on to the DB2 host system with a valid DB2 user ID that has System Administrative (SYSADM) authority on the instance that contains the database you just registered with the DLFM. By default, any user ID that belongs to the primary group of the instance owner has SYSADM authority on an instance.

b. Set the name of the DB2 instance containing the database to enable to use Data Links Manager by using one of the following commands:

```
export DB2INSTANCE=instance_name // UNIX Bash, Bourne, or Korn shell
setenv DB2INSTANCE=instance_name // UNIX C shell
set DB2INSTANCE=instance_name //Windows
```

c. Issue the following additional commands:

```
db2 get instance //optional, to ensure the database instance name set correctly
db2start
db2 add datalinks manager for database database_alias using node hostname port port_number
```

- *database\_alias* represents the database alias name of the database.
- *hostname* represents the fully qualified hostname of the Data Links server.
- *port\_number* represents the port number that you have reserved for communications between the Data Links server and the DB2 server. You specified this port number during the installation of DB2 Data Links Manager.

- d. Log off.
5. Grant the required permissions and authorities to the people and accounts who will be using the Data Links server.  
See the *authorizing users* topic in Related concepts for more information about granting user permissions and authorizations.

Repeat this entire procedure for each DB2 database that you want to add to a Data Links server.

**Related concepts:**

- “Installation overview for DB2 servers (UNIX)” in the *Quick Beginnings for DB2 Servers*
- “Installation overview for DB2 servers (Windows)” in the *Quick Beginnings for DB2 Servers*
- “Authorizing users to perform actions” on page 33

**Related reference:**

- “UPDATE DATABASE MANAGER CONFIGURATION Command” in the *Command Reference*
- “Enable Data Links Support configuration parameter - datalinks” in the *Administration Guide: Performance*
- “ADD DATALINKS MANAGER Command” in the *Command Reference*
- “dlfm add\_db command” on page 183
- “Basic DLFM configuration tasks” on page 37

### **Adding DLFMs to a specific DB2 database on the DB2 host**

This procedure describes how to add a new Data Links Manager to a specific DB2 database on the DB2 host. As part of this procedure, you will associate a specific Data Links File Manager (DLFM) to a specific DB2 database.

**Prerequisites:**

- Data Links Manager Administrator authority.
- A valid DB2 user ID that has System Administrative (SYSADM) authority on the database instance that contains the database you want to use.

**Restrictions:**

A single DB2 database can communicate with a maximum of 16 DLFMs.

**Procedure:**

To add a new Data Links Manager to a specific DB2 database on the DB2 host:



1. Install DB2 Data Links Manager on the machine that you want to use as a Data Link server, if it is not already installed.

If you do not already have Data Links Manager installed, see the end of this topic for links to detailed information and instructions about installing Data Links Manager.

2. Create and configure the Data Links server, if necessary.

If you do not already have a Data Links server configured and ready, you must do the following:

- Identify the Data Links File System (DLFS) paths to contain linked files.
- Create a new native file system, if required.
- Enable file system sharing.
- Turn the native file system into a DLFS by activating the Data Links Filesystem Filter (DLFF).

As part of this task, you actually define DLFS paths to the DLFM with the **dlfm add\_prefix** command.

- Enable client connectivity to the DLFS.

See Related tasks for links to detailed information and instructions about each of these procedures.

3. Enable a specific DB2 database to use DB2 Data Links Manager.

- a. Log on to the DB2 host system with a valid DB2 user ID that has System Administrative (SYSADM) authority on the instance that contains the database you want to use. By default, any user ID that belongs to the primary group of the instance owner has SYSADM authority on an instance.

- b. Set the name of the DB2 instance containing the database to enable to use Data Links Manager by using one of the following commands:

```
export DB2INSTANCE=instance_name // UNIX Bash, Bourne, or Korn shell
setenv DB2INSTANCE=instance_name // UNIX C shell
set DB2INSTANCE=instance_name //Windows
```

- c. Issue the following additional commands:

```
db2 get instance //optional, to ensure the database instance name set correctly
db2 update dbm cfg using datalinks yes
db2start
```

- d. Log off.

4. Register the database you chose in the previous step with the Data Links File Manager (DLFM) on the Data Links server.

- a. Log on to the Data Links server as the Data Links Manager Administrator.

- b. Issue the following command: **dlfm add\_db database instance hostname**

- *database* represents the database alias name of the remote database.

- *instance* represents the instance where database resides. If you are registering a Windows instance on an AIX Data Links Manager, instance must be in uppercase.
  - *hostname* represents the hostname of the DB2 UDB server where database resides.
- c. Log off.
5. Register the Data Links server with the DB2 database.
    - a. Log on to the DB2 host system with a valid DB2 user ID that has System Administrative (SYSADM) authority on the instance that you want. By default, any user ID that belongs to the primary group of the instance owner has SYSADM authority on an instance.
    - b. Set the name of the DB2 instance containing the database to enable to use Data Links Manager by using one of the following commands:
 

```
export DB2INSTANCE=instance_name // UNIX Bash, Bourne, or Korn shell
setenv DB2INSTANCE=instance_name // UNIX C shell
set DB2INSTANCE=instance_name //Windows
```
    - c. Issue the following additional commands:
 

```
db2 get instance //optional, to ensure the database instance name set correctly
db2start
db2 add datalinks manager for database database_alias using node hostname port port_number
```

      - *database\_alias* represents the database alias name of the database.
      - *hostname* represents the fully qualified hostname of the Data Links server.
      - *port\_number* represents the port number that you have reserved for communications between the Data Links server and the DB2 server. You specified this port number during the installation of DB2 Data Links Manager.
    - d. Log off.
  6. Grant the required permissions and authorities to the people and accounts who will be using the Data Links server, if necessary.
 

See the Related concepts links for more information about granting user permissions and authorizations.

Repeat this entire procedure for each Data Links Manager that you want to add to a specific DB2 database. A single DB2 database can communicate with up to 16 DLFMs.

**Related concepts:**

- “Before you install DB2 Data Links Manager (AIX)” in the *Quick Beginnings for Data Links Manager*
- “Before you install DB2 Data Links Manager (Solaris Operating Environment)” in the *Quick Beginnings for Data Links Manager*

- “Before you install DB2 Data Links Manager (Windows)” in the *Quick Beginnings for Data Links Manager*
- “Authorizing users to perform actions” on page 33

**Related tasks:**

- “Creating additional Data Links File Systems on a Data Links server” on page 38

**Related reference:**

- “UPDATE DATABASE MANAGER CONFIGURATION Command” in the *Command Reference*
- “Enable Data Links Support configuration parameter - datalinks” in the *Administration Guide: Performance*
- “ADD DATALINKS MANAGER Command” in the *Command Reference*
- “dlfm add\_db command” on page 183
- “Basic DLFM configuration tasks” on page 37

## **Dropping Data Links Manager from a specific DB2 database on the DB2 host**

The process of removing information about Data Links Manager from a DB2<sup>®</sup> database is referred to as *dropping* Data Links Manager from a DB2 database.

You can drop a Data Links Manager from a DB2 database when necessary. For example, if you no longer use a Data Links Manager on a particular database, you can reduce processing overhead by removing the Data Links Manager from that database. Or, you might need to remove an incorrectly defined Data Links Manager from a database.

The process of dropping a Data Links Manager from a DB2 database is not difficult, but it is a major change to your database environment. Also, you *cannot* roll back the changes made to a database as a result of dropping a Data Links Manager.

See the links at the end of this topic for detailed information about the potential consequences of dropping a Data Links Manager.

**Prerequisites:**

Data Links Manager Administrator authority.

Because of the potential impact to your data, do the following tasks in preparation for dropping a Data Links Manager from a database:

- Ensure that all replication subscriptions have replicated all changes involving the Data Links Managers that you want to drop.

- Back up the database from which you want to drop the Data Links Manager.
- Ensure that the database from which you want to drop Data Links Manager does not contain *any* references to *any* files on the Data Links server for that Data Links Manager.

Files corresponding to links between a database and a dropped Data Links Manager remain linked, and will be inaccessible to operations such as read, write, rename, delete, change of permissions, or change of ownership until the Data Links Manager has been completely dropped as described in this procedure.

### **Procedure:**

To drop a Data Links Manager from a database:

1. Log on to the DB2 host that contains the DB2 database instance from which you want to drop a Data Links Manager.
2. Issue the **DROP DATALINKS MANAGER** command, using the name of the database from which you want to drop a Data Links Manager, and the Data Links server that contains the Data Links Manager.

If the database uses multiple Data Links servers, some or all of which are to be dropped, issue this command for each Data Links server to be dropped.

3. Using the Data Links Manager Administrator account, log onto the Data Links server machine that corresponds to the DB2 database instance from which you have dropped the Data Links Manager.
4. Issue the **dlfm drop\_dlm** command.

If you dropped multiple Data Links servers in step 2, repeat steps 3 – 4 on each Data Links server machine that you dropped.

### **Related tasks:**

- “Removing information about a DB2 database from Data Links Manager” on page 75

### **Related reference:**

- “DROP DATALINKS MANAGER Command” in the *Command Reference*
- “Basic DLFM configuration tasks” on page 37
- “dlfm drop\_dlm command” on page 189

## **Enabling the Data Links Manager Replication daemon**

When replicating linked files using DB2 DataPropagator (DB2 replication), you can customize how external files get copied from the source file system to the target file system. You can either use an FTP daemon, or the Data Links

Manager Replication daemon, DLFM\_ASNCOPYD, which is included with Data Links Manager. Both daemons work with the DB2 replication ASNDLCOPY exit routine to copy the files that are referenced by DATALINK column values. The DLFM\_ASNCOPYD daemon supports basic FTP commands, and provides the following additional functionality:

- Allows retrieval of a particular version of a file that is referenced by a DATALINK column defined with RECOVERY YES
- Allows retrieval of files referenced by DATALINK columns defined with READ PERMISSION DB depending on the access privilege of the user
- Provides the ability to preserve the last modification time of replicated files

DLFM\_ASNCOPYD is launched during the Data Links Manager startup time as a component process. Any authorized user can connect to DLFM\_ASNCOPYD through a dedicated port.

During the Data Links Manager installation process, you are prompted to enable DLFM\_ASNCOPYD, and to specify a port for the daemon to use. The default is to leave the daemon disabled. However, you can change your Data Links Manager configuration at any time to enable DLFM\_ASNCOPYD.

### **Prerequisites:**

You must have DB2 Data Links Manager Administrator authority.

### **Procedure:**

To enable the Data Links Manager Replication daemon on a Data Links server:

1. Log on to the specific Data Links server machine where you want to enable the Data Links Manager Replication daemon.
2. Use the **db2set** command to set the registry variable DLFM\_START\_ASNCOPYD to be YES.
3. Use the **db2set** command to set the registry variable DLFM\_ASNCOPYD\_PORT to be a value from 1 to 65535. This port number value must match the port number value that is specified in the ASNDLCOPY exit routine ASNDLUSER configuration file, in DB2 DataPropagator.

**UNIX systems only:** The port number should be reserved in /etc/services.

4. Restart the Data Links File Manager (DLFM) by invoking the **dlfm restart** command.

### **Related concepts:**

- “Before you install DB2 Data Links Manager (AIX)” in the *Quick Beginnings for Data Links Manager*

- “Before you install DB2 Data Links Manager (Solaris Operating Environment)” in the *Quick Beginnings for Data Links Manager*
- “Before you install DB2 Data Links Manager (Windows)” in the *Quick Beginnings for Data Links Manager*
- “DB2 registry and environment variables” in the *Administration Guide: Performance*

**Related tasks:**

- “Replicating special data types” in the *Replication Guide and Reference*

**Related reference:**

- “db2set - DB2 Profile Registry Command” in the *Command Reference*
- “dlfm grant replication read command” on page 191
- “dlfm grant replication write command” on page 193
- “dlfm revoke replication command” on page 205

---

## Configuration tuning

To maintain optimal system performance, check the following aspects of a Data Links server configuration on a periodic basis, and adjust them as needed:

- The machine system clock
- The amount of storage space available for Data Links File Manager (DLFM) database logging
- The number of active copy processes

**Related concepts:**

- “Synchronizing system clocks” on page 57
- “Ensuring sufficient DB2 log space for the DLFM\_DB” on page 57

**Related tasks:**

- “Setting the number of copy processes” on page 58

---

## Configuration tuning: details

The following sections describe:

- How to check and synchronize all system clocks on all machines.
- Suggestions for maintaining adequate disk space for Data Links File Manager (DLFM) database log files.
- How to configure the number of active copy daemon processes.

## Synchronizing system clocks

The system clocks on all Data Links server machines and all associated DB2® host machines must be synchronized, and must remain synchronized at all times.

Synchronization of machine system clocks is essential for file access token expiry intervals to work correctly. File access token expiry intervals control how long a selected DATALINK column value (consisting of a URL with an embedded file authorization token) can be used.

- To check the system time and date on an AIX® machine, use the **date -u** command. See your AIX administration documentation for information about how to set the system time and date.
- To check the system time and date on a machine that is using the Solaris Operating Environment, use the **date** command. See your Solaris Operating Environment administration documentation for information about how to set the system time and date.
- To set and check the system time and date on a Windows® machine, use the **Date/Time** application in the Control Panel.

### Related reference:

- “Data Links Access Token Expiry Interval configuration parameter - dl\_expint” in the *Administration Guide: Performance*
- “Data Links Write Token Initial Expiry Interval configuration parameter - dl\_wt\_iexpint” in the *Administration Guide: Performance*

## Ensuring sufficient DB2 log space for the DLFM\_DB

Like all DB2® databases, the Data Links File Manager (DFLM) database DLFM\_DB maintains several log files. The log files for DLFM\_DB are stored on the Data Links server machine where the DLFM resides.

**Attention:** If there is not enough disk space for the DLFM\_DB logs, *the DLFM is at risk of shutting down.*

DB2 issues warning messages when disk capacity is running low for logging purposes. You should also periodically monitor your Data Links server machines to ensure that adequate disk space remains available.

To minimize tuning for log file capacity issues, you can optimize the following database configuration variables on a Data Links server machine for the DLFM\_DB database: LOGFILSIZ, LOGPRIMARY, and LOGSECOND.

### Related concepts:

- “Space requirements for log files” in the *Administration Guide: Planning*

**Related reference:**

- “UPDATE DATABASE MANAGER CONFIGURATION Command” in the *Command Reference*
- “Configuration Parameters for Database Logging” in the *Data Recovery and High Availability Guide and Reference*

**Setting the number of copy processes**

Data Links Manager uses copy daemon processes to copy linked files to an archive area or server. The speed at which files get copied to an archive area can impact your overall system performance. You can configure the number of enabled copy daemon processes on specific Data Links servers by optimizing the value of the registry variable `DLFM_NUM_ARCHIVE_SUBSYSTEMS`. The default value is two.

**Recommendation:** If your system is in a high parallelism environment, you can improve your overall Data Links Manager system performance by setting the `DLFM_NUM_ARCHIVE_SUBSYSTEMS` value to correspond with the number of independent I/O channels in your storage system. Your storage system could consist of multiple independent I/O channels, such as local storage devices (for example, hard disks). IBM Tivoli Storage Manager and XBSA-compliant storage applications also support high parallelism environments, because they can be configured to use independent I/O devices.

**Prerequisites:**

Determine the optimal number of copy daemon processes for your system. You can use either or both of the following methods:

- Analyze the storage system hardware and software that comprise your archive file system.
- Change the registry variable value, then test the impact of the change on system performance. Repeat these actions as necessary until you have optimized system performance.

**Important:** Setting the `DLFM_NUM_ARCHIVE_SUBSYSTEMS` registry value to a very high number can *degrade* system performance. To *improve* system performance, it is essential that you optimize the registry value based on the methods described.

You must have DB2 Data Links Manager Administrator authority.

**Procedure:**

To set the number of copy daemon processes on a Data Links server:



1. Log on to the specific Data Links server machine where you want to set the number of copy daemon processes.
2. Use the **db2set** command to set the registry variable `DLFM_NUM_ARCHIVE_SUBSYSTEMS` to be a value from 1 to 64.
3. Restart the Data Links File Manager (DLFM) by invoking the **dlfm restart** command.

**Related concepts:**

- “DB2 registry and environment variables” in the *Administration Guide: Performance*
- “Data Links server file backups” on page 21

**Related reference:**

- “db2set - DB2 Profile Registry Command” in the *Command Reference*



---

## Chapter 3. Working with the Data Links File Manager

This chapter describes the day-to-day operating procedures used by a Data Links Manager Administrator to maintain the Data Links File Manager (DLFM) on a Data Links server.

Unless otherwise noted, the commands in this chapter are common to Data Links servers running on AIX systems, Solaris Operating Environments, and Windows operating systems.

- For a complete reference of all Data Links File Manager (DLFM) commands for all operating systems, see Appendix A, “Data Links File Manager commands” on page 183.
- For a complete reference of all Data Links Filesystem Filter (DLFF) commands specifically for Windows operating systems, see Appendix B, “Data Links Filesystem Filter commands” on page 213.

---

### Basic operations: starting, stopping, and restarting the DLFM

There are several reasons that you might need to start the Data Links File Manager (DLFM) component of a Data Links server:

- To access and link files (with the exception of files referenced from a DATALINK column with the READ PERMISSION attribute set to FS)
- To add a new file system directory
- To create files that are to be linked or re-linked
- To update files that are to be linked or re-linked

There are many reasons that you might need to stop the DLFM component of a Data Links server:

- To modify a DLFM configuration
- To refresh read and write tokens
- To perform maintenance on a Data Links server machine
- To support a planned machine outage
- For testing or troubleshooting purposes
- Because a Data Links server machine fails
- Because a DB2<sup>®</sup> database fails

Restarting a DLFM means performing a stop operation, then performing a start operation. You might need to restart a DLFM for the same reasons as performing a stop or a start operation.

Use the **dlfm start**, **dlfm stop**, and **dlfm restart** commands to start, stop, and restart (that is, start then stop in a single operation) a DLFM.

If a **dlfm stop** action is not completely successful, you might need to use the **dlfm shutdown** command. The Related links section contains references to more information about abnormal terminations and the **dlfm shutdown** command.

**Windows® Operating Systems only:** If you have DLFM started as a service, use the Services panel, which is accessed from the Control Panel, to stop and start the DLFM service.

**Related concepts:**

- “DB2 Data Links Manager recovery scenarios” on page 148

**Related tasks:**

- “Restarting the DLFM after an abnormal termination” on page 62

**Related reference:**

- “dlfm restart command” on page 202
- “dlfm shutdown command” on page 208
- “dlfm start command” on page 208
- “dlfm stop command” on page 209

---

## Restarting the DLFM after an abnormal termination

When you cannot stop the DLFM using the **dlfm stop** command, shut down the DLFM and then restart it according to these instructions. You might also be directed to use these instructions at the direction of IBM service, should the DLFM terminate abnormally for some reason.

**Prerequisites:**

You must have DB2 Data Links Manager Administrator authority.

**Restrictions:**

On AIX and the Solaris operating environment: do not use the kill or sigkill -9 signal to stop DLFM processes, where 9 is sigkill. Use the commands documented here.

**Procedure:**

To restart the DLFM after an abnormal termination:

1. Log on to the Data Links server as the DB2 Data Links Manager Administrator.
2. Enter the **dlfm shutdown** command to bring down the active DLFM.  
**Attention:** Running the **dlfm shutdown** command removes *all* shared resources, including all IPCs, for the account that you are using (in this situation, the Data Links Manager Administrator account). If you are running other processes under this same account, those processes will probably terminate.
3. Enter the **dlfm start** command to start the DLFM.

**Related concepts:**

- “DB2 Data Links Manager recovery scenarios” on page 148
- “Basic operations: starting, stopping, and restarting the DLFM” on page 61

**Related reference:**

- “dlfm shutdown command” on page 208
- “dlfm start command” on page 208
- “dlfm stop command” on page 209

## Monitoring the DLFM background processes (AIX, Solaris Operating Environment)

A DLFM agent background process starts for every connection that DB2 makes to a Data Links server on an AIX system or on a Solaris Operating Environment.

**Procedure:**

To monitor the background processes of a Data Links File Manager (DLFM) on a Data Links server:

1. Log on to the specific Data Links server machine where you want to monitor the DLFM processes.
2. Enter the **dlfm see** command.
  - If the background processes are running successfully, you receive output similar to the following:

PID	PPID	PGID	RUNAME	UNAME	ETIME	DAEMON	NAME
71326	185894	119252	root	dlfm	00:07	dlfm_gcd_	(dlfm)
75788	185894	119252	root	dlfm	00:07	dlfm_deigrpd_	(dlfm)
100042	100280	119252	root	dlfm	00:08	dlfm_agent_	(dlfm)
100280	185894	119252	root	dlfm	00:07	dlfm_cmgrd_	(dlfm)
154834	234604	119252	root	dlfm	00:05	dlfm_ar_ag_	(dlfm)
185894	1	119252	root	dlfm	00:08	dlfm_wd_	(dlfm)
210534	234604	119252	root	dlfm	00:05	dlfm_ar_ag_	(dlfm)
226860	185894	119252	root	dlfm	00:07	dlfm_upca11d_	(dlfm)

```
234604 185894 119252 root dlfm 00:07 dlfm_archived_(dlfm)
250654 185894 119252 root dlfm 00:07 dlfm_mon_wd_(dlfm)
```

DLFM222I: The "DLFM SEE" request was successful.

- If the background processes did not start successfully, the **dlfm see** command does not return any output.

**Related reference:**

- “dlfm see command (AIX and Solaris Operating Environment)” on page 206

---

## Monitoring the DLFM background processes (Windows)

A DLFM agent background process starts for every connection that DB2 makes to a Data Links server on a Windows system.

**Procedure:**

To monitor the background processes of a Data Links File Manager (DLFM) on a Data Links server:

1. Log on to the specific Data Links server machine where you want to monitor the processes.
2. Open the Windows Task Manager.
3. The DLFM background processes display with all other processes that are currently running on the Data Links server machine.

DLFM background processes are shown in the format `dlfm_processname.exe`, where *processname* is the name of the specific process that is running.

When the DLFM is running successfully, you will see the following background processes:

- `dlfm_ar_ag.exe`
- `dlfm_ar_ag.exe`
- `dlfm_archived.exe`
- `dlfm_cmgrd.exe`
- `dlfm_delgrpd.exe`
- `dlfm_gcd.exe`
- `dlfm_mon_wd.exe`
- `dlfm_upcalld.exe`
- `dlfm_wd.exe`

Other processes might also be shown, depending on the particular activity or operation that the DLFM is performing at that moment.

**Related reference:**

- “dlff get loglevel command (Windows operating system)” on page 215
- “dlff refreshtrace command (Windows operating system)” on page 216
- “dlff set loglevel command (Windows operating system)” on page 221

---

## Enabling and registering file systems with DLFM (AIX, Solaris Operating Environment)

To enable the Data Links File Manager (DLFM) on a Data Links server to recognize a Solaris UNIX file system (UFS) or an AIX Journaled file system (JFS), you must do the following in the order listed:

1. Enable the file system to use a Data Links Filesystem Filter (DLFF).
2. Register that file system with the DLFM.

### Prerequisites:

- You must have DB2 Data Links Manager Administrator authority.
- You must have system root authority.

### Procedure:

To enable a file system to use DLFF and register the file system with DLFM:

1. Log on to the system as a user with root authority.
2. Choose the file system for which you want to enable a DLFF. You can either create a new file system, or use an existing file system.
  - **On Solaris Operating Environments**, create a UFS with the **newfs** command. For more information about the **newfs** command, refer to your Solaris product documentation or “man” pages.
  - **On AIX systems**, create a JFS using the **smit manfs** command, and set the *Mount AUTOMATICALLY at system restart?* option to *no*. For more information about creating a new JFS, refer to your AIX product documentation or “man” pages.
3. Mount the file system by invoking one of the following scripts:

```
/opt/IBM/db2/V8.1/instance/dlfmsmd dlfm_mountpoint //Solaris Operating Environments
/usr/opt/db2_08_01/instance/dlfmsmd dlfm_mountpoint //AIX systems
```

where *dlfm\_mountpoint* is the mount point from the previous step.

As part of the mounting process, the `dlfmsmd` script automatically updates the necessary files, parameters, and attributes.

- **On Solaris Operating Environments**, the following entries in the `/etc/vfstab` file get modified:

```
/dev/dsk/c0t0d0s6 /dev/rdisk/c0t0d0s6 /dlfs dlfs - yes Basefs=ufs
```

where *c0t0d0s6* is a sample value for this example.

- **On AIX systems**, the following actions occur:
  - In the `/etc/filesystems` file, the current setting for the *vfs* entry gets recorded, and the stanza gets edited as follows:

```
mount    = false
vfs      = dlfs
```
  - The following *nodename* attribute gets added to the stanza for the file system entry:

```
nodename = -
```

After the script has run, you should verify that there are no blank spaces after the null indicator character ( - ).

- The *Basefs* parameter in the *options* attribute gets set to the value that was recorded for the *vfs* entry above. For example, the entry would be similar to the following:

```
options = rw,Basefs=jfs
```

4. Log off.
5. Log on to the system as the DB2 Data Links Manager Administrator.
6. Start the Data Links File Manager by entering the **dlfm start** command.
7. Register a file system that is under the control of a DLFF by entering the following command:

```
dlfm add_prefix prefix_path
```

where *prefix\_path* is the location of the file system that is under the control of a DLFF.

**Example:** Register the Data Links server to use the DLFF on the `test` file system by entering the following command:

```
dlfm add_prefix /test
```

**Related tasks:**

- “Listing registered file systems (AIX, Solaris Operating Environment)” on page 68

**Related reference:**

- “dlfm add\_prefix command” on page 184
- “dlfm start command” on page 208

---

## Registering a drive with DLFF (Windows operating systems)

To enable the Data Links File Manager (DLFM) on a Data Links server to recognize a Windows NT or Windows 2000 drive, you must do the following in the order listed:



1. Enable the file system (NTFS for both Windows NT and Windows 2000) on the Data Links server drive to use a Data Links Filesystem Filter (DLFF).
2. Register that drive with the DLFM.

**Prerequisites:**

You must be a member of the Windows Administrator group.

**Procedure:**

To enable an NTFS file system to use DLFF and register the drive with DLFM:

1. Log on to the system and open a command prompt session.
2. Enter the following command to place the drive under control of the DLFF:

```
dlff add drive:
```

where *drive*: is the letter of the drive that you want to place under the control of the DLFF. For example, if the drive letter is **d:**, you would enter the command `dlff add d:`.

3. Enter the following command to register the drive with the Data Links server DLFM:

```
dlfm add_prefix \sharename
```

where *sharename* is the name of an existing sharename on the drive you just added. For example, if the sharename is **grepository**, you would enter the command `dlfm add_prefix \grepository`.

**Attention:** Once you add a prefix you cannot remove it.

**Important:** If you want to use the drive that you just added as a sharename, and that drive does yet not have a sharename associated with it, you must associate a sharename with the drive before running the **dlfm add\_prefix** command. Open **Windows Explorer** or **My Computer** and modify the drive's properties to assign it a sharename.

**Related tasks:**

- "Listing registered drives (Windows operating systems)" on page 68

**Related reference:**

- "dlff add command (Windows operating system)" on page 213
- "dlff list command (Windows operating system)" on page 215

---

## Listing registered file systems (AIX, Solaris Operating Environment)

Use this procedure to list all file systems that are under control of a Data Links Filesystem Filter (DLFF) *and* registered with the Data Links File Manager (DLFM) on a Data Links server.

### Prerequisites:

You must have DB2 Data Links Manager Administrator authority.

### Procedure:

To list any file systems that are currently under the control of a DLFF and registered with DLFM:

1. Log on to the system as the DB2 Data Links Manager Administrator.
2. To list all of the registered prefixes on the Data Links server, enter the **dlfm list registered prefixes** command.
3. To list the registered file systems where a DLFF is loaded, use the appropriate commands:

On AIX systems, enter the **lsfs -v dlfs** command to list the Data Links File Systems (DLFSs) that are currently defined. Then, enter the **mount -f | awk '\$3 == "dlfs"'** command to list the DLFSs for which the DLFF is loaded.

On Solaris Operating Environments, enter the **cat /etc/vfstab | awk '\$4 == "dlfs"'** command to list the DLFSs that are currently defined. Then, enter the **/sbin/mount -v | awk '\$5 == "dlfs"'** command to list the DLFSs for which the DLFF is loaded.

### Related tasks:

- “Enabling and registering file systems with DLFM (AIX, Solaris Operating Environment)” on page 65

### Related reference:

- “dlfm list registered prefixes command” on page 196

---

## Listing registered drives (Windows operating systems)

Use this procedure to list all NTFS drives that are under control of a Data Links Filesystem Filter (DLFF) *and* registered with the Data Links File Manager (DLFM) on a Data Links server.

### Prerequisites:

- DLFM must be running as a Windows service. If DLFM is not already running, use the **dlfm start** command to start the DLFM.
- You must be a member of the Windows Administrator group.

**Procedure:**

To list any NTFS drives that use a DLFF and are registered with DLFM on a Data Links server:

1. Log on to the system as the DB2 Data Links Manager Administrator.
2. Open a command prompt session.
3. Enter the **dlfm list registered prefixes** command on the Data Links server.
4. To list any drives where a DLFF is loaded, enter the **dlff list** command.

**Related tasks:**

- “Registering a drive with DLFF (Windows operating systems)” on page 66

**Related reference:**

- “dlff list command (Windows operating system)” on page 215
- “dlfm list registered prefixes command” on page 196

## Loading, querying, and unloading a DLFF (AIX)

The following procedures describe how to load, query, and unload a Data Links Filesystem Filter (DLFF). Reference these instructions when installing FixPaks, debugging programs, or performing maintenance routines.

**Prerequisites:**

- The DLFF driver must be loaded before starting a Data Links File Manager (DLFM), otherwise the DLFM will not be able to start. This prerequisite does *not* require that a Data Links File System (DLFS) also be mounted.
- You must have root authority on the system you will be using.

**Procedure:**

To load a DLFF:

1. Log on to the system as a user with root authority.
2. Enter the **strload -f /usr/opt/db2\_08\_01/cfg/dlfs\_cfg** command.

To query a DLFF:

1. Log on to the system as a user with root authority.
2. Enter the **strload -q -f /usr/opt/db2\_08\_01/cfg/dlfs\_cfg** command.

To unload a DLFF:

1. Log on to the system as a user with root authority.
2. Enter the **strload -u -f /usr/opt/db2\_08\_01/cfg/dlfs\_cfg** command.

**Related tasks:**

- “Enabling and registering file systems with DLFM (AIX, Solaris Operating Environment)” on page 65
- “Listing registered file systems (AIX, Solaris Operating Environment)” on page 68

---

## Loading, querying, and unloading a DLFF (Solaris Operating Environment)

The following procedures describe how to load, query, and unload a Data Links Filesystem Filter (DLFF). Reference these instructions when installing FixPaks, debugging programs, or performing maintenance routines.

**Prerequisites:**

- The DLFF driver must be loaded before starting a Data Links File Manager (DLFM), otherwise the DLFM will not be able to start. This prerequisite does *not* require that a Data Links File System (DLFS) also be mounted.
- You must have root authority on the system you will be using.

**Procedure:**

To load a DLFF:

1. Log on to the system as a user with root authority.
2. Enter the **add\_drv -m '0777 dlfm dlfmgrp' dlfsdrv** command. In this example, it is assumed that the dlfm id is created on the machine in the primary group dlfmgrp.

To query a DLFF:

1. Log on to the system as a user with root authority.
2. Enter the **/usr/sbin/modinfo | grep dlfs** command.

To unload a DLFF:

1. Log on to the system as a user with root authority.
2. Enter the **rem\_drv dlfsdrv** command.

**Related tasks:**

- “Enabling and registering file systems with DLFM (AIX, Solaris Operating Environment)” on page 65

- “Listing registered file systems (AIX, Solaris Operating Environment)” on page 68

---

## Registering, querying and de-registering a DLFF (Windows operating system)

The following procedures describe how to register, query (or list) and deregister a Data Links Filesystem Filter (DLFF) on Windows NT and Windows 2000 drives. Reference these instructions when installing FixPaks, debugging programs, or performing maintenance routines.

### Prerequisites:

- The DLFF driver must be loaded before starting a Data Links File Manager (DLFM), otherwise the DLFM will not be able to start. This prerequisite does *not* require that a Data Links File System (DLFS) also be layered over any drive.
- You must log in with the Data Links Manager Administrator account on the system you will be using.

### Procedure:

To register a DLFF:

1. Log on to the system using the Data Links Manager Administrator account.
2. Open a Windows command prompt session.
3. Enter the **dlff add** *drive* command. The *drive* parameter is the disk drive for which you want to load a DLFF.

To query a DLFF:

1. Log on to the system using the Data Links Manager Administrator account.
2. Open a Windows command prompt session.
3. Enter the **dlff list** command.

To deregister a DLFF:

1. Log on to the system using the Data Links Manager Administrator account.
2. Open a Windows command prompt session.
3. Enter the **dlff remove** *drive* command. The *drive* parameter is the disk drive from which you want to unload a DLFF.
4. Reboot the machine.

The Data Links Filesystem Filter is deregistered.

**Related tasks:**

- “Registering a drive with DLFF (Windows operating systems)” on page 66
- “Listing registered drives (Windows operating systems)” on page 68

**Related reference:**

- “dlff add command (Windows operating system)” on page 213
- “dlff remove command (Windows operating system)” on page 217
- “dlff list command (Windows operating system)” on page 215

---

## Increasing a DLFF-controlled file system size (AIX, Solaris Operating Environment)

This procedure describes how to increase the size of a file system using a Data Links Filesystem Filter (DLFF) on both AIX systems and Solaris Operating Environments.

**Prerequisites:**

You must have root authority on the system you will be using.

**Procedure:**

To allocate more space to an existing file system that is using a DLFF:

1. Log on to the system as a user with root authority.
2. Modify the properties of the file system so that it is no longer under the control of the DLFF. Then unmount the file system by entering one of the following commands:

```
/usr/opt/db2_08_01/instance/dlffmsmd -j filesystem_name //AIX systems  
/opt/IBM/db2/V8.1/instance/dlffmsmd -j filesystem_name //Solaris Operating Environments
```

where *filesystem\_name* specifies the name of the mounted file system that is using the DLFF.

3. On AIX systems, enter the **smit jfs** command to increase the size of the file system.

On Solaris Operating Environments, use a utility program, such as Solstice DiskSuite, to increase the size of the file system.

4. Modify the properties of the file system so that it comes under the control of the DLFF. Then mount the file system by entering one of the following commands:

```
/usr/opt/db2_08_01/instance/dlffmsmd dlfm_mountpoint //AIX systems  
/opt/IBM/db2/V8.1/instance/dlffmsmd dlfm_mountpoint //Solaris Operating Environment
```

where *dlfm\_mountpoint* is the mount point of the file system that you created for the DLFF in the previous step.

5. Log off.

**Related tasks:**

- “Enabling and registering file systems with DLFM (AIX, Solaris Operating Environment)” on page 65
- “Listing registered file systems (AIX, Solaris Operating Environment)” on page 68

---

## Registering DB2 Universal Databases with the Data Links server

To enable the Data Links File Manager (DLFM) on a Data Links server to recognize a DB2 database, you must register that DB2 database with the DLFM.

**Prerequisites:**

You must have DB2 Data Links Manager Administrator authority.

**Procedure:**

To register a DB2 database with the DLFM on a Data Links server:

1. Log on to the system as the DB2 Data Links Manager Administrator.
2. Register the remote DB2 Universal Database where the DATALINK data type was defined by entering the following command:

```
dlfm add_db database instance hostname
```

where:

- *database* is the database alias of the remote database.
- *instance* is the name of the instance where the database resides. If you are registering a Windows NT or Windows 2000 instance on an AIX or Solaris Data Links Manager, enter the instance name in uppercase.
- *hostname* represents the hostname of the DB2 UDB server where the database resides. See the links for related concepts, below, for more information about how to determine the DB2 UDB server hostname.

For example, the following command registers a database called STAFF.

```
dlfm add_db staff validate db2server.services.com
```

The database resides on the `validate` instance on a DB2 Universal Database with a hostname of `db2server.services.com`.

3. Log off.

Each time you register a database with a Data Links File Manager, the affected DLFM\_DB tablespace is automatically backed up. The DLFM\_DB is a DB2 database that acts as a logging manager for the DLFM on a Data Links server.

**Related concepts:**

- “Before you install DB2 Data Links Manager (AIX)” in the *Quick Beginnings for Data Links Manager*
- “Before you install DB2 Data Links Manager (Solaris Operating Environment)” in the *Quick Beginnings for Data Links Manager*
- “Before you install DB2 Data Links Manager (Windows)” in the *Quick Beginnings for Data Links Manager*

**Related tasks:**

- “Listing databases registered with the Data Links server” on page 74

**Related reference:**

- “dlfm add\_db command” on page 183

---

## Listing databases registered with the Data Links server

This procedure explains how to show all DB2 databases that are registered with a Data Links File Manager (DLFM) on a Data Links server.

**Prerequisites:**

You must have DB2 Data Links Manager Administrator authority.

**Procedure:**

To list the databases that are registered with a Data Links server:

1. Log on to the system as the DB2 Data Links Manager Administrator.
2. Enter the **dlfm list registered databases** command on the Data Links server.

**Related tasks:**

- “Registering DB2 Universal Databases with the Data Links server” on page 73

**Related reference:**

- “dlfm list registered databases command” on page 194



---

## Removing information about a DB2 database from Data Links Manager

You can remove (drop) a Data Links Manager from a DB2 database when necessary. For example, if you no longer use a Data Links Manager on a particular database, you can reduce processing overhead by removing the Data Links Manager from that database. Or, you might need to remove an invalid Data Links Manager from a database.

When you drop a DB2 Data Links Manager from a database using the **DROP DATALINKS MANAGER** command, the command itself does not clean up the corresponding information on the DB2 Data Links Manager. You must explicitly initiate unlinking of any files linked to the database, which enables the clean up, at a later time, of backup and file management security information. The clean up operations get done using the **dlfm drop\_dlm** command. The **dlfm drop\_dlm** command initiates asynchronous deletion of all information for a particular database.

You must invoke the **dlfm drop\_dlm** command as described in this section to complete the overall task of dropping a Data Links Manager from a DB2 database.

### Prerequisites:

- The Data Links File Manager (DLFM) must be running on the required Data Links server.
- Ensure that you have deleted all DATALINK column values in the DB2 database that reference any files in the Data Links Manager that you want to remove.
- Issue the **DROP DATALINKS MANAGER** command from the DB2 database instance from which you want to remove the Data Links Manager, just before starting this procedure. See Related Reference at the end of this procedure for pointers to more information about the **DROP DATALINKS MANAGER** command.

### Procedure:

To remove a Data Links Manager from a DB2 database:

1. Issue the **dlfm drop\_dlm** command from the Data Links server machine that corresponds to the DB2 database instance from which you want to drop Data Links Manager.

**Attention:** The **dlfm drop\_dlm** command must only be used after dropping a DB2 Data Links Manager from a database with the **DROP DATALINKS MANAGER** command. Otherwise, important information about the DB2 Data Links Manager will be irrecoverably lost.

**Related reference:**

- “RECONCILE Command” in the *Command Reference*
- “DROP DATALINKS MANAGER Command” in the *Command Reference*
- “dlfm drop\_dlm command” on page 189

---

**Changing the diagnostic level for error message log files**

Data Links maintains error message log files from the Data Links server and from the DB2 databases that manage DATALINK data files. The primary error message log file is db2diag.log.

**On AIX systems or Solaris Operating Environments**, the db2diag.log file is located, by default, in the */INSTHOME/sqllib/db2dump* directory, where *INSTHOME* is the home directory of the instance owner.

**On Windows NT and Windows 2000 systems**, the db2diag.log file is located, by default, in the *x:\sqllib\instance* directory, where:

- *x*: is the drive where DB2 Data Links Manager is installed.
- *instance* is the name of the instance for which you want to change the diagnostic setting. The instance name in which Data Links Manager is running is DLFM.

The location of the db2diag.log file is controlled by the DB2 server configuration parameter *DIAGPATH*, so the directory paths on your system might be different from the default paths.

**Procedure:**

You control the level of detailed information that is written to the db2diag.log file by using the *DIAGLEVEL* configuration parameter and the *DLFM\_LOG\_LEVEL* registry value.

**DIAGLEVEL**

Determines the severity of DB2 diagnostic information recorded in the db2diag.log error log file. Valid values are from 1–4. 1 denotes that a minimal amount of information is to be recorded, and 4 denotes that the maximum amount of information is to be recorded. The default setting is 3. You can increase the amount of error information recorded using the following command: db2 update dbm cfg using *DIAGLEVEL* 4. This setting should be changed only at the request of IBM service or development for debugging purposes.

**DLFM\_LOG\_LEVEL**

Determines the severity of DLFM diagnostic information recorded in the db2diag.log error log file. Its default setting is *LOG\_ERR*. You can

increase the amount of error information recorded using the following command: `db2set DLFM_LOG_LEVEL=LOG_DEBUG`.

**Attention:** Increasing the amount of diagnostic output can result in both performance degradation and insufficient storage conditions in your database instance file system. This procedure should only be used when troubleshooting problems requiring the additional diagnostics.

**Related reference:**

- “Diagnostic Error Capture Level configuration parameter - diaglevel” in the *Administration Guide: Performance*
- “Diagnostic Data Directory Path configuration parameter - diagpath” in the *Administration Guide: Performance*

---

## Modifying logging for Data Links Filesystem Filter (DLFF) processing (AIX)

You can modify the logging for Data Links Filesystem Filter (DLFF) processing by changing the `d1fs_cfg` file. The `d1fs_cfg` contains configuration parameters that get input to the `strload` command, which is used to load the DLFF driver. The file is located in the `/usr/opt/db2_08_01/cfg` directory. Through a symbolic link, the file can also be found in the `/etc` directory.

**Procedure:**

The `d1fs_cfg` file has the following format:

```
d <driver-name> <vfs number> <dlfm id> <dlfm write group id>  
<global message priority> <global module priority> - 0 1
```

where:

**d** Specifies that the driver is to be loaded.

**driver-name**

The full path of the driver to be loaded. For instance, the full path for DB2 Version 8.1 is `/usr/opt/db2_08_01/bin/dlfsdrv`. The name of the driver is `dlfsdrv`.

**vfs number**

The `vfs` entry for DLFS in `/etc/vfs`.

**dlfm id**

The user ID of the Data Links Manager Administrator. By default, the *name* of this user is `d1fm`.

**dlfm write group id**

The ID of the special write group to which Data Links Manager Administrator belongs. By default, the *name* of this group is `d1fmxgrp`.

**global message priority**

A configurable parameter in the DLFS driver, which defines the list of driver routines, VFS operations and Vnode operations that will be logged to the system log file.

**global module priority**

A configurable parameter in the DLFS driver, which defines the list of the message categories that will be logged to the system log file.

**0 1** The minor numbers for creating non-clone nodes for this driver. The node names are created by appending the minor number to the cloned driver node name. No more than five minor numbers can be given (0-4).

Example:

```
d /usr/opt/db2_08_01/bin/dlfsdrv 14,208,210,255,-1 - 0 1
```

The messages that get logged depend on the settings for global message priority and global module priority. To minimize logging, you can change the value for the global message priority.

There are four message priority values you can use:

```
#define LOG_EMERGENCY    0x01
#define LOG_TRACING      0x02
#define LOG_ERROR        0x04
#define LOG_TROUBLESHOOT 0x08
```

Most of the messages in DLFF have LOG\_TROUBLESHOOT as the message priority. Here are a few alternative configuration examples.

If you require emergency messages and error messages, set the global message priority to 5 (1+4) in the dlfs\_cfg configuration file:

```
d /usr/opt/db2_08_01/bin/dlfsdrv 14,208,210,5,-1 - 0 1
```

If you only require error messages, set the global message priority to 4:

```
d /usr/opt/db2_08_01/bin/dlfsdrv 14,208,210,4,-1 - 0 1
```

If you do not require logging for DLFF, then set global message priority to 0:

```
d /usr/opt/db2_08_01/bin/dlfsdrv 14,208,210,0,-1 - 0 1
```

**Related tasks:**

- “Logging messages after the DLFF driver has been loaded (AIX)” on page 79

---

## Logging messages after the DLFF driver has been loaded (AIX)

If you need to log emergency, error, and troubleshooting messages after the DLFF driver has been loaded, you must modify the `dlfs_cfg` file with very specific values. The `dlfs_cfg` file is located in the `/usr/opt/db2_08_01/cfg` directory.

### Prerequisites:

- All Data Links File Systems (DLFSs) need to be unmounted.
- The Data Links File Manager (DLFM) must not be running.

### Procedure:

To log emergency, error, and troubleshooting messages after a DLFF driver has been loaded:

1. Unload the `dlfsdrv` kernel extension.
2. Change the global message priority of the `dlfs_cfg` file to be either of the following:
  - 255 (maximum priority)
  - 13 (8+4+1). Setting the priority to 13 (8+4+1) will log emergency, error, and troubleshooting information.
3. Reload the `dlfsdrv` driver to have the new priority values set at load time.
4. Remount all DLFS filter file systems.

**Important:** The settings for `dlfs_cfg` will remain for any subsequent loading of `dlfsdrv` driver until the `dlfs_cfg` file is again changed and reloaded.

### Related tasks:

- “Modifying logging for Data Links Filesystem Filter (DLFF) processing (AIX)” on page 77

---

## Minimizing logging for Data Links Filesystem Filter (DLFF) processing (Solaris Operating Environment)

You can minimize the logging for Data Links Filesystem Filter (DLFF) processing by changing the `/etc/syslog.conf` file. The `/etc/syslog.conf` contains information that the system log daemon, `syslogd`, uses to forward a system message to the appropriate log files.

### Procedure:

To reduce logging for DLFF processing:

1. Comment out the entries for `kern.notice` and `kern.debug` in the `/etc/syslog.conf` file.

2. Stop, then restart, the syslogd daemon.

If you ever need to log all kernel notices and errors again, remove the comments around the entries for `kern.notice` and `kern.debug` in the `/etc/syslog.conf` file. Stop, then restart, the syslogd daemon.

---

## Modifying logging for Data Links Filesystem Filter (DLFF) processing (Windows operating systems)

You can modify the logging for Data Links Filesystem Filter (DLFF) processing with the `dlff set loglevel` command. The `dlff set loglevel` command enables you customize the message severity level for all DLFS file logs. The messages get placed into the Windows system log.

### Prerequisites:

Check the current message severity level with the `dlff get loglevel` command.

### Procedure:

To modify the logging for Data Links Filesystem Filter (DLFF):

1. Open a Windows command prompt on the required DLFF-enabled drive.
2. Enter the command: `dlff set loglevel x`, where `x` is a number between 0 and 3 that sets the message severity level.
3. Run the `dlff refreshtrace` command for the changes to take effect.

---

## Migrating a DLFF-enabled file system to a different hard disk—overview

A file system that is enabled with the Data Links Filesystem Filter (DLFF) program can be migrated from one hard disk to another hard disk. Reasons to migrate a Data Links File System (DLFS) to another hard disk include:

- To move to a different hard disk on the same machine. For example, you might need to move one or more Data Links File Systems (DLFSs) to another, larger hard disk that already exists on the same machine.
- To replace the current hard disk. For example, you might need to replace the current hard disk with a new hard disk. All of the DLFSs will be moved to the new hard disk.

The *source* hard disk is the hard disk where the DLFSs currently reside. The *target* hard disk can either be an existing hard disk or a new hard disk.

You cannot move a DLFS from one DLFM server to another DLFM server.

The specific steps to migrate the DLFSs from one hard disk to another hard disk vary from operating system to operating system. However, the following basic tasks are necessary to migrate to a different hard disk:

1. On the file server, stop the Data Links File Manager (DLFM).
2. Log on with the user ID that has the necessary authorities to perform administrative tasks on your operating system.
3. Determine the file system ID value (UNIX systems) or share name (Windows systems) on the source drive where the DLFS resides.
4. Unmount the DLFS that you want to migrate (UNIX systems only).
5. Copy the DLFS contents from the source hard disk to the target hard disk.
  - If you are moving the DLFS to an existing hard disk on the same machine, prepare the location on the target hard disk to receive the data. Then copy the DLFS contents to the location on the target hard disk.
  - If you are replacing the source hard disk:
    - a. Copy the DLFS contents from the hard disk to tape.
    - b. Prepare the location on the target hard disk to receive the data.
    - c. Copy the DLFS contents from the tape to the location on the target hard disk.
6. Mount the new file system as Data Links enabled and check the file system ID value (UNIX systems only).
7. Update the drives listed under DLFF control and change the drive share names (Windows systems only).
8. Log on to the DLFM server as a user with Data Links Manager Administrator authority.
9. Start the Data Links File Manager (DLFM).
10. On each host database machine for each of the host databases, run the `db2_recon_aid` utility. This utility checks database tables and runs RECONCILE on tables that are potentially inconsistent with the DATALINK file data on the file server.

**Related tasks:**

- “Migrating a DLFS to a different hard disk (AIX)” on page 82
- “Migrating a DLFS to a different hard disk (Solaris Operating Environment)” on page 86
- “Migrating a DLFS to a different drive (Windows)” on page 89

---

## Migrating a DLFS to a different hard disk: details

How you move a Data Links File System (DLFS) to a different hard disk depends on the operating system on which the DLFS resides. The following sections describe the specific steps necessary to move a DLFS to a different hard disk for the AIX, Solaris Operating Environment, and Windows operating systems.

### Migrating a DLFS to a different hard disk (AIX)

On AIX, each Data Links File System (DLFS) is mapped to one logical volume. A logical volume can reside solely on a single hard disk, or span multiple hard disks. When you move a DLFS from one hard disk to another, you need to make sure that you move the entire logical volume.

You can move a DLFS to another hard disk that already exists on the same machine or to a hard disk that is replacing the current hard disk.

#### Prerequisites:

You must have root authority to move a DLFS.

#### Procedure:

This procedure assumes that you have a DLFS named `/dlfsfs` which is mapped to the logical volume `/dev/dlfslv`. Commands are entered from the operating system command prompt.

To move the `/dlfsfs` DLFS to a different hard disk, follow these steps:

1. On the DLFM server where the hard disk (or disks) that contain the DLFS currently reside, run the **dlfm stop** command to stop the Data Links File Manager (DLFM).
2. Log on to the DLFM server as a user with root authority.
3. Obtain the file system ID (fsid) value of the DLFS that is being moved.
  - a. Run the **ls** command to list the major and minor numbers of the logical volume that is mounted on `/dlfsfs`. Use the following format of the **ls** command:

```
ls -l /dev/dlfslv | awk '{print "major " $5,"minor " $6}'
```

where `/dev/dlfslv` is the logical volume corresponding to the file system `/dlfsfs`. The major and minor numbers will display on your screen.



- b. Convert the major and minor numbers to hexadecimal format. For example, suppose that the major number is 10 and the minor number is 9. The hexadecimal equivalent of 10 is 000a and the hexadecimal equivalent of 9 is 0009.
  - c. Append the hexadecimal equivalent for the minor number to the hexadecimal equivalent for the major number. This gives you the file system ID value. Using the above example, the file system ID value is 000a0009. This value is equal to 655369 in decimal format.
4. Use the **umount** command to unmount the DLFS that you want to move from the source hard disk. For example:  

```
umount /dlfsfs
```
  5. Copy the DLFS contents from the logical volume on the source hard disk to the target hard disk. The size of the new logical volume must be the same as or greater than the size of the original logical volume. The specific steps that you use to copy the contents depend on the status of the target hard disk:
    - If the target hard disk exists on the same machine as the source hard disk, run the **dd** command to copy the contents of the original logical volume to the new logical volume. For example, if the new logical volume is `/dev/newdlfslv`, the command syntax is:  

```
/usr/bin/dd if=/dev/dlfslv of=/dev/newdlfslv
```

**Restriction:** It is not possible for the new logical volume to have the same file system ID value as the original logical volume when both the source and target hard drives are connected to the same machine.

- If the target hard disk is replacing the source hard disk:
  - a. Copy the contents of the original logical volume to tape. Using the **dd** command, the syntax is:  

```
/usr/bin/dd if=/dev/dlfslv of=/dev/rmt0 bs=512b
```
  - b. Replace the source hard disk with the target hard disk.
  - c. Configure the target hard disk using the standard configuration.
  - d. Create a new logical volume on the target hard disk. Assume for this example that the new volume is `/dev/newdlfslv`.

**Recommendation:** If the major number and minor number of the new logical volume are same as those numbers of the original logical volume, migration time will be reduced significantly. You can create the new volume group using the **mkvg** command with the **-V** option. Then create a logical volume in this volume group. There is no option with which you can create a logical volume with a predefined minor number. To do so you have to create the logical volumes in the order that they appeared in the old volume group. For more information about the **mkvg** and **mklv** commands, see the AIX Command Reference.

- e. Copy the contents of the original logical volume from tape to the new logical volume on the target hard disk. For example:

```
/usr/bin/dd if=/dev/rmt0 of=/dev/newdlfs1v bs=512b
```

6. Specify the file system mount point that you want to use.

- If you want to keep the same file system mount point, change the file system entry for the DLFS in the `/etc/filesystems` file on the file server. Change the value of device name from the original logical volume name to the new logical volume name.
- If you want to change the file system mount point and use a different file system mount point, remove the original file system entry from the `/etc/filesystems` file on the file server. Then change the prefix name entries in the DLFM\_DB database.

- a. Log on to the DLFM server as a user with Data Links Manager Administrator authority.

- b. Run the `dlfm_migrate_prefix` utility on the DLFM file server for each of the migrated file systems. For example, if `/dlfsfs` is the name of the original DLFS and `/dlfsfsnew` is the name of the new DLFS, the syntax for the `dlfm_migrate_prefix` utility is:

```
dlfm_migrate_prefix /dlfsfs /dlfsfsnew amit
```

where *amit* is the DLFM database instance owner ID. The default DLFM database instance owner ID at Data Links Manager installation time is `dlfm`.

To view the parameter syntax for the `dlfm_migrate_prefix` utility, run the utility without any parameters. For example:

```
dlfm_migrate_prefix
```

- c. Log on to the DLFM server as a user with root authority.
- d. Run the **dlfmfsmd** script to enable the new file system. For example:

```
/usr/opt/db2_08_01/instance/dlfmfsmd dlfm_mountpoint
```

where *dlfm\_mountpoint* is the file system mount point of the new DLFS, for example `/dlfsfsnew`.

7. If the file system is not already mounted, mount the file system as Data Links Manager enabled. For example, if you have changed the mount point to `/dlfsfsnew`, the syntax is:

```
mount -v dlfs /dlfsfsnew
```

8. Check the file system ID (`fsid`). If the new logical volume has a different major and minor number than the original logical volume, the `fsid` will be different. Refer back to steps 3 and 5.

- If your new logical volume has the same major and minor number as the original logical volume, go to step 9

- If your new logical volume has a different major and minor number than the original logical volume, update the file system ID (fsid) entries in the DLFM\_DB database.
  - a. Log on to the DLFM server as a user with Data Links Manager Administrator authority.
  - b. Run the `dlfm_migrate_fsid` utility on the DLFM file server for each of the migrated file systems.

To view the parameter syntax for the `dlfm_migrate_fsid` utility, run the utility without any parameters. For example, if `/dlfsfsnew` is the name of the new DLFS, the syntax for the `dlfm_migrate_fsid` utility is:

```
dlfm_migrate_fsid /dlfsfsnew amit
```

where *amit* is the DLFM database instance owner ID. The default DLFM database instance owner ID at Data Links Manager installation time is `dlfm`.

9. Run the **`dlfm start`** command to start the Data Links File Manager (DLFM).

After you complete the steps on the DLFM server, run the `db2_recon_aid` utility on each host database machine for each of the host databases. The `db2_recon_aid` utility provides a mechanism for checking database tables and running RECONCILE on tables that are potentially inconsistent with the DATALINK file data on the file server.

On AIX systems, the `db2_recon_aid` utility is located in `INSTHOME/sql1lib/adm` directory, where `INSTHOME` is the home directory of the instance owner. An example of the `db2_recon_aid` syntax is:

```
db2_recon_aid -db abc
              -selective
              -server udbnew.in.ibm.com
              -reportdir /home/amit/changedsk/reports
              -prefixes /dlfsfs1:/dlfsfs2
```

To view the parameter syntax for the `db2_recon_aid` utility, run the utility without any parameters. For example:

```
db2_recon_aid
```

#### **Related tasks:**

- “Enabling and registering file systems with DLFM (AIX, Solaris Operating Environment)” on page 65

#### **Related reference:**

- “`db2_recon_aid` - RECONCILE Multiple Tables Command” in the *Command Reference*

- “db2\_recon\_aid utility” on page 146

## Migrating a DLFS to a different hard disk (Solaris Operating Environment)

In the Solaris Operating Environment, each Data Links File System (DLFS) is mapped to a hard disk slice. You can move a DLFS to another hard disk that already exists on the same machine or to a hard disk that is replacing the current hard disk.

### Prerequisites:

You must have root authority to move a DLFS.

### Procedure:

This procedure assumes that you have a DLFS named `/dlfsfs` that is mapped to the hard disk slice `/dev/dsk/c0t0d0s5`. Commands are entered from the operating system command prompt.

To move the `/dlfsfs` DLFS to a different hard disk, follow these steps:

1. On the DLFM server where the hard disk containing the DLFS currently reside, run the **dlfm stop** command to stop the Data Links File Manager (DLFM).
2. Log on to the DLFM server as a user with root authority.
3. Obtain the file system ID (fsid) value of the DLFS that is being moved by issuing the **df** command. Use the following format of the **df** command:  

```
df -g /dlfsfs | grep fileysys | awk {'print "fsid: "$4'}
```

where `/dlfsfs` is file system corresponding to the hard disk slice `/dev/dsk/c0t0d0s5`

The file system ID (fsid) value will displayed on your screen. An example of an fsid value is: 8388671

4. Use the **umount** command to unmount the DLFS that you want to move from the source hard disk. For example:  

```
umount /dlfsfs
```
5. Copy the DLFS contents from the hard disk slice on the source hard disk to the target hard disk. The size of the new hard disk slice must be the same as or greater than the size of the original hard disk slice. The specific steps that you use to copy the contents depend on the status of the target hard disk:
  - If the target hard disk exists on the same machine as the source hard disk, run the **dd** command to copy the contents of the original hard disk slice to the new hard disk slice. For example, if the new hard disk slice is `/dev/dsk/c0t8d0s5`, the command syntax is:

```
/usr/bin/dd if=/dev/dsk/c0t0d0s5 of=/dev/dsk/c0t4d0s5
```

**Restriction:** It is not possible for the new hard disk slice to have the same file system ID value as the original hard disk slice when both the source and target hard drives are connected to the same machine.

- If the target hard disk is replacing the source hard disk:
  - a. Copy the contents of the original hard disk slice to tape. Using the **dd** command, the syntax is:

```
/usr/bin/dd if=/dev/dsk/c0t0d0s5 of=/dev/rmt0 bs=512b
```

- b. Replace the source hard disk with the target hard disk.
- c. Configure the target hard disk using the standard configuration.
- d. Copy the contents of the original hard disk slice from tape to the new hard disk slice on the target hard disk. For example:

```
/usr/bin/dd if=/dev/rmt0 of=/dev/dsk/c0t4d0s5 bs=512b
```

6. Specify the file system mount point that you want to use.
  - If you want to keep the same file system mount point, change the file system entry for the DLFS in the `/etc/vfstab` file on the file server. Change the value of the device name from the original disk slice name to the new disk slice name.
  - If you want to change the file system mount point and use a different file system mount point, remove the original file system entry from the `/etc/vfstab` file on the file server. Then change the prefix name entries in the DLFM\_DB database.
    - a. Log on to the DLFM server as a user with Data Links Manager Administrator authority.
    - b. Run the `dlfm_migrate_prefix` utility on the DLFM file server for each of the migrated file systems. For example, if `/dlfsfs` is the name of the original DLFS and `/dlfsfsnew` is the name of the new DLFS, the syntax for the `dlfm_migrate_prefix` utility is:

```
dlfm_migrate_prefix /dlfsfs /dlfsfsnew amit
```

where *amit* is the DLFM database instance owner ID. The default DLFM database instance owner ID at Data Links Manager installation time is `dlfm`.

To view the parameter syntax for the `dlfm_migrate_prefix` utility, run the utility without any parameters. For example:

```
dlfm_migrate_prefix
```

- c. Log on to the DLFM server as a user with root authority.
- d. Run the **dlfmfsmd** script to enable the new file system. For example:

```
/opt/IBM/db2/V8.1/int/instance/dlfmfsmd dlfm_mountpoint
```

where *dlfm\_mountpoint* is the file system mount point of the new DLFS, for example */dlfsfsnew*.

7. If the file system is not already mounted, mount the file system as Data Links Manager enabled. For example, if you changed the mount point to */dlfsfsnew*, the syntax is:

```
mount /dlfsfsnew
```

8. Check the file system ID (*fsid*). Refer back to step 3 for details on how to check the *fsid*.
  - If your new disk slice has the same *fsid* as the original disk slice, go to step 9.
  - If your new disk slice has a different *fsid* than the original disk slice, update the *fsid* entries in the *DLFM\_DB* database.
    - a. Log on to the *DLFM* server as a user with Data Links Manager Administrator authority.
    - b. Run the *dlfm\_migrate\_fsid* utility on the *DLFM* file server for each of the migrated file systems.

To view the parameter syntax for the *dlfm\_migrate\_fsid* utility, run the utility without any parameters. For example, if */dlfsfsnew* is the name of the new DLFS, the syntax for the *dlfm\_migrate\_fsid* utility is:

```
dlfm_migrate_fsid /dlfsfsnew amit
```

where *amit* is the *DLFM* database instance owner ID. The default *DLFM* database instance owner ID at Data Links Manager installation time is *dlfm*.

9. Run the ***dlfm start*** command to start the Data Links File Manager (*DLFM*).

After you complete the steps on the *DLFM* server, run the *db2\_recon\_aid* utility on each host database machine for each of the host databases. The *db2\_recon\_aid* utility provides a mechanism for checking database tables and running *RECONCILE* on tables that are potentially inconsistent with the *DATALINK* file data on the file server.

On Solaris Operating Environments, the *db2\_recon\_aid* utility is located in *INSTHOME/sql/lib/adm* directory, where *INSTHOME* is the home directory of the instance owner. An example of the *db2\_recon\_aid* syntax is:

```
db2_recon_aid -db abc
-selective
-server udbnew.in.ibm.com
-reportdir /home/amit/changedsk/reports
-prefixes /dlfsfs1:/dlfsfs2
```

To view the parameter syntax for the `db2_recon_aid` utility, run the utility without any parameters. For example:

```
db2_recon_aid
```

**Related tasks:**

- “Enabling and registering file systems with DLFM (AIX, Solaris Operating Environment)” on page 65

**Related reference:**

- “`db2_recon_aid - RECONCILE Multiple Tables Command`” in the *Command Reference*
- “`db2_recon_aid` utility” on page 146

## Migrating a DLFS to a different drive (Windows)

On Windows, you can move a DLFS to another drive that already exists on the same machine or to a new disk drive that is replacing the current disk drive.

**Prerequisites:**

You must have Windows Administrator authority to move a DLFS.

**Procedure:**

This procedure assumes that you have a DLFS residing on the `D:\` disk drive and that the share name of this disk drive is `dl_drive`. Commands are entered from the operating system command prompt.

To move the DLFS to a different hard disk, follow these steps:

1. On the DLFM server where the disk drive containing the DLFS currently resides, use the **`dlfm stop`** command to stop the Data Links File Manager (DLFM).
2. Log on to the DLFM server using the Administrator user ID.
3. Obtain the share name of the source disk drive where the DLFS resides by following these steps:
  - a. Use the **`dlff list`** command to list all the logical drives under DLFF control, for example:

```
x:\PROGRA~1\IBM\SQLLIB\BIN>dlff list
```

where `x:` is the drive where the `\SQLLIB\BIN` directory is installed. The output from this command is similar to the following example:

```
LogicalDrives = d:;m:
```

- b. Use the **net share** command to list the resources being shared, for example

```
x:\PROGRAM~1\IBM\SQLLIB\BIN>net share
```

where *x*: is the drive where the \SQLLIB\BIN directory is installed.

4. Copy the DLFS contents from the source disk drive to the target disk drive. The size of the new disk drive must be the same as or greater than the size of the original disk drive. The share name of the target disk drive needs to be changed to the same share name that was used by the original disk drive. The specific steps that you use to copy the contents depend on the status of the target disk drive:

- If the target disk drive exists on the same machine as the source disk drive, use the **SCOPY** command to copy the contents of the original disk drive to the new disk drive. For example, to copy the DLFS from the D:\ drive to the E:\ drive, the command syntax is:

```
SCOPY D:\ E:\ /o /a /s
```

**Note:** On Windows 2000, you can use the XCOPY command instead. The command syntax is:

```
XCOPY D:\ E:\ /x /o /s
```

- If the target disk drive is replacing the source disk drive.
  - a. Copy the contents of the original disk drive to tape using the Backup utility. On Windows NT, you access this utility under **Administrative Tools**. On Windows 2000, you access this utility under **Accessories—>System Tools**.
  - b. Replace the source disk drive with the target disk drive.
  - c. Configure the target disk drive using the standard configuration.
  - d. Create and format a logical drive on the new disk drive.
  - e. Restore the contents of the DLFS that you copied to tape on to the new drive using the Backup utility under **Administrative Tools**.

5. Remove the original drive from the list of drives under the control of DLFF. For example:

```
DLFF remove D:
```

6. Add the new drive name to the list of drives under the control of DLFF. For example, if the new drive is E:\ the command is:

```
DLFF add E:
```

7. Change the share name of the original disk drive to something else. For example, suppose the original disk drive is D:\ and share name is **dl\_drive**. To change the share name to **d\_drive**, follow these steps:

- a. Delete the original share name on the D:\ drive, for example:

```
net share dl_drive /DELETE
```

- b. Assign the new share name **d\_drive** to the D:\ drive, for example



```
net share d_drive=d:
```

8. Assign the new disk drive the share name that you used on the original disk drive. For example, suppose the original share name is **dl\_drive** and the new disk drive is E:\. To assign the share name **dl\_drive** to the E:\ disk drive:

```
net share dl_drive=e:
```

9. Reboot the machine and log on using the Data Links Manager Administrator user ID. Start the Data Links File Manager (DLFM).

```
dlfm start
```

After you complete the steps on the DLFM server, run the `db2_recon_aid` utility on each host database machine for each of the host databases. The `db2_recon_aid` utility provides a mechanism for checking database tables and running RECONCILE on tables that are potentially inconsistent with the DATALINK file data on the file server.

On Windows systems, the `db2_recon_aid` utility is located in the `x:\sql11ib\bin` directory, where `x:` is the drive where you installed DB2 Data Links Manager. If the DB2 server is also on Windows, an example of the `db2_recon_aid` syntax is:

```
db2_recon_aid -db mydb
               -selective
               -server dlserver.in.ibm.com
               -reportdir c:\reports
               -prefixes \dl_drive1:\dl_drive2
```

To view the parameter syntax for the `db2_recon_aid` utility, run the utility without any parameters. For example:

```
db2_recon_aid
```

#### **Related concepts:**

- “SCOPY versus COPY” on page 92

#### **Related reference:**

- “`db2_recon_aid` - RECONCILE Multiple Tables Command” in the *Command Reference*
- “`db2_recon_aid` utility” on page 146

## **Using SCOPY instead of COPY**

This section describes the Windows copy utilities you can use when migrating a DLFS to a different disk.

## **SCOPY versus COPY**

Windows<sup>®</sup> provides several utilities for copying files. For the DLFS to function properly after it is migrated to another disk drive, the DLFS security information must remain intact.

**Restriction:** If the drive being copied contains directories that have the user permission set to SYSTEM, then the copy operation will fail for those directories. You must add ADMINISTRATOR authority to the user permissions for those directories to ensure that the copy operation is successful.

On Windows NT, use the SCOPY utility. This utility copies files and directories to and from NTFS partitions with the security access control lists (ACLs) intact. On Windows NT, the COPY and XCOPY utilities do not transfer the security information inherent within NTFS. The Windows NT<sup>®</sup> Resource Kit includes the SCOPY utility.

The SCOPY utility provides several parameters that enable you to copy the owner security information, the auditing information, and the all the files in subdirectories. You must have Backup and Restore files User Rights on both the source and destination disk drives to use the SCOPY utility. An example of the SCOPY syntax is:

```
SCOPY <source> <destination> /o /a /s
```

where:

**<source>**

Specifies the files to copy

**<destination>**

Specifies where to copy the files

**/o** copies the owner security information

**/a** Copies auditing information. This requires that you have the Manage Auditing User Right on both the source and destination disk drives.

**/s** Copies all files in subdirectories

On Windows 2000, you can use the SCOPY utility or the XCOPY utility. To ensure the DLFS security information remains intact with the XCOPY utility, the proper syntax is:

```
XCOPY <source> <destination> /o /x /s
```

where:

**<source>**

Specifies the files to copy

**<destination>**

Specifies where to copy the files

**/o** copies the ACL information

**/x** Copies auditing information. This requires that you have the Manage Auditing User Right on both the source and destination disk drives.

**/s** Copies all files in subdirectories

**Related tasks:**

- “Migrating a DLFS to a different drive (Windows)” on page 89

---

## Retrieving archive server backup file information

This procedure describes how to retrieve a list of files that have been backed up to the archive server, and the status of each of those files.

**Prerequisites:**

You must have DB2 Data Links Manager Administrator authority.

**Procedure:**

To retrieve a list of files that have been backed up to the archive server:

1. Log on to the system as the DB2 Data Links Manager Administrator.
2. Enter the **dlfm retrieve** command using the following syntax:

```
dlfm retrieve -o output_file -h hostname -d database_name -i instance_name -p registered_prefix
```

where:

- *output\_file* is the name of the file to which you want to write the command output. If not specified, output goes to the screen.
- *hostname* is the hostname of the DB2 host server.
- *database\_name* is the name of the database that contains the DATALINK column references to files that were backed up to the archive server.
- *instance\_name* is the name of the instance where the specified database resides. The *instance\_name* value is case sensitive.
- *registered\_prefix* is the path of the file system that was registered using the **dlfm add\_prefix** command.

If you enter the **dlfm retrieve** command without any parameters, the system prompts you for these parameters. This command, entered without parameters, produces output similar to the following:

Using default datalinks server database dl\_fm\_db.

Please make your choice of hosts registered with DLFM.

0 ARROW.TOROLAB.IBM.COM

Enter the number

Please make your choice of the database/instance.

0 TSTDB001 regress ARROW.TOROLAB.IBM.COM

1 TSTDB002 regress ARROW.TOROLAB.IBM.COM

2 TSTDB003 regress ARROW.TOROLAB.IBM.COM

3 TSTDB004 regress ARROW.TOROLAB.IBM.COM

4 TSTDB005 regress ARROW.TOROLAB.IBM.COM

Enter the number

Please make your choice of the prefix Name.

0 \dlfstest\

Enter the number

RETRIEVE QUERY OUTPUT

The following files were backed up from database TSTDB001, on host  
ARROW.TOROLAB.IBM.COM from the instance regress

Copy Status	Link Status	Operation time	File Name
E1	L	2000-06-03-13.26.49.586476	\dlfstest\fileA1
E1	L	2000-06-03-13.26.50.243762	\dlfstest\fileA2
E1	L	2000-06-03-13.25.55.345240	\dlfstest\fileA3
E1	L	2000-06-03-13.27.03.034247	\dlfstest\fileA31
E1	L	2000-06-03-13.27.03.937676	\dlfstest\fileA32
E1	L	2000-06-03-13.25.56.176132	\dlfstest\fileA4
E1	L	2000-06-03-13.25.56.961493	\dlfstest\fileA5
E1	L	2000-06-03-13.25.58.424379	\dlfstest\fileB1
E1	L	2000-06-03-13.25.59.126102	\dlfstest\fileB2
E1	L	2000-06-03-13.26.51.973211	\dlfstest\fileB3
E1	L	2000-06-03-13.26.52.623260	\dlfstest\fileB4
E1	L	2000-06-03-13.26.53.278827	\dlfstest\fileB5

Legend:

L - Linked

U - Unlinked

G - File to be garbage collected

E1 - Marked Copied and in backup

E2 - Marked Copied and not in backup

E3 - Marked To be Copied and not in backup

E4 - Marked To be copied but in backup

\*\*\*\*\*

### Related reference:

- “dl\_fm retrieve command” on page 202

---

## Chapter 4. Security

This chapter describes the Data Links Manager security features and how to implement them.

---

### Introduction to Data Links Manager security

Data security is an important benefit of using DB2® Data Links Manager. Data Links Manager has both built-in and customizable security features that enable you to secure access to your linked files.

**Important:** A superuser can perform almost any action, including all file operations, in a Data Links File System (DLFS), regardless of the Data Links Manager data access security features that you use. In AIX® and Solaris operating environments, root is a superuser. In Windows® NT and Windows 2000 operating environments, the dlmadmin account is a superuser. Data Links Manager Administrators should be cautious about the actions they take when logged in under a superuser id. It is possible to bypass various data integrity controls that are crucial to the Data Links Manager environment and normal operations, particularly on the DLFS. However, under some circumstances, even a superuser cannot perform certain functions (for example, creating a directory if the DLFM is not up and running, or attempting to perform an action that can severely compromise linked file referential integrity, like renaming a directory).

#### Related concepts:

- “Built-in security features” on page 95
- “Data access security features” on page 96
- “Basic linked file security controls” on page 97
- “Advanced file management security features” on page 98
- “Advanced file management security rules and guidelines” on page 101
- “Read operation security” on page 105
- “Write operation security” on page 105
- “Security issues for updating linked files” on page 107

---

### Built-in security features

The following security features are enabled for all installations. You cannot modify or disable these features.

### **Data Links File Manager (DLFM) application security**

The DLFM application uses a DB2<sup>®</sup> database to store all information about the linked files on a Data Links server. Using DB2 enables DLFM to use the DB2 built-in security and data integrity features. For example, the communications between a DLFM and its DB2 database are secured by the DB2 client authentication processes.

### **Data Links File Filter (DLFF) security**

DLFF is loaded as an operating system kernel extension, which enables the operating system to secure the controls that DLFF performs.

### **DLFM and DB2 server database communications security**

Any DB2 database on a DB2 server that wants to use Data Links Manager must register with a DLFM on a Data Links server.

The database registration process involves: 1) Invoking the **dlfm add\_db** command, with the required database and instance names. 2) Invoking the **ADD DATALINKS MANAGER** command on the DB2 server. Database registration typically need only be done once. However, if you drop a database that is already registered with DLFM, then recreate that exact same database again later, you must reregister the database. The **DB2 DROP DATABASE** command automatically invalidates a database's registration as an added security precaution.

You must also register all Data Links servers and the DLFM service port number with each DB2 server database. The registration process consists of invoking the DB2 **ADD DATALINKS MANAGER** command.

Each time a DB2 database-to-DLFM socket connection gets initiated, DLFM ensures that the requesting DB2 database has been registered. If the requesting DB2 database is not registered, then DLFM rejects the socket connection and generates an SQL error message.

### **DLFM and DLFF communications security**

The communications between the DLFM and the Data Links File Filter (DLFF) are secured with a private messaging mechanism.

### **Related concepts:**

- "Introduction to Data Links Manager security" on page 95
- "Data access security features" on page 96

---

## **Data access security features**

The Data Links Manager data access security features range from simple to elaborate. All of the features can be customized. Compare and contrast each feature, then decide which ones will work best for you based on your organization's goals and requirements.

**Important:** A superuser can perform almost any action, including all file operations, in a Data Links File System (DLFS), regardless of the Data Links Manager data access security features that you use. In AIX<sup>®</sup> and Solaris operating environments, root is a superuser. In Windows<sup>®</sup> NT and Windows 2000 operating environments, the dlmadmin account is a superuser. Data Links Manager Administrators should be cautious about the actions they take when logged in under a superuser id. It is possible to bypass various data integrity controls that are crucial to the Data Links Manager environment and normal operations, particularly on the DLFS. However, under some circumstances, even a superuser cannot perform certain functions (for example, creating a directory if the DLFM is not up and running, or attempting to perform an action that can severely compromise linked file referential integrity, like renaming a directory).

**Important:** A DB2<sup>®</sup> server's DATALINK column access permissions and database configuration have a major impact on Data Links File System (DLFS) operations and, by inference, also impact any data access security features that you implement. For example, suppose you have a DATALINK table column that is defined with the READ PERMISSION attribute set to DB. You assign the SELECT privilege to a user for that particular table. This particular user has local or remote access to the given DLFS file system. In this situation, the user can read all linked files in that column on the corresponding Data Links servers, regardless of how you have configured data access security on those Data Links servers.

**Related concepts:**

- "Basic linked file security controls" on page 97
- "Advanced file management security features" on page 98

---

## Data access security features: details

The following sections describe the data access security features in detail.

### Basic linked file security controls

When you link a file, Data Links Manager controls most of the operations that can be performed on the file. One generally prohibited operation is a deletion. No one, with the exception of a superuser, can **ever** delete or rename a linked file. All controls stay in place for as long as a file remains linked. Once a file is unlinked, operational control reverts to the native file system.

Table 2 on page 98 provides a complete list of the operations that Data Links File Manager prohibits for linked files. These operations are prohibited for all users *except* superusers.

Table 2. Prohibited linked file operations

DATALINK column access permission attributes <sup>1</sup>	Prohibited linked file operations
<ul style="list-style-type: none"> <li>• READ PERMISSION DB</li> <li>• WRITE PERMISSION BLOCKED or WRITE PERMISSION ADMIN</li> </ul>	Read/delete/rename/write and set file attributes. <sup>2</sup> <ul style="list-style-type: none"> <li>• The read operation is prohibited <i>only</i> if a valid access token is not specified.</li> <li>• The write operation is permitted <i>only</i> if a DATALINK column is defined with WRITE PERMISSION ADMIN and a valid write token is specified.</li> </ul>
<ul style="list-style-type: none"> <li>• READ PERMISSION FS</li> <li>• WRITE PERMISSION BLOCKED</li> </ul>	Delete/rename files and reset the R/O flag attribute.
<ul style="list-style-type: none"> <li>• READ PERMISSION FS</li> <li>• WRITE PERMISSION FS</li> </ul>	Delete/rename files.
<p><b>Notes:</b></p> <ol style="list-style-type: none"> <li>1. The DATALINK column must also be defined with the FILE LINK CONTROL attribute.</li> <li>2. For Windows<sup>®</sup> only: <i>Any</i> Administrator user account, not just the dladmin account, can view a linked file's permissions without using a read or write token. However, only the dladmin user account can read files.</li> </ol>	

Depending upon how a particular table's DATALINK column access permissions are defined, Data Links Manager enables you to further control who can perform link, read, write, and replication operations to the files stored on a Data Links server.

**Related concepts:**

- "Advanced file management security features" on page 98

**Related reference:**

- "dlfm grant replication read command" on page 191
- "dlfm grant replication write command" on page 193
- "dlfm grant command" on page 189
- "dlfm list registered replication access control command" on page 196
- "dlfm list registered directories command" on page 194
- "dlfm set link security command" on page 206

**Advanced file management security features**

Data Links Manager provides customizable security features for restricting link and write operations for the files that are stored on a Data Links server.



These features enable you to authorize who can link to *any* file, and who can write to the *linked* files that reside on a particular Data Links server. Anyone who is not explicitly authorized to perform a file link operation, or to write to a linked file, is *prohibited* from performing that operation. These security features work in conjunction with:

- Any access and operational controls as defined and enforced by the native file system on a Data Links server.
- The write access permissions of a DB2<sup>®</sup> host table DATALINK column.
- The Data Links Manager basic linked file security controls.

You configure the advanced file management security controls with a series of **dlfm** commands, which you invoke from the Data Links server where you want to implement the controls.

- The **dlfm grant** command enables you to assign link and write privileges to individual DB2 server users, or to groups of DB2 server users, on specific Data Links server directories.
- The **dlfm deny** command enables you to prohibit link and write privileges to individual DB2 server users, or to groups of DB2 server users, on specific Data Links server directories.
- The **dlfm revoke** command enables you to reverse the effects of the **dlfm grant** or **dlfm deny** commands.

Table 3 shows all the possible write permission attributes of a DB2 host table DATALINK column, and compares the privileges that you can grant, deny, and revoke based on those attributes.

*Table 3. DATALINK column attributes and applicable DLM security privileges*

	Link privilege?	Write privilege?	Both Link and Write privileges?
WRITE PERMISSION ADMIN	Yes	Yes	Yes
WRITE PERMISSION FS	Yes	No	No
WRITE PERMISSION BLOCKED	Yes	No	No

Note that you can only control the write privilege when a DB2 host table DATALINK column is set to WRITE PERMISSION ADMIN.

The file link security control feature is turned on (activated) by default at installation time. No one, not even a superuser, can link to the files on a Data

Links server until the Data Links Manager Administrator explicitly grants the link privilege. You can grant users the link privilege as soon as you have done the following:

- Registered the Data Links server and the DLFM service port number with each DB2 server database.
- Registered the prefixes on the Data Links server.

**Important:** If you do not require user- or group-level controls (including the group PUBLIC) on the link privilege for a particular Data Links server, or if you operate in a secure network environment where such controls are not needed, you can deactivate file link security control feature completely, thereby granting the link privilege to all users. To deactivate file link security control, invoke the **dlfm set link security** command exactly as shown:

```
dlfm set link security off
```

If you have used the above command to only temporarily disable the file link security control feature (for example, to use Data Links Manager in a test environment), you can re-enable all previously defined file link security controls by invoking the **dlfm set link security** command again, exactly as shown:

```
dlfm set link security on
```

Effectively disabling the file link security control feature can slightly improve performance, but should **only** be done if you are certain that operating in such an "open" environment is in compliance with your company's security requirements.

**Important:** All file link and file write security access privileges are stored on the Data Links server. If you deactivate the file link file security control feature, then reactivate it at a later time, all file link security access privileges that you had previously defined also get reactivated.

**Related concepts:**

- "Basic linked file security controls" on page 97
- "Advanced file management security rules and guidelines" on page 101

**Related reference:**

- "dlfm deny command" on page 186
- "dlfm grant command" on page 189
- "dlfm revoke command" on page 203
- "dlfm set link security command" on page 206

## Advanced file management security controls: detail

The following section provides more information about the file link security feature.

### Advanced file management security rules and guidelines

You use the **dlfm grant** command to assign link and write privileges to individual DB2® server users, or to groups of DB2 server users, on specific Data Links server directories. The **dlfm grant** command has the following syntax:

```
►—dlfm grant—write privilege—on dir—directory_name—to—user—db2_authorization_id—►
      |link privilege|
      |all privileges|
      |group|
      |public|
►—for db—db_name—inst—db_inst_name—node—server_node_name—►
```

**Attention:** The privileges granted apply to all files in the specified directory and its subdirectories.

The specified DB2 server user or users can link or write to the specified files only when they are operating on the exact database, instance, and node as specified in the **dlfm grant** command. For example, suppose you issue the following command on a particular Data Links server:

```
dlfm grant link privilege on dir /dlfs/test to user SHERRYG for
db EMP_TEST inst HRaccess node olympus.sanjose.bigco.com
```

The result is that DB2 server user SHERRYG is allowed to link files in the /dlfs/test directory only when SHERRYG is linking those files to DB2 server database EMP\_TEST under the HRaccess instance on the olympus.sanjose.bigco.com node.

When any user attempts an unauthorized file link action, an audit record gets written in the DB2 host sqllib/db2dump/db2diag.log file. Also, when an unauthorized user tries to link to a file during an SQL INSERT or UPDATE operation, an SQL error will be generated.

To see the users, their access privileges, and the specific Data Links server directories to which each user has access, invoke the **dlfm list registered users** or the **dlfm list registered directories** command.

- To list all users and their access privileges on a specific directory on a Data Links server, use the **dlfm list registered users** command.

#### Example:

Issuing this command:

```
dlfm list registered users for directory /localfs/dbfiles/photos on db
employee inst acct1 node storage.ca2.bigco.com
```

would produce output similar to the following:

TYPE	AUTHID	PRIVILEGE
user	JDOE	write
user	MOHAN	link

- To list all Data Links server directories to which all or a specific group of users has access, and their access privileges, use the **dlfm list registered directories** command.

**Example:**

Issuing this command:

```
dlfm list registered directories for all users on db drawings inst acct2
node storage.ak1.bigco.com
```

might provide output similar to the following:

TYPE	AUTHID	PRIVILEGE	DIRECTORY
user	appl	write	/localfs/files/chips/
user	srgordon	link	/localfs/files/widgets/

You use the **dlfm deny** command to deny a specific privilege to a DB2 server user or group. The **dlfm deny** command syntax is similar to that of the **dlfm grant** command, and all of the same conditions apply.

The **dlfm grant** and **dlfm deny** commands define a set of rules regarding a specific privilege (the authority to link a file or to write to a file) to the DLFM on a specific Data Links server. The rules are a grouping of the following attributes:

authorization-type , directory , user-set , specific-database

- authorization-type is either a grant or deny
- directory is an absolute path, or an "\*" to indicate all directories under all defined prefixes
- user-set is an individual user, group, or all users (PUBLIC)
- specific-database is the database name, instance, and node indicating the specific DB2 server database that is affected by this rule

You might define a deny rule instead of a grant rule for ease of administration, because in some situations it might be easier to specify what is *not* allowed than what *is* allowed.

**Example 1:**

Suppose you want to give file link authority to all users under the directory /dlfs/test *except* the specific directory /dlfs/test/restricted, and there are a number of directories under /dlfs/test.

You could grant file link authority to all users for each and every subdirectory under `/dlfs/test`, but that would require issuing a separate **dlfm grant** command for each subdirectory. Also, if another subdirectory gets added at a later time, issuing another **dlfm grant** command would be required to give everyone link authority under that new subdirectory.

A simpler approach would be to define two authorization rules: one that grants all users file link authority to the `/dlfs/test` directory, and one that denies all users file link authority to the `/dlfs/test/restricted` subdirectory. This approach requires issuing only two **dlfm** commands, regardless of the number of subdirectories that might get added to `/dlfs/test` at a later time.

- First, give all users file link authority under `/dlfs/test` with the following command: `dlfm grant link privilege on dir /dlfs/test to PUBLIC for db EMP_TEST inst HRaccess node olympus.sanjose.bigco.com`
- Then, deny everyone access to the specific directory `/dlfs/test/restricted/` with the following command: `dlfm deny link privilege on dir /dlfs/test/restricted to PUBLIC for db EMP_TEST inst node olympus.sanjose.bigco.com`

Running these commands results in defining the following two rules to the DLFM:

```
grant , /dlfs/test          , PUBLIC , {EMP_TEST;HRaccess;olympus.sanjose.bigco.com}
deny , /dlfs/test/restricted, PUBLIC ,{EMP_TEST;HRaccess;olympus.sanjose.bigco.com}
```

It is important to understand the way in which the DLFM interprets these rules when deciding whether or not a given action is authorized. The DLFM first examines all applicable deny rules, *then* examines all applicable grant rules. If the requested action was not prohibited by a deny rule *and* was explicitly authorized by a grant rule, then the action is allowed. You must have at least one grant rule in order for anyone to be authorized to perform some action.

The purpose of the **dlfm revoke** command is to remove a rule previously defined with a **dlfm grant** or a **dlfm deny** command. There is some overlap in the effects of the **dlfm revoke** and **dlfm deny** commands, as their names imply. Both commands can be used to restrict a granted privilege.

### Example 2:

Suppose a user has been granted the link privilege for a specific directory, `/dlfs/test/APP1`. There are two ways you could remove that privilege:

- Define a deny rule on this same directory for the same database and user
- Revoke the existing grant rule

The two commands being contrasted would look as follows:

```
dlfm deny link privilege on dir /dlfs/test/APP1 to APP1 for db EMP_TEST
inst HRaccess node olympus.sanjose.bigco.com

dlfm revoke grant for link privilege on dir /dlfs/test/APP1 from APP1 for
db EMP_TEST inst HRaccess node olympus.sanjose.bigco.com
```

The **dlfm deny** command will result in *two* rules defined to the DLFM for the given criteria (the original grant rule and the new deny rule), but the **dlfm revoke** command will result in *no* rules matching the given criteria.

In this situation, it is preferable to revoke the grant rule rather than add the deny rule. If you use deny rules to negate grant rules, you might be required to manage many rules for a given DLFM over time. Most of those rules would be negating each other and would serve no purpose. It is better and easier to manage fewer rules, and authority checking will require less processing overhead when fewer rules apply to a given action.

Ultimately, you must decide the best way to define your DLFM authorization scheme in a way that meets your organization's security goals and needs. Keep the following suggestions and considerations in mind:

- In most situations, you will want to use the **dlfm grant** command to define a more generally applicable grant rule, then negate a specific subset of this rule using one or more **dlfm deny** commands, as illustrated in Example 1. Such an approach tends to be a more manageable way to define these authorization rules, as it often results in fewer rules.
- Generally, it is clearer to use the **dlfm revoke** command to remove an existing rule, rather than using **dlfm deny** to negate an existing rule, as illustrated in Example 2.
- Having fewer rules is usually better than having many rules, both for ease of administration and to reduce processing overhead.

**Related concepts:**

- “Read operation security” on page 105
- “Write operation security” on page 105
- “Security issues for updating linked files” on page 107

**Related reference:**

- “dlfm deny command” on page 186
- “dlfm grant command” on page 189
- “dlfm list registered prefixes command” on page 196
- “dlfm list registered users command” on page 197
- “dlfm list registered directories command” on page 194
- “dlfm revoke command” on page 203
- “dlfm set link security command” on page 206

## Read operation security

For anyone other than a superuser to be able to read a linked file (where the corresponding DATALINK column was defined with FILE LINK CONTROL), all of the following authorizations and conditions are required:

- Authority to connect to the DB2<sup>®</sup> database containing the table of interest.
- SQL SELECT authority against the DATALINK type column of interest, in the given table or view.

If the DATALINK column was defined with READ PERMISSION FS, this authority is not always required, because the user can read the file without having first obtained its URL.

- If this is the first retrieval for this database from the given DATALINK column referencing the given Data Links server, the DLFM on that server must be up and running.
- Access to the DLFS file system referenced in the retrieved DATALINK URL value. For example, by having login access to the Data Links server machine where the Data Links File System (DLFS) is located, or through an NFS mount of the given DLFS.
- The DLFF *must* always be loaded when the given DLFS is mounted (UNIX) or the given drive is online (Windows).
- Directory traversal authority in the native file system on all directories in the file's path.
- If the DATALINK column was defined with READ PERMISSION FS, read permission for the given file, as defined in the native file system.
- If the DATALINK column was defined with READ PERMISSION DB:
  - The Data Links server must be up and running.
  - The user must include the *read token* provided in the retrieved DATALINK URL value to open the file in READ mode.
  - The file must be opened within the time interval specified by the `dl_expint` database configuration parameter as defined for the database from which the DATALINK URL was retrieved.

### Related concepts:

- “Advanced file management security features” on page 98
- “Advanced file management security rules and guidelines” on page 101
- “Write operation security” on page 105
- “Security issues for updating linked files” on page 107

## Write operation security

**Important:** When you update a file using a method that does not involve update-in-place operations (for example, by unlinking the file first, or using

the DLREPLACECONTENT scalar function to replace the file) you are not writing directly to a linked file. The authorizations and conditions listed here only apply when using update-in-place operations. For further details about the other methods of updating a linked file, see the Related links section below.

For anyone other than a superuser to be able to write to a linked file (where the corresponding DATALINK column was defined with FILE LINK CONTROL), all of the following authorizations and conditions are required:

- Authority to connect to the DB2<sup>®</sup> database containing the table of interest.
- SQL SELECT authority against the DATALINK type column of interest, in the given table or view.

If the DATALINK column was defined with WRITE PERMISSION FS, this authority is not always required, because the user can read the file without having first obtained its URL.

- The given DATALINK column must NOT have been defined with WRITE PERMISSION BLOCKED.
- If this is the first retrieval for this database from the given DATALINK column referencing the given Data Links server, the DLFM on that server must be up and running.
- If the DATALINK column was defined with WRITE PERMISSION ADMIN, the authority to write to files for the given database and the directory of interest, as granted with the **dlfm grant** command on the given Data Links server.
- Access to the DLFS file system referenced in the retrieved DATALINK URL value. For example, by having login access to the Data Links server machine where the Data Links File System (DLFS) is located, or through an NFS mount of the given DLFS.
- The DLFF *must* always be loaded when the given DLFS is mounted (UNIX) or the given drive is online (Windows).
- Directory traversal authority in the native file system on all directories in the file's path.
- If the DATALINK column was defined with WRITE PERMISSION FS, write permission for the given file, as defined in the native file system.
- If the DATALINK column was defined with WRITE PERMISSION ADMIN:
  - The Data Links server must be up and running.
  - The user must include the *write token* provided in the retrieved DATALINK URL value to open the file in WRITE mode.
  - The file must be opened in WRITE mode for the first time within the time interval specified by the `dl_wt_iexpint` database configuration parameter, as defined for the database from which the DATALINK URL was retrieved.



- SQL UPDATE authority against the DATALINK column to finalize the update (for example, using the DLNEWCOPY scalar function).
- If the DATALINK column was defined with REQUIRING TOKEN FOR UPDATE, the user must include the same *write token* used to modify the linked file in the DATALINK URL value specified with the scalar function (for instance, DLNEWCOPY), when finalizing the update in the DB2 database.

**Related concepts:**

- “Read operation security” on page 105
- “Security issues for updating linked files” on page 107
- “Updating approaches overview” on page 111
- “Approach 1: Unlink/update/relink files” on page 113
- “Approach 2: Update linked files/native file system defines access” on page 115
- “Approach 3: Update linked files/customize file access privileges” on page 116
- “Approach 4: Replace a linked file's contents” on page 131

**Securing linked files during updates: issues**

The following section describes the security issues for writing to (updating) linked files.

**Security issues for updating linked files**

There are several different approaches that you can take when performing update operations on linked files. Each approach allows for different levels of file access security. Briefly, these approaches are as follows:

**Approach 1—Unlink/perform updates/relink an individual file.**

Update operations occur on files after they are unlinked. Any privileged user with the required SQL UPDATE authority, file system permission, and file link authority can update the file.

**Approach 2—Update linked files referenced from table DATALINK columns defined with WRITE PERMISSION FS.**

Any privileged user on the system where the file resides can write to the file.

**Approach 3—Update linked files referenced from table DATALINK columns defined with WRITE PERMISSION ADMIN.**

File write access privileges get defined by the Data Links Manager Administrator using the **dlfm grant** or **dlfm deny** commands.

**Approach 4—Replace a linked file's contents with another file's contents.**

The update operations are performed within a single unit of work.

Any privileged user on the system where the replacement file resides can update the linked file.

The approach that you use depends on your installation's goals, needs, and any restrictions that your DB2<sup>®</sup> host database DATALINK column definitions impose.

**Related concepts:**

- "Advanced file management security rules and guidelines" on page 101
- "Write operation security" on page 105
- "Updating approaches overview" on page 111
- "Approach 1: Unlink/update/relink files" on page 113
- "Approach 2: Update linked files/native file system defines access" on page 115
- "Approach 3: Update linked files/customize file access privileges" on page 116
- "Approach 4: Replace a linked file's contents" on page 131

**Related reference:**

- "dlfm deny command" on page 186
- "dlfm grant command" on page 189
- "dlfm revoke command" on page 203
- "dlfm set link security command" on page 206

---

## Part 3. Application Programmer's Reference

<b>Chapter 5. Updating linked files</b> . . . . .	111
Updating approaches overview . . . . .	111
Approach 1: Unlink/update/relink files . . . . .	113
Approach 1: procedure . . . . .	114
Unlinking and relinking files . . . . .	114
Approach 2: Update linked files/native file system defines access . . . . .	115
Approach 3: Update linked files/customize file access privileges . . . . .	116
Approach 3: details . . . . .	118
Approach 3 configuration tasks . . . . .	118
Approach 3 maintenance tasks . . . . .	120
Approach 3 application program flow . . . . .	122
Backing out updates in progress . . . . .	126
Usage considerations for write tokens . . . . .	128
Recovering missing write tokens . . . . .	129
Approach 4: Replace a linked file's contents	131
Approach 4: details . . . . .	132
Replacing a linked file's contents using approach 4 . . . . .	132
Approach 4 Usage Scenario . . . . .	133
Updating approaches summary . . . . .	134



---

## Chapter 5. Updating linked files

This chapter details the considerations for updating linked files, and describes several different approaches you can take to update linked files.

---

### Updating approaches overview

Data Links Manager considers **any** changes made to the contents of a linked file to be *file update* operations.

Before you begin updating linked files, you must choose an outcome strategy for the linked files. Your chosen strategy will guide you in determining the specific updating approach to use. You can use either of the following strategies:

- Copy the original file to another location, and add new content to another file with a *different* name. You will have a different (new) linked file at the end of the updating process.

The main advantage of creating new files is that your original files remain untouched in case you need them (for example, as backups or to quickly see the history of changes for a particular file).

- Keep the same file and file name throughout the updating process, regardless of all content changes.

The advantages of keeping the same file and file name include: less work during the updating process, less chance of user confusion, and requires less storage space than creating brand-new files. DB2<sup>®</sup> provides file backup and recovery utilities that encompass linked files, and Data Links Manager provides features that enable you to back out any changes to the contents of a linked file.

Here is an example that illustrates the contrasts between the two outcome strategies.

Suppose you want to perform an update on a particular linked file using the new file outcome strategy. You copy the linked file to another location, create a new version of the file with a different name, then add your new content to that new file. You perform an SQL UPDATE operation in DB2 to change the corresponding DATALINK column value to point to the new file. All of these are straightforward tasks, but consider this inherent risk: when you change the file names, you create a small time window in which DB2 users could have retrieved the original file with the old name. Then, when accessing the file, it could either be unlinked or gone.

By contrast, if you choose to keep the same file and file name throughout the update process, you could use Updating Approach 4, which is fully described later in this section. Using Updating Approach 4 would enable you to have the same outcome — an updated version of the original linked file. However, consider these points:

- The updating process takes place within a single unit of work, meaning less effort for you or your application.
- The original linked file's contents remain available should the transaction fail or roll back in DB2.
- The original linked file's contents remain accessible during the updating process, so there are no potential "time window" risks for DB2 users who might be trying to access the file.

There are benefits and trade-offs associated with each outcome strategy. Ultimately, you must choose an outcome strategy that is based on your organization's goals and requirements. Once you have made that decision, you will be able to assess the updating approach or approaches that will work best for you.

Data Links Manager enables you to choose from several different approaches when performing file update operations. Briefly, these approaches are as follows:

**Approach 1—Unlink/perform updates/relink an individual file.**

Update operations occur on files after they are unlinked. Any privileged user with the required SQL UPDATE authority, file system permission, and file link authority can update the file.

**Approach 2—Update linked files referenced from table DATALINK columns defined with WRITE PERMISSION FS.**

Any privileged user on the system where the file resides can write to the file.

**Approach 3—Update linked files referenced from table DATALINK columns defined with WRITE PERMISSION ADMIN.**

File write access privileges get defined by the Data Links Manager Administrator using the **dlfm grant** or **dlfm deny** commands.

**Approach 4—Replace a linked file's contents with another file's contents.**

The update operations are performed within a single unit of work. Any privileged user on the system where the replacement file resides can update the linked file.

Each of these file updating approaches assumes that you want to keep the same file and file name throughout the updating process, though Approach 1, for example, can be adapted to the new file outcome strategy, as your application's needs dictate.

All of the file updating approaches enable you to customize a particular aspect of the file updating process. These approaches range from simple to complex, and each one has its own set of advantages and disadvantages. Compare and contrast each approach, then decide which one will work best for you based on your organization's goals, needs, and any restrictions that might apply based on the DB2 host environment's administration.

**Related concepts:**

- "Approach 1: Unlink/update/relink files" on page 113
- "Approach 2: Update linked files/native file system defines access" on page 115
- "Approach 3: Update linked files/customize file access privileges" on page 116
- "Approach 4: Replace a linked file's contents" on page 131

**Related reference:**

- "Updating approaches summary" on page 134

---

## Approach 1: Unlink/update/relink files

This is a simple approach for updating linked files. Whenever you need to modify or change a linked file, you unlink (remove) that file reference from a DB2® database DATALINK column, make any necessary modifications to the file itself on the Data Links server, then relink (add) the file reference into the DATALINK column again. After you relink the file, it is under Data Links Manager control once again.

**Important:** Be aware of the following issues when using this approach:

- You must ensure that you have SQL UPDATE authority on both the given table and DATALINK column to update the DATALINK value, or you will be unable to unlink and relink the file.
- Files in an unlinked state are no longer under Data Links Manager control. The original file access permissions as defined by the native file system get restored, so any privileged file system user can modify the unlinked file (for example, change the file contents, rename the file, or delete the file). Also, you must ensure that you have the necessary file system permissions to update an unlinked file.
- Files in an unlinked state are no longer referenced in a DB2 database. Because the unlinked files are no longer referenced in a database, they are not accessible for any database metadata searches, for database backups, etc.

- You must ensure that you have file linking authority both on the required DB2 database and in the required Data Links server directory, or you will be unable to relink the file after updates are complete.

**Related concepts:**

- “Approach 2: Update linked files/native file system defines access” on page 115
- “Approach 3: Update linked files/customize file access privileges” on page 116
- “Approach 4: Replace a linked file's contents” on page 131

**Related tasks:**

- “Unlinking and relinking files” on page 114

**Related reference:**

- “dlfm grant command” on page 189
- “dlfm set link security command” on page 206
- “Updating approaches summary” on page 134

---

## Approach 1: procedure

The following section describes how to manually link and unlink files.

### Unlinking and relinking files

This procedure describes how to unlink and relink a file using Approach 1.

The unlink and relink operations must be done in two separate transactions. The unlink operation has to be committed before the file can be modified, and then you must relink the file again in another, separate transaction.

**Prerequisites:**

- You must ensure that required users have file linking authority both on the required DB2 database and in the required Data Links server directory, or they will be unable to relink the file after updates are complete.
- You must ensure that the required users have the necessary native file system permissions to update an unlinked file.
- You must ensure that required users have SQL UPDATE, DELETE, and INSERT authorities on both the given table and DATALINK column to update the DATALINK value, or they will be unable to unlink and relink the file.

**Procedure:**



To unlink a file from a DATALINK column, you can do either of the following:

- Use the SQL UPDATE statement to modify the DATALINK column value to be NULL.
- Use the SQL DELETE statement to delete the entire row.

When a file is in an unlinked state, you can update or replace it as you normally would (for example, by using an editor or executing a copy command).

To relink a file into a DATALINK column, you can do either of the following:

- Use the SQL UPDATE statement to modify the DATALINK column value to the path of the updated file.
- Use the SQL INSERT statement to insert a new row into the table.

**Related concepts:**

- “Updating approaches overview” on page 111
- “Approach 1: Unlink/update/relink files” on page 113

**Related reference:**

- “DELETE statement” in the *SQL Reference, Volume 2*
- “INSERT statement” in the *SQL Reference, Volume 2*
- “UPDATE statement” in the *SQL Reference, Volume 2*
- “Updating approaches summary” on page 134

---

## **Approach 2: Update linked files/native file system defines access**

This is a simple approach for updating linked files. Whenever you need to modify or change a linked file, make the required changes to the file on the Data Links server where the file resides. That is all you need to do.

The process of making changes to the contents of a linked file while a DATALINK column value in a database is pointing to that file is called *updating-in-place*. Any changes to linked files during an update-in-place operation are instantly visible to any authorized user.

**Requirements:**

- You must define DB2® database DATALINK columns with the WRITE PERMISSION attribute set to FS and the RECOVERY attribute set to NO.
- You must be a privileged user on the Data Links server where the file resides to be able to write to the file.

**Important:** Be aware of the following issues when using this approach:

- The DB2 coordinated backup and recovery options are not supported for columns defined with the WRITE PERMISSION attribute set to FS. If a recovery operation occurs, the linked file will not be restored to a consistent state with database data.
- There is no way to back out changes made to linked files. You must maintain your own backup copies if you want them.
- For columns defined with the WRITE PERMISSION attribute set to FS, the Data Links Manager Administrator must ensure that the proper file access controls are in place for each corresponding linked file in the column. By contrast, columns defined with the WRITE PERMISSION attribute set to DB have file access control centralized in the database, which decreases administration and maintenance overhead.
- Any changes to linked files during an update-in-place operation are visible to any authorized users immediately. If you need to ensure that the updated file is visible only in conjunction with updates to the metadata in the table, then this might not be the best approach for you to use. Consider using updating Approaches 1 or 4. Those approaches enable you to have file updates as part of a larger unit of work on a database.

**Related concepts:**

- “Approach 1: Unlink/update/relink files” on page 113
- “Approach 3: Update linked files/customize file access privileges” on page 116
- “Approach 4: Replace a linked file's contents” on page 131

**Related reference:**

- “Updating approaches summary” on page 134

---

### Approach 3: Update linked files/customize file access privileges

This approach enables you to perform updates-in-place on linked files, as with Approach 2. Unlike Approach 2, you or your application must be an authorized user on both the DB2<sup>®</sup> server where the linked file reference is stored, **and** on the Data Links server where the linked file resides. The authorizations, and way you grant them, are different for the DB2 host server and the Data Links server. If you or your application are not authorized on both servers, you will not be able to write to the linked file.

The access control mechanism uses both DB2 database and Data Links Manager features.

- To open files for updating, a write token is required. You must define DB2 table DATALINK columns with the WRITE PERMISSION attribute set to

ADMIN to generate a write token upon request. Write token requests get made with a SELECT statement that invokes one or more scalar functions provided especially for making write token requests. Only authorized users can retrieve a write token using the SELECT statement.

- Authorized users are those DB2 host server users that have been granted the "privilege" of writing to linked files on specific Data Links servers. You grant the writing privilege to specific DB2 server users with the **dlfm grant** command.
- The user who is performing the update has exclusive write access to the linked file until the update operations are complete, and the DB2 host has been notified about the new version of the file with an SQL UPDATE statement. When a linked file has been opened using a write token, it goes into an *update-in-progress state*. Until DB2 has been notified about a new version of the linked file, the file will remain in the update-in-progress state, and all other users can only read the file.

Approach 3 requires specific configuration and maintenance tasks to be performed, and that certain application programming logic be used. You might need a system or database administrator to perform some of the required configuration and maintenance tasks.

However, Approach 3 is the most customizable updating approach. You can define and control **who** can perform **write operations** to specific sets of linked files on a Data Links server. Other important benefits to using this approach include the ability to use the DB2 coordinated backup and recovery options, and the ability to back out file changes.

**Requirements:**

- You must define DB2 database DATALINK columns with the WRITE PERMISSION attribute set to ADMIN, and the READ PERMISSION attribute set to DB.
- If you want to back out file changes, you must define DB2 database DATALINK columns with the RECOVERY attribute set to YES.

**Restrictions:**

- If you are using the Network File System (NFS), it must be at version 3.0 or later.

**Related concepts:**

- "Approach 3 application program flow" on page 122
- "Usage considerations for write tokens" on page 128

**Related tasks:**

- "Approach 3 configuration tasks" on page 118

- “Approach 3 maintenance tasks” on page 120

**Related reference:**

- “dlfm grant command” on page 189
- “Updating approaches summary” on page 134

---

## **Approach 3: details**

The following sections describe:

- How to set up and configure your DB2 host server tables and your Data Links servers to use Approach 3
- Ongoing maintenance tasks that you might need to perform when using Approach 3
- How to write applications to update your linked files when using Approach 3
- How to back out updates-in-progress
- Usage considerations for write tokens
- How to recover missing write tokens

### **Approach 3 configuration tasks**

This section describes how to set up and configure your DB2 host server tables and your Data Links servers to use updating approach 3.

**Prerequisites:**

You must have Administrator authority on the required DB2 host server and the required Data Links servers.

**Procedure:**

To configure your environment to use Approach 3:

1. Define your DB2 table DATALINK columns with the following attributes.

*Table 4. Required DATALINK column attributes for Approach 3*

Attribute name	Required value	Notes
WRITE PERMISSION	ADMIN  REQUIRING TOKEN FOR UPDATE or NOT REQUIRING TOKEN FOR UPDATE	ADMIN is a required attribute. You must also designate whether to require the same write token value that is used to modify the file also be used when signalling the completion of update operations to DB2.  <ul style="list-style-type: none"> <li>• If you use the NOT REQUIRING TOKEN FOR UPDATE attribute, you only need the write token value when opening a linked file.</li> <li>• If you use the REQUIRING TOKEN FOR UPDATE attribute, the same write token value is also required to be included when invoking the SQL UPDATE statement with DLNEWCOPY or DLPREVIOUSCOPY scalar functions.</li> </ul>
READ PERMISSION	DB	
RECOVERY	YES or NO	This attribute must be set to YES to enable coordinated backup and recovery, and to enable the ability to back out file changes. If you do not want either ability, you do not need to use this attribute, or you can set it to NO.

2. For a particular DATALINK column, determine the DB2 server users who should be authorized to obtain a write token with which they can update a linked file referenced by that column. You must grant the following authorities to these users:

- The authorization to SELECT from that DATALINK column.

- The write privilege on the Data Links server where the corresponding linked file is stored.
3. Determine the Data Links server directories where the linked files reside for each DATALINK column.  
You must grant the required database users the write privilege to these directories to enable them to generate a write token, which in turn will enable them to write to the linked files.
  4. Grant the write privilege to the required users for the required directories on a Data Links server by issuing the **dlfm grant** command on the appropriate Data Links server.

For example, issuing the following command:

```
dlfm grant write privilege on dir /dlfs/test to user SHERRYG for db  
EMP_TEST inst HRaccess node olympus.sanjose.bigco.com
```

enables DB2 server user SHERRYG to generate a write token that can be used to write to linked files in the /dlfs/test directory, and all of its subdirectories, when SHERRYG is updating files referenced on DB2 server database EMP\_TEST under the HRaccess instance on the olympus.sanjose.bigco.com node.

**Important:** On a given Data Links server directory, any users that do not explicitly have the write privilege granted to them for the files referenced from DATALINK columns defined with the WRITE PERMISSION ADMIN attribute are prevented from writing to those linked files.

You can now write your applications to perform updates using Approach 3.

**Related concepts:**

- “Approach 3 application program flow” on page 122

**Related tasks:**

- “Approach 3 maintenance tasks” on page 120

**Related reference:**

- “dlfm grant command” on page 189

### **Approach 3 maintenance tasks**

This section lists the ongoing maintenance tasks that you might need to perform when using Approach 3.

**Prerequisite:**

You must have DB2 Data Links Manager Administrator authority.

**Procedure:**

- To see all or specific files that are in an update-in-progress state on a Data Links server or for a particular DB2 database (for example, to recuperate a lost write token) use one of the **dlfm list upd\_in\_progress** commands.

**Example:** Issuing the following command:

```
dlfm list upd_in_progress files for db EMPLOYEE inst DB2 node
storage.ca2.bigco.com tid 10 tsid 5 user joshua
```

shows all linked files that are currently in an update-in-progress state from table id number 10, in the EMPLOYEE database, in the tablespace id number 5, for the user joshua. The command output will also show the write tokens that are in use.

- To list all users and their access privileges on a specific directory on a Data Links server, use the **dlfm list registered users** command.

**Example:** Issuing the following command:

```
dlfm list registered users for directory /localfs/dbfiles/photos on
db employee inst acct1 node storage.ca2.bigco.com
```

shows all people currently using the employee database, in instance acct1, on node storage.ca2.bigco.com, and their file access privileges (link or write) for the Data Links server directory /localfs/dbfiles/photos.

- To list all Data Links server directories to which all or a specific group of users has access, and their access privileges, use the **dlfm list registered directories** command.

**Example:** Issuing the following command:

```
dlfm list registered directories for all users on db drawings inst
acct2 node storage.ak1.bigco.com
```

shows all Data Links server directories and file access privileges (link or write) for all specifically authorized users on the drawings database, in instance acct2, on node storage.ak1.bigco.com.

- To assign the write privilege to specific users for specific sets of linked files, use the **dlfm grant** command. For more information about using this command, see the Related Links section.
- To deny (not allow) specific users the privilege of writing to specific sets of linked files, use the **dlfm deny** command. For more information about using this command, see the Related Links section.
- To revoke the effects of the **dlfm grant** or the **dlfm deny** command for specific users to specific sets of linked files, use the **dlfm revoke** command. For more information about using this command, see the Related Links section.
- To back out updates that are in progress, update the affected DATALINK column file reference on the DB2 server using the DLPREVIOUSCOPY scalar function. For more information about this task, see the Related Links section.

- To recover misplaced write tokens, for example, should a write token get lost. This task can become critical because, once a user modifies a particular linked file with a write token, it is as if that user has a "lock" on the file. Until the update operations are complete, only that specific user, using that specific write token, can modify the linked file. Also, if a particular DATALINK column is defined with the REQUIRING TOKEN FOR UPDATE attribute, that same write token is needed to issue the SQL UPDATE statement with DLNEWCOPY or DLPREVIOUSCOPY scalar functions. For more information about this task, see the Related Links section.

**Related concepts:**

- "Approach 1: Unlink/update/relink files" on page 113
- "Approach 2: Update linked files/native file system defines access" on page 115
- "Approach 3 application program flow" on page 122
- "Approach 4: Replace a linked file's contents" on page 131

**Related tasks:**

- "Backing out updates in progress" on page 126
- "Recovering missing write tokens" on page 129

**Related reference:**

- "dlfm deny command" on page 186
- "dlfm grant command" on page 189
- "dlfm list registered users command" on page 197
- "dlfm list upd\_in\_progress files for db command" on page 198
- "dlfm list upd\_in\_progress files for prefix command" on page 199
- "dlfm list registered directories command" on page 194
- "dlfm revoke command" on page 203
- "Updating approaches summary" on page 134

**Approach 3 application program flow**

After you have configured your environment to use Approach 3, you can write application programs to perform updates.

Suppose you have a table called EMPLOYEE, which has the following schema.

```
CREATE TABLE EMPLOYEE (
  ID INTEGER NOT NULL,
  NAME      VARCHAR(20),
  DEPT_NO  SMALLINT,
  TITLE    VARCHAR(50),
```



```

PHOTO   DATALINK LINKTYPE URL FILE LINK CONTROL
        INTEGRITY ALL
        READ PERMISSION DB
        WRITE PERMISSION ADMIN REQUIRING TOKEN FOR UPDATE
        RECOVERY YES ON UNLINK RESTORE,
RESUME  DATALINK LINKTYPE URL FILE LINK CONTROL
        INTEGRITY ALL
        READ PERMISSION DB
        WRITE PERMISSION ADMIN NOT REQUIRING TOKEN FOR UPDATE
        RECOVERY YES ON UNLINK RESTORE,
PRIMARY KEY (ID)
)

```

EMPLOYEE is in the HR\_DB database. The DB2® server where the EMPLOYEE table resides is called HR\_DB.XYZ.COM. The referenced linked files for the PHOTO column are stored on the Data Links server HR\_SRV.XYZ.COM, in the /hr/emp\_pict directory.

Figure 3 on page 124 illustrates the typical calls and operations that an application program must perform to update linked files referenced in the PHOTO column of the EMPLOYEE table when using Approach 3.

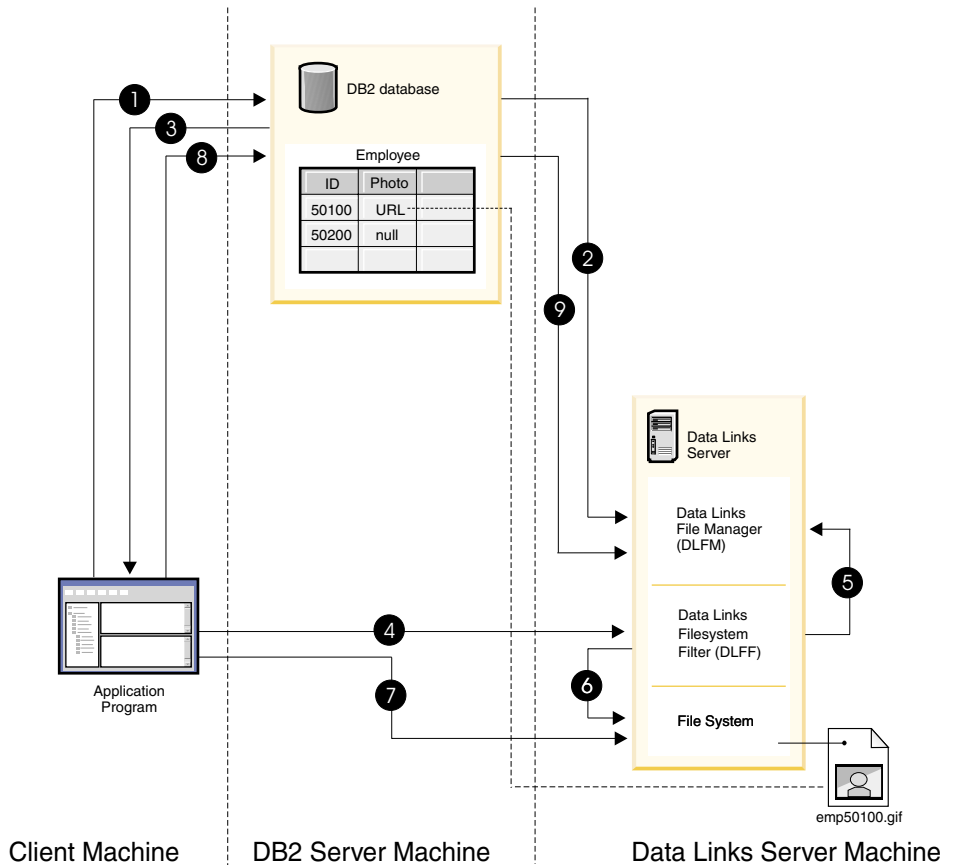


Figure 3. Approach 3 application program flow

The following explanations include sample SQL statements which, together, show a complete programming example.

1. The application program connects to a DB2 database, and issues a SELECT statement on a DATALINK column in a table row to retrieve a URL path with a write token.

Example SQL statement:

```
EXEC SELECT DLURLCOMPLETEWRITE( PHOTO )
INTO :url
FROM EMPLOYEE
WHERE ID = 50100;
```

In the example SQL statement, the URL path is stored in the url host variable.

**Note:** The DLURLPATHWRITE scalar function could have been used instead of DLURLCOMPLETEWRITE to retrieve just the path for the file to be opened.

2. DB2 checks with the Data Links File Manager (DLFM) to see if the connected user has the authority to update the given file.
3. If the connected user has authority to update the file, DB2 returns a file reference value that embeds a write token. Example:  
HTTP://HR\_SRV.XYZ.COM/a/b/xxxx;emp50100.gif, where xxxx is the embedded write token.
4. The application program uses the file reference value to open the linked file.

Example of a file system call in a C application:

```
fptr = fopen(/a/b/xxxx;emp50100.gif);
```

5. The Data Links Filesystem Filter (DLFF) validates the write token.
6. If the write token is valid, DLFF allows the file open operation to proceed, and returns a file descriptor pointer to the application program.  
If the write token is invalid, then DLFF does not allow the file open operation to proceed, and returns a standard error code to the application. The error code value is determined by the native file system.
7. The application program uses the file descriptor value to read the file and write data to the file.

**Attention:** At this point, the application program has exclusive write access to the file. Until the file changes get finalized by performing an SQL UPDATE in DB2 using the DLNEWCOPY or DLPREVIOUSCOPY scalar functions, anyone else can only read the file, not write to the file.

8. When the file changes are completed, the application program notifies the DB2 database that the new version of the file is ready, and that the archiving process can be started upon SQL commit. The application program makes this notification by issuing an SQL UPDATE statement to update the same row and column with a value constructed by the DLNEWCOPY scalar function. The update involving the scalar function invocation is qualified by the table row, and includes the original file reference and the write token if required.

Example SQL statement:

```
EXEC SQL UPDATE EMPLOYEE  
SET PHOTO = DLNEWCOPY( :url, 1 )  
WHERE ID = 50100;
```

```
EXEC SQL COMMIT;
```

**Important:**

- You must have SQL UPDATE authority on the table and column to finalize the update process.

- The need to provide a write token in the DLNEWCOPY scalar function invocation depends on whether you defined the DB2 host table DATALINK column with the REQUIRING TOKEN FOR UPDATE attribute or with the NOT REQUIRING TOKEN FOR UPDATE attribute.
9. DB2 forwards information about the UPDATE request to the corresponding DLFM. Should a valid write token be required and provided, and the DATALINK column is set with the RECOVERY attribute to YES, DLFM triggers the file archive process to asynchronously back up the modified file.

**Attention:** The Data Links File System (DLFS) prevents anyone from modifying the file while it is being archived. If someone attempts to modify the file during the archiving process, a file system error message gets generated.

**Related concepts:**

- “Approach 3: Update linked files/customize file access privileges” on page 116
- “Usage considerations for write tokens” on page 128

**Related tasks:**

- “Backing out updates in progress” on page 126
- “Recovering missing write tokens” on page 129

**Related reference:**

- “Select-statement” in the *SQL Reference, Volume 1*
- “UPDATE statement” in the *SQL Reference, Volume 2*
- “DLPREVIOUSCOPY scalar function” in the *SQL Reference, Volume 1*
- “DLURLPATHWRITE scalar function” in the *SQL Reference, Volume 1*
- “DLNEWCOPY scalar function” in the *SQL Reference, Volume 1*
- “DLURLCOMPLETEWRITE scalar function” in the *SQL Reference, Volume 1*

**Backing out updates in progress**

You can easily back out file updates that are still in progress by issuing an SQL UPDATE statement that invokes the scalar function DLPREVIOUSCOPY, instead of invoking the scalar function DLNEWCOPY.

**Prerequisites:**

DATALINK columns must be defined with the RECOVERY attribute set to YES.

**Example:**

Suppose you have a table called EMPLOYEE, which has the following schema.

```
CREATE TABLE EMPLOYEE (  
  ID INTEGER NOT NULL,  
  NAME      VARCHAR(20),  
  DEPT_NO   SMALLINT,  
  TITLE     VARCHAR(50),  
  PHOTO     DATALINK LINKTYPE URL FILE LINK CONTROL  
           INTEGRITY ALL  
           READ PERMISSION DB  
           WRITE PERMISSION ADMIN REQUIRING TOKEN FOR UPDATE  
           RECOVERY YES ON UNLINK RESTORE,  
  RESUME    DATALINK LINKTYPE URL FILE LINK CONTROL  
           INTEGRITY ALL  
           READ PERMISSION DB  
           WRITE PERMISSION ADMIN NOT REQUIRING TOKEN FOR UPDATE  
           RECOVERY YES ON UNLINK RESTORE,  
  PRIMARY KEY (ID)  
)
```

EMPLOYEE is in the HR\_DB database. The DB2 server where the EMPLOYEE table resides is called HR\_DB.XYZ.COM. You have started updating file emp50100.gif, but you decide that you do not want to change the file after all, and you want to revert back to the original, unchanged version of emp50100.gif. The file is stored on the Data Links server HR\_SRV.XYZ.COM, in the /hr/emp\_pict directory.

To undo the file update, you would issue the following SQL UPDATE statement:

```
EXEC SQL UPDATE EMPLOYEE  
  SET PHOTO = DLPREVIOUSCOPY( 'HTTP://HR_SRV.XYZ.COM/hr/emp_pict/xxxx;emp50100.gif', 1 )  
  WHERE ID = 50100;  
  
EXEC SQL COMMIT;
```

xxxx is the write token value.

#### **Related concepts:**

- “Approach 3: Update linked files/customize file access privileges” on page 116
- “Usage considerations for write tokens” on page 128

#### **Related tasks:**

- “Recovering missing write tokens” on page 129

#### **Related reference:**

- “UPDATE statement” in the *SQL Reference, Volume 2*

## Usage considerations for write tokens

A *write token* is the authorization key embedded in a retrieved DATALINK column value when the DATALINK column is defined with the WRITE PERMISSION ADMIN attribute. The write token value is generated and returned by using the scalar functions DLURLCOMPLETEWRITE or DLURLPATHWRITE.

Write tokens have an essential role in Updating Approach 3, because they provide the authorization mechanism for restricting the users who can perform write operations on a linked file. Also, if a particular DATALINK column is defined with the REQUIRING TOKEN FOR UPDATE attribute, that same write token is needed when signalling the completion of update operations to DB2.

When a user obtains and uses a write token to modify a particular linked file, it is as if that user has a "lock" on the file. Until the file update operations are complete, only that specific user, using that specific write token value, can write to that specific linked file. That specific user must continue to use that specific write token for any sort of access to that linked file, until all file update operations are completed. No one else can *write* to the file until the "controlling user" has completed all file update operations. Anyone else can still *read* the linked file, assuming that they have a valid read or write token value.

Once a write token value is in use, it remains valid for as long as the user is updating the file. However, if there is a large time interval between when a write token value gets generated and when it actually gets used, the token value might expire — that is, become invalid. When someone uses an invalid write token, the attempted file operation will fail, providing an additional layer of file access security.

The DB2® configuration parameter DL\_WT\_IEXPINT controls the initial write token expiration time interval. This interval is an amount of time, in seconds, between when a write token gets generated and the *first* event in which the write token value gets used. The triggering event is the first file open for write call after the write token gets generated.

The default DL\_WT\_IEXPINT value is 60 seconds. You can change this value to be up to one calendar year, in seconds.

**Recommendation:** Use write tokens only for file update (write) operations. Use read tokens only for file read access. It is possible to use a write token to read a file that requires a read token, but it is not advisable to do so unless there is no other alternative. Allowing or encouraging users to casually use write tokens because they *might* want to write to a file allows for a potentially

large number of write tokens to be generated at any given time, but remain unused. In a secure environment, it is best to restrict access to write tokens as much as possible.

**Related concepts:**

- “Approach 3: Update linked files/customize file access privileges” on page 116
- “Approach 3 application program flow” on page 122

**Related tasks:**

- “Approach 3 configuration tasks” on page 118
- “Recovering missing write tokens” on page 129

**Related reference:**

- “Data Links Write Token Initial Expiry Interval configuration parameter - dl\_wt\_iexpint” in the *Administration Guide: Performance*

## Recovering missing write tokens

Once a user modifies a particular linked file with a write token, it is as if that user has a “lock” on the file. Until the update operations are complete, only that specific user, using that specific write token, can write to the linked file. No one else will be able to write to the file. Also, if a particular DATALINK column is defined with the REQUIRING TOKEN FOR UPDATE attribute, that same write token is needed to issue the SQL UPDATE statement with DLNEWCOPY or DLPREVIOUSCOPY scalar functions.

The Data Links File Manager (DLFM) on a Data Links server keeps track of all files that are currently being updated, along with the write tokens that are in use. If a write token is misplaced before you are able to complete any updates that are in progress (for example, you lose the token), you can easily get the token back and continue with the update using the following procedure.

**Prerequisites:**

You must have DB2 Data Links Manager Administrator authority.

You must know the name, instance, and node of the DB2 database where the updates are in progress.

**Procedure:**

To recover a write token:

1. Log on, as the DB2 Data Links Manager Administrator, to the system where the file that is being updated resides.

2. Enter one of the the **dlfm list upd\_in\_progress** files commands.  
The return information contains the pathname and write token for all files that are currently being updated for the specified DB2 database.
3. Continue performing the required update operations using the write token value that you just recovered.

**Example:**

You have a table, EMPLOYEE, in a database, HR\_DB. The database instance is called HR\_INST. The DB2 server where the EMPLOYEE table resides is called HR\_DB.XYZ.COM.

You lost the write token while you were updating file emp50100.gif, and you need it in order to continue updating the file. The file is stored on the Data Links server HR\_SRV.XYZ.COM, in the /hr/emp\_pict directory. To obtain the write token, you would do the following:

1. Log onto the Data Links server where emp50100.gif resides.
2. Enter the following command:

```
dlfm list upd_in_progress files for db HR_DB inst HR_INST node
HR_DB.XYZ.COM
```

The command output might look like this:

```
DATABASE: HR_DB
INSTANCE: HR_INST
NODE      : HR_DB.XYZ.COM
```

```
TABSID TABID COLID UPDATER USERID FILENAME (WITH WRITE TOKEN)
-----
2       2       1 hrap1 (2069 ) /hr/emp_pict/04E6_D1thRE2_Oiuz.VtRA;emp50100.gif
2       2       3 newton (1130 ) /hr/emp_rez/04E6_D2.7bU2_IBAuKyZLD11;emp007.ocr
```

User ID 2069 (hrap1) is in the process of updating emp50100.gif using the write token value 04E6\_D1thRE2\_Oiuz.VtRA. Notice that the Data Links server also reports that user ID 1130 (newton) is in the process of updating a file (emp007.ocr) that is referenced in *another* DATALINK column of the same table in the HR\_DB database.

**Related concepts:**

- “Approach 3: Update linked files/customize file access privileges” on page 116
- “Approach 3 application program flow” on page 122
- “Usage considerations for write tokens” on page 128

**Related reference:**

- “dlfm list upd\_in\_progress files for db command” on page 198
- “dlfm list upd\_in\_progress files for prefix command” on page 199



---

## Approach 4: Replace a linked file's contents

With this approach, you do not change the content of linked files directly. Instead, you replace the contents of the linked file with the contents of another file. The linked file keeps its original name and permissions. The file with which you replace a linked file's contents is referred to as a *replacement file*. To obtain a replacement file, you can either make a copy of the original linked file and make changes to that copy, or use an existing file that contains the required changes.

You perform the replacement operation with an SQL UPDATE transaction on the value in the DATALINK column that contains the original linked file reference. The scalar function DLREPLACECONTENT enables you to specify both the original and replacement file names.

The major benefit to using Approach 4 is that changes to linked files become visible to users only when the SQL UPDATE transaction gets committed in the DB2® database. If the transaction fails or gets rolled back, the original contents of the linked file remain unchanged.

### Requirements:

- Replacement files must reside in the same directory as the linked files.
- You must have sufficient disk space available to accommodate the replacement files.
- Replacement file names must consist of the original file name plus a suffix string that can be a maximum of 20 characters. For example, suppose the original file is named resume1.ocr. The replacement file name could be resume1.ocr.update, but not resume1a.ocr.
- You must ensure that you have granted file linking privileges for the required users to the required Data Links server directories.

### Related concepts:

- “Approach 1: Unlink/update/relink files” on page 113
- “Approach 2: Update linked files/native file system defines access” on page 115
- “Approach 3: Update linked files/customize file access privileges” on page 116
- “Approach 4 Usage Scenario” on page 133

### Related reference:

- “UPDATE statement” in the *SQL Reference, Volume 2*
- “DLREPLACECONTENT scalar function” in the *SQL Reference, Volume 1*
- “dlfm grant command” on page 189

- “dlfm set link security command” on page 206
- “Updating approaches summary” on page 134

---

## Approach 4: details

This section describes how to replace a linked file’s contents using Approach 4, and provides an example that shows how you can use Approach 4.

### Replacing a linked file's contents using approach 4

This section provides a step-by-step procedure for replacing a linked file’s contents using Updating Approach 4.

#### Prerequisites:

- You must have a replacement file available.
- Replacement files must reside in the same directory as the linked files.
- You must have sufficient disk space available to accommodate the replacement files.
- Replacement file names must consist of the original file name plus a suffix string that can be a maximum of 20 characters. For example, suppose the original file is named resume1.ocr. The replacement file name could be resume1.ocr.update, but not resume1a.ocr.
- You must ensure that you have granted the file link privilege to the required users, for the required Data Links server directories.
- You must ensure that the required users have SQL UPDATE authority on the required DB2 database table and DATALINK columns.

#### Procedure:

To replace a linked file’s contents using Updating Approach 4:

1. Obtain a replacement file for the linked file that you want to update, and place it in the same file system directory as the linked file.
2. Use the DLREPLACECONTENT scalar function in an SQL UPDATE statement to update the source DATALINK column value in a DB2 table. The statement must include references to both the original and replacement files.

When the update transaction commits, the replacement file is renamed to be the original linked file’s name. The permissions of the original linked file are retained.

#### Related concepts:

- “Approach 1: Unlink/update/relink files” on page 113

- “Approach 2: Update linked files/native file system defines access” on page 115
- “Approach 3: Update linked files/customize file access privileges” on page 116
- “Approach 4: Replace a linked file's contents” on page 131
- “Approach 4 Usage Scenario” on page 133

**Related reference:**

- “UPDATE statement” in the *SQL Reference, Volume 2*
- “DLREPLACECONTENT scalar function” in the *SQL Reference, Volume 1*
- “Updating approaches summary” on page 134

### Approach 4 Usage Scenario

Suppose you have a table called EMPLOYEE, which has the following schema.

```
CREATE TABLE EMPLOYEE (
  ID INTEGER NOT NULL,
  NAME      VARCHAR(20),
  DEPT_NO   SMALLINT,
  TITLE     VARCHAR(50),
  PHOTO     DATALINK LINKTYPE URL FILE LINK CONTROL
           INTEGRITY ALL
           READ PERMISSION DB
           WRITE PERMISSION ADMIN REQUIRING TOKEN FOR UPDATE
           RECOVERY YES ON UNLINK RESTORE,
  RESUME    DATALINK LINKTYPE URL FILE LINK CONTROL
           INTEGRITY ALL
           READ PERMISSION DB
           WRITE PERMISSION ADMIN NOT REQUIRING TOKEN FOR UPDATE
           RECOVERY YES ON UNLINK RESTORE,
  PRIMARY KEY (ID)
)
```

EMPLOYEE is in the HR\_DB database. The DB2® server where the EMPLOYEE table resides is called HR\_DB.XYZ.COM. The referenced linked files for the RESUME column are stored on the Data Links server HR\_SRV.XYZ.COM, in the /hr/emp\_resumes directory.

You want to update the linked file resume1.doc for employee ID 50100. The linked file reference is HTTP://HR\_SRV.XYZ.COM/hr/emp\_resumes/resume1.doc, which is stored in DATALINK column RESUME in EMPLOYEE. To perform the update, you would do the following:

1. Copy the resume1.doc file to a new file, resume1.doc.new, in the same directory where resume1.doc is stored.
2. Make the required updates to resume1.doc.new in your word processor.
3. Issue the following SQL UPDATE statement:

```
EXEC SQL UPDATE EMPLOYEE
SET RESUME = DLREPLACECONTENT( 'HTTP://HR_SRV.XYZ.COM/hr/emp_resumes/resume1.doc',
                              'HTTP://HR_SRV.XYZ.COM/hr/emp_resumes/resume1.doc.new' )
WHERE ID = 50100;
```

4. When the update transaction commits, resume1.doc.new is renamed to resume1.doc. The permissions of the original resume1.doc file are retained.

**Related concepts:**

- “Approach 1: Unlink/update/relink files” on page 113
- “Approach 2: Update linked files/native file system defines access” on page 115
- “Approach 3: Update linked files/customize file access privileges” on page 116
- “Approach 4: Replace a linked file's contents” on page 131

**Related reference:**

- “UPDATE statement” in the *SQL Reference, Volume 2*
- “DLREPLACECONTENT scalar function” in the *SQL Reference, Volume 1*
- “Updating approaches summary” on page 134

---

## Updating approaches summary

Table 5 on page 135 summarizes the DATALINK column definition requirements, restrictions, advantages, and disadvantages of all four updating approaches.

Table 5. Linked file updating approaches summary

Approach	Restrictions	DATALINK column requirements <sup>1</sup>	Benefits	Potential issues
1 - Unlink, update, then relink files	None	None	A simple approach to implement.	<ul style="list-style-type: none"> <li>• Unlinked files are no longer under Data Links Manager control. The original file access permissions as defined by the native file system get restored, so any privileged file system user can modify the unlinked file (for example, change the file contents, rename the file, or delete the file). Also, you must ensure that the required users have the necessary native file system permissions to update an unlinked file.</li> <li>• Unlinked files are no longer referenced in a DB2 database. Because the unlinked files are no longer referenced in a database, they are not accessible for any database metadata searches, for database backups, etc.</li> <li>• You must ensure that required users have file linking authority both on the required DB2 database and in the required Data Links server directory, or they will be unable to relink the file after updates are complete.</li> <li>• You cannot perform updates-in-place.</li> </ul>

Table 5. Linked file updating approaches summary (continued)

Approach	Restrictions	DATALINK column requirements <sup>1</sup>	Benefits	Potential issues
<p>2 - Update linked files/native file system defines access</p>		<ul style="list-style-type: none"> <li>• WRITE PERMISSION attribute set to FS</li> <li>• RECOVERY attribute set to NO</li> </ul>	<ul style="list-style-type: none"> <li>• Simple approach to implement.</li> <li>• Enables update-in-place operations. Any changes to linked files during an update-in-place operation are visible to database users immediately, and you need not return to the DB2 host table to perform an SQL UPDATE operation.</li> </ul>	<ul style="list-style-type: none"> <li>• DB2 coordinated backup and recovery options are not supported for columns defined with the WRITE PERMISSION attribute set to FS. If a recovery operation occurs, the linked file will not be restored to a consistent state with database data.</li> <li>• Cannot back out changes made to linked files.</li> <li>• For columns defined with the WRITE PERMISSION attribute set to FS, the Data Links Manager Administrator must ensure that the proper file access controls are in place for each corresponding linked file in the column.</li> <li>• Any changes to linked files during an update-in-place operation are visible to database users immediately. If you need to ensure that an updated file is visible only in conjunction with updates to the metadata in the DB2 host table, then this might not be the best approach for you.</li> </ul>

Table 5. Linked file updating approaches summary (continued)

Approach	Restrictions	DATALINK column requirements <sup>1</sup>	Benefits	Potential issues
<p>3 - Update linked files/customize file access</p>	<p>If using Network File System (NFS), must be at version 3.0 or later</p>	<ul style="list-style-type: none"> <li>• WRITE PERMISSION attribute set to ADMIN</li> <li>• READ PERMISSION attribute set to DB</li> <li>• To back out file changes, column must also be defined with the RECOVERY attribute set to YES</li> </ul>	<ul style="list-style-type: none"> <li>• Authorization to modify linked files can be controlled at a very fine level.</li> <li>• Update privileges do not need to be specified for each and every file, as with Approach 2, but instead can be specified by groupings of files or of users on the Data Links server where the files are stored.</li> <li>• Enables update-in-place operations. Any changes to linked files during an update-in-place operation are visible to database users immediately, but you must perform an SQL UPDATE operation on the DB2 host table to actually commit the changes.</li> <li>• DB2 coordinated backup and recovery options supported.</li> <li>• Can back out file changes if necessary, if DATALINK table column defined with the RECOVERY attribute set to YES.</li> </ul>	<ul style="list-style-type: none"> <li>• Multiple tasks necessary to implement this approach.</li> <li>• You must ensure that you have granted file writing privileges for the required users to the required Data Links server directories.</li> <li>• Any changes to linked files during an update-in-place operation are visible to database users immediately. If you need to ensure that an updated file is visible only in conjunction with updates to the metadata in the DB2 host table, then this might not be the best approach for you.</li> <li>• Cannot back out changes made to linked files if a table's DATALINK column is not defined with the RECOVERY attribute set to YES.</li> </ul>

Table 5. Linked file updating approaches summary (continued)

Approach	Restrictions	DATALINK column requirements <sup>1</sup>	Benefits	Potential issues
4 - Replace a linked file's contents	Replacement files must meet these requirements: <ul style="list-style-type: none"> <li>• Must reside in the same directory as the linked files you want to update.</li> <li>• Files names must consist of the original file name plus a suffix string that can be a maximum of 20 characters.</li> </ul>	None	<ul style="list-style-type: none"> <li>• Simple approach to implement.</li> <li>• Linked file changes become visible only when committed in DB2, so if transaction fails or rolls back, original file contents remain unchanged.</li> </ul>	<ul style="list-style-type: none"> <li>• A replacement file is not under Data Links Manager or DB2 control until it replaces the original file.</li> <li>• Need additional disk space to accommodate replacement files.</li> <li>• You must ensure that you have granted file linking privileges for the required users to the required Data Links server directories.</li> </ul>
<b>Notes:</b> 1. All four updating approaches require that the DATALINK column be defined with the FILE LINK CONTROL attribute.				

**Related concepts:**

- “Updating approaches overview” on page 111
- “Approach 1: Unlink/update/relink files” on page 113
- “Approach 2: Update linked files/native file system defines access” on page 115
- “Approach 3: Update linked files/customize file access privileges” on page 116
- “Approach 4: Replace a linked file's contents” on page 131



---

## Part 4. Troubleshooting

<b>Chapter 6. Server recovery</b> . . . . .	141
Data Links server machine failure . . . . .	141
Failure and recovery overview . . . . .	141
DB2 Data Links Manager system setup and backup recommendations . . . . .	143
File system backup and restore recommendations . . . . .	143
Bringing the file system directory hierarchy to the current point in time . . . . .	144
db2_recon_aid utility. . . . .	146
DB2 Data Links Manager recovery scenarios	148

<b>Chapter 7. Data Links File Manager error message reference</b> . . . . .	155
DLFM001I . . . . .	157
DLFM101E . . . . .	157
DLFM121E . . . . .	158
DLFM122E . . . . .	158
DLFM124E . . . . .	158
DLFM126E . . . . .	158
DLFM128E . . . . .	159
DLFM129I . . . . .	159
DLFM151I . . . . .	159
DLFM201E . . . . .	160
DLFM202E . . . . .	160
DLFM203E . . . . .	160
DLFM204E . . . . .	161
DLFM205E . . . . .	161
DLFM206E . . . . .	161
DLFM207E . . . . .	161
DLFM208I . . . . .	161
DLFM209E . . . . .	162
DLFM210E . . . . .	162
DLFM211E . . . . .	162
DLFM221E . . . . .	163
DLFM222I . . . . .	163
DLFM223E . . . . .	163
DLFM252E . . . . .	164
DLFM254I . . . . .	164
DLFM341E . . . . .	164
DLFM402E . . . . .	164
DLFM501E . . . . .	165
DLFM701E . . . . .	171
DLFM703E . . . . .	171
DLFM704E . . . . .	172

DLFM706W . . . . .	172
DLFM707I . . . . .	172
DLFM721E . . . . .	172
DLFM741E . . . . .	173
DLFM742E . . . . .	173
DLFM743E . . . . .	173
DLFM744E . . . . .	174
DLFM746E . . . . .	174
DLFM747E . . . . .	174
DLFM748I . . . . .	175
DLFM749E . . . . .	175
DLFM750I . . . . .	175
DLFM751E . . . . .	175
DLFM801I . . . . .	176
DLFM802I . . . . .	176
DLFM803I . . . . .	176
DLFM804I . . . . .	177
DLFM805I . . . . .	177
DLFM806I . . . . .	177
DLFM807I . . . . .	178
DLFM808I . . . . .	178
DLFM809I . . . . .	178
DLFM810I . . . . .	179
DLFM811E . . . . .	179
DLFM812I . . . . .	179
DLFM813I . . . . .	180
DLFM814I . . . . .	180
DLFM900I . . . . .	180
DLFM901E . . . . .	180
DLFM908E . . . . .	181
DLFM909E . . . . .	181



---

## Chapter 6. Server recovery

This chapter describes how to perform recovery after a disk or machine failure on the Data Links server, and the backup strategies that the administrator must undertake to support such disaster recovery scenarios.

Unless otherwise noted, the information in this chapter is common to Data Links servers running on AIX, Solaris Operating Environment, and Windows.

---

### Data Links server machine failure

In the case of a machine failure on a Data Links server, DB2 applications interacting with the Data Links File Manager can hang. Use the **db2 force application** command to force DB2 applications off the system.

Also, the following components might require recovery after a Data Links server machine fails:

- The DB2 database containing a table which has a DATALINK column
- The Data Links Manager database (DLFM\_DB)
- Data Links Filesystem Filter (DLFF)
- File systems under the control of the DLFF that are registered to the Data Links Manager
- The Data Links Manager archive area

#### Related concepts:

- “Failure and recovery overview” on page 141

#### Related reference:

- “FORCE APPLICATION Command” in the *Command Reference*

---

### Failure and recovery overview

If a disk that contains files referenced by a DATALINK column fails, all the user files, along with the directory hierarchy of the file system, might be destroyed. To recover from such a scenario, the administrator should make periodic backups of the file system containing the user data and directory hierarchy, so that it can be restored. The restored file system must preserve directory and file ownerships, and time stamps.

After restoring the file system, the directory structure must be brought up to the point-in-time of the crash by applying the directory changes that occurred after the file system backup was taken. After this step, the DB2® **RECONCILE** command must be run on all tables referencing files on the damaged disk. The *db2\_recon\_aid* utility is provided to simplify this task.

Following a crash, there are three possible file states:

1. Files that are in linked state, and are referenced in a DATALINK column with the RECOVERY NO attribute, are treated as follows:
  - If the file is not found on the file system, the corresponding DATALINK column value will be set to NULL.
  - If the file is found, and the DATALINK column also has the READ PERMISSION FS and WRITE PERMISSION FS attributes defined, no additional checks will be made to validate the correctness of the file.
  - If the file is found, and the referencing DATALINK column also has the WRITE PERMISSION BLOCKED attribute defined, the file's modification time and size will be checked. If there is a mismatch in the values, the DATALINK column value will be set to NULL.
2. Files that are in a linked state, when the corresponding DATALINK columns have the RECOVERY YES attribute, will be restored from the archive server if the file modification time is different than the file modification time at link time, or if the file is not found.

If the modification time of the version on the file system is different, it is renamed with extension .MOD so that the more recent changes are not lost. The archived version is still retrieved, and the renamed version is reported in the exception report.

If a renamed version of the file with .MOD extension already exists, the file will not be retrieved, the DATALINK column value will be changed to NULL, and it will be reported in the exception report.
3. Files that are in the unlinked state on the file server are not restored or checked for correctness.

**Related reference:**

- "RECONCILE Command" in the *Command Reference*
- "db2\_recon\_aid - RECONCILE Multiple Tables Command" in the *Command Reference*
- "Data Links server machine failure" on page 141
- "db2\_recon\_aid utility" on page 146

---

## DB2 Data Links Manager system setup and backup recommendations

The following system setup and backup procedures are recommended for easier recovery:

1. Place the DLFM database (DLFM\_DB), any file systems under control of the DB2 Data Links Filesystem Filter (DLFF), the DLFM backup directory, and the DLFM home directory on different file systems. Ensure they do not share disks.
2. Back up any file systems under the control of the DLFF and the DLFM on a storage manager, such as IBM's Tivoli Storage Manager, on a regular basis. In the event of disk failure, this will provide added protection.
3. Attempt to have only **one** DB2 database associated with **one or more** DB2 Data Links Managers. If possible, avoid having multiple databases associated with one DB2 Data Links Manager, because certain recovery scenarios will become more complex than is necessary. Perform full database backups of the DB2 databases and the DLFM\_DB database on the Data Links server at regular intervals. Coordinate the timing of the database backups so that they occur together.

### Related concepts:

- "Failure and recovery overview" on page 141

### Related reference:

- "File system backup and restore recommendations" on page 143

---

## File system backup and restore recommendations

To ensure your IT environment's data integrity and recoverability, it is imperative that you institute a regular backup scheme.

Many commercial backup-and-restore programs are available for all types of file systems. These programs enable you to back up your file systems onto any media that you choose, and typically provide easy-to-use restore operations. An example of such a program is IBM's Tivoli Storage Manager. Tivoli Storage Manager runs in all environments that the DB2 database product supports.

Your operating system will also provide backup-and-restore utilities. Windows NT and Windows 2000 provide the Backup utility. AIX provides the System Storage Management utilities. On Solaris operating environments, you can use the **ufsdump** and **ufsrestore** commands. See your operating system documentation for more information about the backup utilities.

**Important:** To reduce recovery time after a crash, use an incremental backup strategy where level 0 refers to a full backup, and levels 1 through 9 refer to incremental backups. A level *n* backup backs up only those files that have changed since a level (*n*-1) backup. After a level *n* backup, the next backup to be taken will be a level (*n*+1) backup.

**Related concepts:**

- “Failure and recovery overview” on page 141
- “Bringing the file system directory hierarchy to the current point in time” on page 144

**Related reference:**

- “Data Links server machine failure” on page 141

---

## Bringing the file system directory hierarchy to the current point in time

After a file system restore operation, you must manually bring the file system directory hierarchy to the current point in time by recreating directories as necessary.

Use the Data Links Manager `fsysadm.log` file to help you determine directories that you need to recreate. Data always gets appended to the `fsysadm.log`.

**On AIX<sup>®</sup> and Solaris Operating Environments**, the directory changes are logged in the `INSTHOME/sqllib/fsysadm.log` file, where `INSTHOME` is the home directory of the Data Links Administrator. There is one entry for each event. Setting the attributes of a file is also logged. The format of the entries for the `fsysadm.log` file is as follows.

```
Time = <timestamp> EUID = <integer> UID = <integer> GID = <integer> Mode = <octal>  
Action = <CREATE/REMOVE/SETATTR> Object type = <DIR/FILE> Path = <fully qualified name>
```

where:

- *Time* is the time of the activity in local time
- *EUID* is the effective user ID of the user performing the action
- *UID* is the user ID attribute of the file or directory that was created, or whose attributes were modified
- *GID* is the group ID attribute of the file or directory that was created, or whose attributes were modified
- *Mode* is the octal representation of the mode of the file or directory

where Action can be:

- *CREATE* indicates a directory was created

- *REMOVE* indicates the file or directory was removed
- *SETATTR* indicates the mode of the file or directory was modified by the user

where Object type can be:

- *DIR* the directory
- *FILE* the file

and where Path is the fully qualified path of the file or directory.

**On Windows<sup>®</sup> systems**, the directory changes are logged in the `x:\sql\lib\dlfm\fsysadm.log` file, where *x*: represents the drive where you installed DB2<sup>®</sup> Data Links Manager. A single event can have multiple entries, depending upon how many users or groups of users have Access Control Lists for the given file or directory. The format of the first entry for the `fsysadm.log` file is as follows.

```
Time = <timestamp> User = <string> Action = <CREATE/REMOVE/SETATTR>
      Object type =<DIR/FILE> Owner = <string> Path = <fully qualified name>
```

The format of any additional entries associated with the first entry is as follows:

```
ACE User = <string> Access = <Hex integer> ACE Type = <Hex integer>
      ACE Flags =<Hex integer>
```

where:

- *Time* is the time of the activity in local time
- *User* is the name of the user performing the action
- *Owner* is the name of the owner of the file or directory
- *Path* is the fully qualified path of the file or directory
- *ACE User* is the the name of a user who has an ACL entry for this file or directory
- *Access\** is the set of flags indicating the types of access the user has
- *ACE Type\** is the type of ACE (for example, allow/deny)
- *ACE Flags\** is a set of ACE type-specific control flags

where Action can be:

- *CREATE* indicates a directory was created
- *REMOVE* indicates the file or directory was removed
- *SETATTR* indicates the mode of the file or directory was modified by the user

where Object type can be:





**-reportdir** *directory\_name*

Required when the **-check** parameter is not specified. Specifies the directory where the utility is to place a report for each of the reconcile operations. For each table on which reconcile was performed, files of the format <tbschema>.<tbname>.<ext> will be created where:

- <tbschema> is the schema of the table.
- <tbname> is the table name.
- <ext> is .ulk or .exp. The .ulk file contains a list of files that were unlinked on the Data Links server, and the .exp file contains a list of files that were in exception on the Data Links server.

**-selective**

An optional parameter that instructs the utility to process only those tables with DATALINK columns containing file references that match the specified **-server** and **-prefixes** criteria.

- If you use this parameter, you must also both the **-server** and **-prefixes** parameters.
- If you do not use this parameter, then all Data Links servers and their prefixes that are registered with the specified DB2 database will either be reconciled, or will be flagged as needing reconciliation.

**-server** *data\_links\_server\_name*

Required when the **-selective** parameter is used. Specifies the name of the Data Links server for which the reconcile operation is to be performed. The name value must be an IP hostname that is identical to the Data Links server hostname registered with the specified DB2 database.

If this parameter is not used, all Data Links servers that are registered with the specified DB2 database will be reconciled.

**-prefixes** *prefix\_list*

Required when the **-selective** parameter is used. Specifies the name of one or more Data Links File System (DLFS) prefixes. Prefix values must start with a slash, and must be registered with the specified Data Links file server. Separate multiple prefix names with a colon (:), but do not include any embedded spaces. For example:  
/dlfsdir1/smith/:/dlfsdir2/smith/.

The path in a DATALINK column value is considered to match the *prefix\_list* if any of the prefixes in the list are a left-most substring of the path.

If this parameter is not used, all prefixes for all Data Links servers that are registered with the specified DB2 database will be reconciled.

**Examples:**

```
db2_recon_aid -db STAFF -check
db2_recon_aid -db STAFF -reportdir /home/smith
db2_recon_aid -db STAFF -check -selective -server
dlmserver.services.com -prefixes /dlfsdir1/smith/
db2_recon_aid -db STAFF -reportdir /home/smith -selective -server
dlmserver.services.com -prefixes /dlfsdir1/smith/:/dlfsdir2/smith/
```

**Related concepts:**

- “Failure and recovery overview” on page 141

**Related reference:**

- “RECONCILE Command” in the *Command Reference*
- “db2\_recon\_aid - RECONCILE Multiple Tables Command” in the *Command Reference*

---

## DB2 Data Links Manager recovery scenarios

This topic presents some sample DB2<sup>®</sup> Data Links Manager failure scenarios and the steps required to recover from them.

The scenarios use the following terms:

**DLFS file system**

Registered prefix (example for AIX: /dlink)

**DLFM backup directory**

Directory where files are backed up (example: /dlfm/dlfm\_backup)

**DLFM instance directory**

Instance directory of the Data Links Manager Administrator ID (example: /home/dlfm)

**DLFM DB2 database**

DB2 database that contains all metadata (DLFM\_DB)

**DB2 database**

Registered database that contains DATALINK data type (example: CROWN)

The example DB2 database is referred to as “CROWN” throughout all scenarios.

**Important:** Some of the following scenarios might require operations to be performed on both the DB2 node and the DB2 File Manager node.

Scenario	Recovery Steps
<p>DB2 database is lost or was accidentally dropped, but DB2 backup and log files are available.</p> <p><b>Important:</b> When a database gets dropped, it happens within the period of time as specified by the db2 database configuration parameter <i>dl_time_drop</i>.</p> <p>In this example, the DB2 database has <b>not</b> been dropped from the Data Links server.</p>	<ol style="list-style-type: none"> <li>1. On the DB2 host, enter the following commands. As a result, all of the affected tables will be put into the Datalink_Reconcile_Not_Possible (DRNP) state. <pre>db2 "restore database CROWN" db2 "rollforward database CROWN to end of logs and stop" db2 "connect to CROWN"</pre> </li> <li>2. Place all tables with DATALINK columns into Datalink_Reconcile_Pending (DRP) state with the following commands: <pre>db2 set integrity for &lt;table&gt; to datalink reconcile pending db2 set integrity for &lt;table&gt; datalink reconcile pending immediate unchecked db2 reconcile &lt;table&gt; dlreport &lt;filename&gt;</pre> </li> </ol>
<p>DB2 database was explicitly dropped, but DB2 backup and log files are available.</p> <p><b>Important:</b> When a database gets dropped, it happens within the period of time as specified by the db2 database configuration parameter <i>dl_time_drop</i>.</p>	<p><b>Requirement:</b> You <i>must</i> take a backup of your DLFM_DB database before updating it as described in the following procedure.</p> <p><b>Recommendation:</b> Work with IBM® Service in updating your DLFM_DB database. This database is an essential part of your Data Links Manager configuration.</p> <ol style="list-style-type: none"> <li>1. Ensure that the drop database operation is complete, and that all files associated with that database have been unlinked.</li> <li>2. On the Data Links server enter the following commands. It is critical that you set the dbid value in the db2 update statement to be exactly as found in the db2 select statement. <pre>db2 "connect to dl_fm_db" db2 "select dbid, dbname, dbinst, hostname from dfm_dbid" db2 "update dfm_dbid set action=5 where dbid='35B3D7BE006BF7B'"</pre> </li> <li>3. On the DB2 host, enter the following commands. As a result, all of the affected tables will be put into the Datalink_Reconcile_Not_Possible (DRNP) state. <pre>db2 "restore database CROWN" db2 "rollforward database CROWN to end of logs and stop" db2 "connect to CROWN"</pre> </li> <li>4. For each table placed in DRNP state in step 3, enter the following commands: <pre>db2 set integrity for &lt;table&gt; to datalink reconcile pending db2 set integrity for &lt;table&gt; datalink reconcile pending immediate unchecked db2 reconcile &lt;table&gt; dlreport &lt;filename&gt;</pre> </li> </ol>

Scenario	Recovery Steps
<p>The DLFM_DB database is lost, but the backup and all log files for the DLFM_DB database are available.</p>	<ol style="list-style-type: none"> <li>1. On the Data Links server enter the following commands: <pre>db2 "restore database dlfm_db" db2 "rollforward database dlfm_db to end of logs and stop"</pre> </li> <li>2. On the DB2 host, enter the following command to run the <b>db2_recon_aid</b> utility. This utility automatically runs RECONCILE for each table with URL file references to the affected Data Links server: <pre>db2_recon_aid -db CROWN -reportdir &lt;dirpath&gt; -selective -server &lt;d1m_hostname&gt; -prefixes &lt;dlfs_prefix&gt;</pre> <ul style="list-style-type: none"> <li>• <b>d1m_hostname</b> is the registered IP hostname of the affected Data Links Manager</li> <li>• <b>dlfs_prefix</b> is the registered prefix corresponding to the affected Data Links File System (DLFS)</li> </ul> </li> </ol>
<p>The DLFM_DB database is lost, a backup of the DLFM_DB database is available, but not all of the log files are available.</p>	<ol style="list-style-type: none"> <li>1. On the Data Links server enter the following commands: <pre>db2 "restore database dlfm_db" db2 "rollforward database dlfm_db to end of logs and stop"</pre> </li> <li>2. On the DB2 host, enter the following commands. As a result, all of the affected tables will be put into the Datalink_Reconcile_Pending (DRP) state. <pre>db2 "restore database CROWN" db2 "rollforward database CROWN to end of logs and stop" db2 "connect to CROWN"</pre> </li> <li>3. Place all tables with data link values into DRP state by entering the following commands: <pre>db2 set integrity for &lt;table&gt; to datalink reconcile pending db2 set integrity for &lt;table&gt; datalink reconcile pending immediate unchecked db2 reconcile &lt;table&gt; dlreport &lt;filename&gt;</pre> </li> </ol>
<p>The Data Links File System (DLFS) is lost.</p>	<ol style="list-style-type: none"> <li>1. Restore the DLFS from your storage manager.</li> <li>2. On the DB2 host, enter the following command to run the <b>db2_recon_aid</b> utility. Notice that by using the <b>-selective</b> options, you can perform reconciliation for just the DLFS that was lost. <pre>db2_recon_aid -db CROWN -reportdir &lt;dirpath&gt; -selective -server &lt;d1m_hostname&gt; -prefixes &lt;dlfs_prefix&gt;</pre> <ul style="list-style-type: none"> <li>• <b>d1m_hostname</b> is the registered IP hostname of the affected Data Links Manager</li> <li>• <b>dlfs_prefix</b> is the registered prefix corresponding to the affected Data Links File System (DLFS)</li> </ul> </li> </ol>
<p>The DLFM backup directory is lost.</p>	<p>Restore the DLFM backup directory from your storage manager.</p>

Scenario	Recovery Steps
<p>The DLFS file system and the DLFM backup directory are lost.</p>	<ol style="list-style-type: none"> <li>1. Restore the DLFM backup directory from your storage manager.</li> <li>2. Restore the DLFS from your storage manager.</li> <li>3. On the DB2 host, enter the following command to run the <b>db2_recon_aid</b> utility. Notice that by using the <b>-selective</b> options, you can perform reconciliation for just the DLFS that was lost. <pre>db2_recon_aid -db CROWN -reportdir &lt;dirpath&gt; -selective -server &lt;d1m_hostname&gt; -prefixes &lt;d1fs_prefix&gt;</pre> <ul style="list-style-type: none"> <li>• <b>d1m_hostname</b> is the registered IP hostname of the affected Data Links Manager</li> <li>• <b>d1fs_prefix</b> is the registered prefix corresponding to the affected Data Links File System (DLFS)</li> </ul> </li> </ol>
<p>The DLFM_DB database, the DLFM backup directory, and the DLFS file system are lost, but the backup and all log files for the DLFM_DB database are available.</p>	<ol style="list-style-type: none"> <li>1. On the Data Links server enter the following commands: <pre>db2 "restore database d1fm_db" db2 "rollforward database d1fm_db to end of logs and stop"</pre> </li> <li>2. Restore the DLFM backup directory from your storage manager.</li> <li>3. Restore the DLFS from your storage manager.</li> <li>4. On the DB2 host, enter the following command to run the <b>db2_recon_aid</b> utility. Notice that by using the <b>-selective</b> options, you can perform reconciliation for just the DLFS that was lost. <pre>db2_recon_aid -db CROWN -reportdir &lt;dirpath&gt; -selective -server &lt;d1m_hostname&gt; -prefixes &lt;d1fs_prefix&gt;</pre> <ul style="list-style-type: none"> <li>• <b>d1m_hostname</b> is the registered IP hostname of the affected Data Links Manager</li> <li>• <b>d1fs_prefix</b> is the registered prefix corresponding to the affected Data Links File System (DLFS)</li> </ul> </li> </ol>

Scenario	Recovery Steps
<p>The DLFM_DB, the DLFS file system, and the DLFM backup directory are lost. The backup of the DLFM_DB database is available, but not all log files are available.</p>	<ol style="list-style-type: none"> <li>1. On the Data Links server enter the following commands: <pre>db2 "restore database dlfm_db" db2 "rollforward database dlfm_db to end of logs and stop"</pre> </li> <li>2. Restore the DLFM backup directory from your storage manager.</li> <li>3. Restore the DLFS from your storage manager.</li> <li>4. On the DB2 host, enter the following command to run the <b>db2_recon_aid</b> utility. Notice that by using the <b>-selective</b> options, you can perform reconciliation for just the DLFS that was lost. <pre>db2_recon_aid -db CROWN -reportdir &lt;dirpath&gt; -selective -server &lt;dlm_hostname&gt; -prefixes &lt;dlfs_prefix&gt;</pre> <ul style="list-style-type: none"> <li>• <b>dlm_hostname</b> is the registered IP hostname of the affected Data Links Manager</li> <li>• <b>dlfs_prefix</b> is the registered prefix corresponding to the affected Data Links File System (DLFS)</li> </ul> </li> </ol>
<p>The DB2 database, the DLFM_DB database, the DLFS file system and DLFM backup directory are lost, but backup, and all log files for the DLFM_DB database are available.</p>	<ol style="list-style-type: none"> <li>1. On the Data Links server enter the following commands: <pre>db2 "restore database dlfm_db" db2 "rollforward database dlfm_db to end of logs and stop"</pre> </li> <li>2. Restore the DLFM backup directory from your storage manager.</li> <li>3. Restore the DLFS from your storage manager.</li> <li>4. On the DB2 host, enter the following commands. As a result, all of the affected tables will be put into the Datalink_Reconcile_Not_Possible (DRNP) state. <pre>db2 "restore database CROWN" db2 "rollforward database CROWN to end of logs and stop" db2 "connect to CROWN"</pre> </li> <li>5. For each table placed in DRNP state in step 4, enter the following commands to place them in DRP state: <pre>db2 set integrity for &lt;table&gt; to datalink reconcile pending db2 set integrity for &lt;table&gt; datalink reconcile pending immediate unchecked db2 reconcile &lt;table&gt; dlreport &lt;filename&gt;</pre> </li> </ol> <p><b>Note:</b> Rollforward to a point-in-time might not put tables that have all DATALINK columns defined with RECOVERY NO into Datalink_Reconcile_Pending (DRP) state. For all such tables, run the RECONCILE utility. You can also use the db2_recon_aid utility to automatically identify and run RECONCILE against those tables.</p>

**Related reference:**

- "RECONCILE Command" in the *Command Reference*
- "db2\_recon\_aid - RECONCILE Multiple Tables Command" in the *Command Reference*

- “db2\_recon\_aid utility” on page 146





---

## Chapter 7. Data Links File Manager error message reference

This topic lists the messages and errors that you might encounter when using the Data Links File Manager on Windows NT, Windows 2000, AIX, or Solaris Operating Environment. Each message or error code is listed along with its corresponding message, possible causes, and suggested actions.

On Windows NT and Windows 2000 systems, error messages are directed to the `x:\sqlib\instance\db2diag.log` file, where `x:` is the directory where you installed DB2 Data Links Manager and `instance` is the name of the instance owned by the Data Links Manager Administrator (dlfm, by default). Error messages are also displayed to the person who executed the **dlfm** command.

On AIX and Solaris Operating Environment systems, error messages are directed to the `INSTHOME/sqlib/db2dump/db2diag.log` file, where `INSTHOME` is the home directory of the instance owner. Error messages are also displayed to the person who executed the **dlfm** command.

In this topic:

- “DLFM001I” on page 157
- “DLFM101E” on page 157
- “DLFM121E” on page 158
- “DLFM122E” on page 158
- “DLFM124E” on page 158
- “DLFM126E” on page 158
- “DLFM128E” on page 159
- “DLFM129I” on page 159
- “DLFM151I” on page 159
- “DLFM201E” on page 160
- “DLFM202E” on page 160
- “DLFM203E” on page 160
- “DLFM204E” on page 161
- “DLFM205E” on page 161
- “DLFM206E” on page 161
- “DLFM207E” on page 161
- “DLFM208I” on page 161
- “DLFM209E” on page 162

- “DLFM210E” on page 162
- “DLFM211E” on page 162
- “DLFM221E” on page 163
- “DLFM222I” on page 163
- “DLFM223E” on page 163
- “DLFM252E” on page 164
- “DLFM254I” on page 164
- “DLFM341E” on page 164
- “DLFM402E” on page 164
- “DLFM501E” on page 165
- “DLFM701E” on page 171
- “DLFM703E” on page 171
- “DLFM704E” on page 172
- “DLFM706W” on page 172
- “DLFM707I” on page 172
- “DLFM721E” on page 172
- “DLFM741E” on page 173
- “DLFM742E” on page 173
- “DLFM743E” on page 173
- “DLFM744E” on page 174
- “DLFM746E” on page 174
- “DLFM747E” on page 174
- “DLFM748I” on page 175
- “DLFM749E” on page 175
- “DLFM750I” on page 175
- “DLFM751E” on page 175
- “DLFM801I” on page 176
- “DLFM802I” on page 176
- “DLFM803I” on page 176
- “DLFM804I” on page 177
- “DLFM805I” on page 177
- “DLFM806I” on page 177
- “DLFM807I” on page 178
- “DLFM808I” on page 178
- “DLFM809I” on page 178
- “DLFM810I” on page 179
- “DLFM811E” on page 179

- “DLFM812I” on page 179
- “DLFM813I” on page 180
- “DLFM814I” on page 180
- “DLFM900I” on page 180
- “DLFM901E” on page 180
- “DLFM908E” on page 181
- “DLFM909E” on page 181

---

## DLFM001I

DLFM server started.

**Cause:** The Data Links File Manager has started successfully.

**Action:** None required.

---

## DLFM101E

Error in DLFM server startup. Reason code = "<reason-code>".

### Reason Code -3

Data Links File Manager initialization failed.

**Cause:** Possible causes are:

1. There was an error in trace initialization.
2. There was an error in Common Subsystem initialization.
3. There was an error in initializing DB2 log manager.
4. There was an error in creating shared resources.
5. There was an error in creating the monitor thread.
6. There was an error in sqlwait or sqlpost code.
7. There was a stop request before DLFM started successfully.
8. There was an error in changing to the Initted state.

**Action:** Perform the following steps:

1. Log on to the system as the DB2 Data Links Manager Administrator.
2. Enter the **dlfm stop** command to stop the Data Links File Manager.
3. Enter the **dlfm shutdown** command to shut down the Data Links Server.
4. Start the Data Links Manager by entering the **dlfm start** command.

5. If the problem persists, look at the error information in the `db2diag.log`.
6. Contact the system administrator for further help.

---

## DLFM121E

Prefix name "<prefix-name>" already registered.

**Cause:** The given *prefix-name* has already been defined on this Data Links File Manager.

**Action:** Specify the correct value for the *prefix-name* parameter. Use the **dlfm list registered prefixes** command to display a list of current prefix registrations.

---

## DLFM122E

Prefix name "<prefix-name>" cannot be registered. Given shared drive already registered under another prefix name.

This error only occurs in the Windows environment.

**Cause:** The shared name specified in the given *prefix-name* has already been registered with this Data Links File Manager under another prefix name.

**Action:** Specify the shared name for a different shared drive in the *prefix-name* being registered. Use the **dlfm list registered prefixes** command to display a list of current prefix registrations.

---

## DLFM124E

Prefix name "<prefix-name>" not registered as a shared drive.

This error only occurs in the Windows environment.

**Cause:** The shared name specified in the given *prefix-name* does not correspond to any shared drive defined on this system.

**Action:** Specify the shared name for a different shared drive in the *prefix-name* being registered. Check the shared name defined to Windows for the shared drive for which you want to register a prefix.

---

## DLFM126E

Specified prefix name "<prefix-name>" is not correct or not registered previously. Return code = "<return-code>".

- Cause:** The given *prefix-name* has not been registered with this Data Links File Manager.
- Action:** Specify the correct *prefix-name* parameter value. Use the **dlfm list registered prefixes** command to display a list of current prefix registrations.
- 

## DLFM128E

Unable to perform requested administration function. Return code = "<return-code>".

- Cause:** The requested administration function has failed. *return-code* represents the internal SQL code associated with the failure.
- Action:** Perform the following steps:
1. See the error information in the `db2diag.log`.
  2. Follow any corrective actions described there.
  3. If the problem persists, report it to IBM Service.
- 

## DLFM129I

Automatic backup of DLFM\_DB <db-scope> has been triggered. Please wait for the backup to complete.

- Cause:** The Data Links File Manager has initiated a backup of its DLFM\_DB database or a tablespace contained therein. This is done in order to preserve critical information recorded as a result of the last **dlfm** command entered.
- Action:** Allow the backup to complete uninterrupted. Depending upon the amount of file information being maintained by the Data Links File Manager, this may take several minutes or more to complete.
- 

## DLFM151I

The following files are being updated:

- Cause:** The set of files linked to a DATALINK column with the WRITE PERMISSION ADMIN attribute for which an update is in progress will be listed following this message. Only those files which have already been opened for update using a write token will be displayed.
- Action:** No action required. However, if a user has misplaced the write token needed to continue or complete the file update, you may want to take note of the token listed along with the file name.

---

## DLFM201E

Error in DLFM registration service. Reason code = "<reason-code>".

### Reason Code -2

Error in database registration.

**Cause:** There is an error with the Data Links File Manager log manager.

**Action:** Perform the following steps:

1. See the error information in the `db2diag.log`.
2. Report the problem to IBM Service.

### Reason Code -4

Error in prefix registration.

**Cause:** There is an error with the Data Links File Manager log manager.

**Action:** Perform the following steps:

1. See the error information in the `db2diag.log`.
2. Report the problem to IBM Service.

For more information about registering a file system, see the Related links at the end of this topic.

---

## DLFM202E

Specified database server registration not valid.

**Cause:** The *database\_name*, *instance\_name*, and/or *node\_name* parameters specified in the command are not valid.

**Action:** Specify the correct values for the *database\_name*, *instance\_name*, and *node\_name* parameters. Use the **dlfm list registered databases** command to display a list of current database server registrations.

---

## DLFM203E

Specified directory not valid.

**Cause:** The *directory* parameter specified in the command is invalid.

**Action:** Specify a valid directory name.

---

## DLFM204E

Specified directory does not belong to a registered prefix.

**Cause:** The specified *directory* parameter in the command is not contained in any registered prefixes.

**Action:** Specify the correct value for the *directory* parameter. Use the **dlfm list registered prefixes** command to display a list of current database prefix registrations.

---

## DLFM205E

Specified directory access control already defined.

**Cause:** The access control for the specified *directory*, *db2\_authorization\_id*, and *database\_server* parameters have already been defined.

**Action:** Use the **dlfm list registered directories** command to display any existing access control entries of these specified parameters before re-running the command.

---

## DLFM206E

Specified directory access control cannot be found.

**Cause:** The access control for the specified *directory*, *db2\_authorization\_id*, and *database\_server* parameters have not been registered.

**Action:** Specify the correct values for the parameters.

---

## DLFM207E

Specified prefix not valid.

**Cause:** The specified *prefix* parameter in the command is not a registered prefix.

**Action:** Specify the correct *prefix* parameter value. Use the **dlfm list registered prefixes** command to display the current prefix registrations. If necessary, use the **dlfm add prefix** command to register the desired prefix.

---

## DLFM208I

This change will not be effective until all of the existing DB2 server connections belonging to "<user-set>" are disconnected from the database (database name "<db-name>", instance "<instance>", node "<node>").

*user-set* describes the set of users (DB2 authids) affected by this command, e.g. an individual user, a group of users, or all users (i.e. PUBLIC).

- Cause:** A command requiring the disconnection of the given DB2 server to the Data Links File Manager has been issued while existing connections may exist.
- Action:** If the change from the specified command is required immediately, terminate all DB2 server connections to this Data Links File Manager initiated by the given set of users belonging to the given database.

---

### DLFM209E

Specified user or group id not valid.

- Cause:** A specified identifier is not a valid user or group id in the system.
- Action:** Either create the identifier in the system or specify an identifier that already exists.

---

### DLFM210E

Specified access control for this user or group has already been defined.

- Cause:** The access control for the specified user or group identifier is already defined.
- Action:** Use the **dlfm list registered replication access control** command to verify that the desired access control has already been defined. If not, replace the user, group, or other identifier with the desired parameter value, and issue the command again.

---

### DLFM211E

Specified access control is not defined for this user or group.

- Cause:** The access control for the specified user or group identifier cannot be found.
- Action:** Specify the correct access control value. You can use the **dlfm list registered users** command, **dlfm list registered directories** command, or **dlfm list registered replication access control** command to display registered access control values.



---

## DLFM221E

DROP\_DLM processing failed. Refer to db2diag.log file for additional details.

- Cause:** The Data Links Manager was unable to initiate unlink and garbage collection processing for the specified database. Possible causes are:
- The Data Links Manager is not running.
  - The values for the *database\_name*, *instance\_name*, and/or *host\_name* parameters were specified incorrectly.
  - There was a failure in one of the component services in the Data Links Manager.
- Action:** Perform the following steps:
1. Make sure that the Data Links Manager is running. If not, enter the **dlfm start** command to restart the DLFM.
  2. Specify the correct values for the *database\_name*, *instance\_name*, and *host\_name* parameters. Use the **dlfm list registered databases** command to display a list of current database server registrations.

If the problem persists, refer to the information in the db2diag.log to see if any component services (Connection Management Service, Transaction Management Service, etc.) have failed. Note the error code in the db2diag.log, and take the appropriate actions suggested under that error code.

---

## DLFM222I

The "<request-type>" request was successful.

**Cause:** The given request (or command) completed successfully.

**Action:** None required.

---

## DLFM223E

The "<request-type>" request failed.

**Cause:** The given request (or command) did not complete successfully.

**Action:** Take note of any other messages and codes accompanying this one, or refer to any additional error information in the db2diag.log file.

---

## DLFM252E

The DLFM registry variable DLFM\_INSTALL\_PATH ("*<variable-value>*") does not point to the required "sqllib" directory.

**Cause:** The DLFM\_INSTALL\_PATH registry variable has the value indicated by *variable-value*, but does not point to the required "sqllib" subdirectory.

**Action:** Use the **db2set** command to assign the DLFM\_INSTALL\_PATH registry variable a path including the "sqllib" directory, in which the Data Links File Manager is installed.

---

## DLFM254I

DLFS configuration file "*<file-name>*" is being restored from backup file "*<backup-file-name>*".

**Cause:** The DLFS configuration file, *file-name*, was not found, but a backup copy, *backup-file-name* was located, and is being used to replace the *file-name* configuration file.

**Action:** None required.

---

## DLFM341E

Data Links Manager previously terminated abnormally. "DLFM SHUTDOWN" command may be required. Number of active processes was: *<num-processes>*, return code = "*<return-code>*".

**Cause:** The Data Links Manager has detected system resources left over from a previous run. This indicates that the Data Links Manager terminated abnormally, and cannot be restarted until the leftover system resources are cleaned up.

**Action:** Perform the following steps:

1. See the error information in the `db2diag.log`.
2. Follow any corrective actions described there.
3. Enter the **dlfm shutdown** command to perform the cleanup of any remaining resources.
4. If the problem persists, report it to IBM Service.

---

## DLFM402E

The DLFM is currently operating in Maintenance mode. Only restart recovery processing is allowed. No other activity can be performed with this DLFM at this time.

**Cause:** There was a severe error in the Data Links File Manager.

- Action:** Perform the following steps:
1. See the error information in the db2diag.log.
  2. Perform any corrective actions.
  3. Restart the Data Links File Manager.
  4. Report the problem to IBM Service if you are unable to resolve it on your own.

---

## DLFM501E

<manager-type> management service "<service-type>" failed. Return code = "<return-code>".

For the following *manager-type* values: Connection, File, Group, Prefix, and Transaction, refer to the corresponding *service-type* below:

### AbortTxn

- Cause:** Possible causes are:
1. There is an error with the Data Links File Manager log manager.
  2. There is an error when accessing system shared resources.
  3. There is an operating system error.
  4. The Data Links File Manager log file is corrupt.
- Action:** Perform the following steps:
1. See the error information in the db2diag.log.
  2. Restart the Data Links File Manager (if needed).
  3. Report the problem to IBM Service.

### BeginTxn

- Cause:** Possible causes are:
1. There is an error with the Data Links File Manager log manager.
  2. There is an error when accessing system shared resources.
  3. There is an operating system error.
  4. The Data Links File Manager log file is corrupt.
- Action:** Perform the following steps:
1. See the error information in the db2diag.log.
  2. Restart the Data Links File Manager (if needed).
  3. Report the problem to an IBM Service.

## **CommitTxn**

**Cause:**

Possible causes are:

1. There is an error with the Data Links File Manager log manager.
2. There is an error when accessing system shared resources.
3. There is an operating system error.
4. The Data Links File Manager log file is corrupt.

**Action:**

Perform the following steps:

1. See the error information in the `db2diag.log`.
2. Restart the Data Links File Manager (if needed).
3. Report the problem to an IBM Service.

## **Connect**

**Cause:**

The database failed to connect to the Data Links File Manager.  
Possible causes are:

1. The database is not registered with the Data Links File Manager.
2. An error occurred when accessing system shared resources.
3. An operating system error occurred.

**Action:**

Perform the following steps:

1. See the error information in the `db2diag.log`.
2. The DB2 Data Links Manager Administrator should register the database with the Data Links File Manager.
3. If the problem persists, report it to IBM Service.

## **DefineGroup**

**Cause:**

Possible causes are:

1. There is an error with the Data Links File Manager log manager.
2. There is an error when accessing system shared resources.
3. There is an operating system error.
4. The Data Links File Manager log file is corrupt.

**Action:**

Perform the following steps:

1. See the error information in the `db2diag.log`.
2. Restart the Data Links File Manager (if needed).
3. Report the problem to IBM Service.

## DeleteDatabase

- Cause:** Possible causes are:
1. There is an error with the Data Links File Manager log manager.
  2. There is an error when accessing system shared resources.
  3. There is an operating system error.
  4. The Data Links File Manager log file is corrupt.
- Action:** Perform the following steps:
1. See the error information in the db2diag.log.
  2. Restart the Data Links File Manager (if needed).
  3. Report the problem to IBM Service.

## DeleteGroup

- Cause:** Possible causes are:
1. There is an error with the Data Links File Manager log manager.
  2. There is an error when accessing system shared resources.
  3. There is an operating system error.
  4. The Data Links File Manager log file is corrupt.
- Action:** Perform the following steps:
1. See the error information in the db2diag.log.
  2. Restart the Data Links File Manager (if needed).
  3. Report the problem to IBM Service.

## Disconnect

- Cause:** Possible causes are:
1. There is an error with the Data Links File Manager log manager.
  2. An error occurred when accessing system shared resources.
  3. An operating system error occurred.
  4. The Data Links File Manager log file is corrupt.
- Action:** Perform the following steps:
1. See the error information in the db2diag.log.
  2. Restart the Data Links File Manager (if needed).
  3. Report the problem to IBM Service.

## LinkFiles

- Cause:** Possible causes are:
1. There is an error with the Data Links File Manager log manager.
  2. There is an error when accessing system shared resources.
  3. There is an operating system error.
  4. The Data Links File Manager log file is corrupt.

- Action:** Perform the following steps:
1. See the error information in the db2diag.log.
  2. Restart the Data Links File Manager (if needed).
  3. Report the problem to IBM Service.

### **PrepareTxn**

- Cause:** Possible causes are:
1. There is an error with the Data Links File Manager log manager.
  2. There is an error when accessing system shared resources.
  3. There is an operating system error.
  4. The Data Links File Manager log file is corrupt.

- Action:** Perform the following steps:
1. See the error information in the db2diag.log.
  2. Restart the Data Links File Manager (if needed).
  3. Report the problem to IBM Service.

### **QueryARTxns**

- Cause:** Possible causes are:
1. There is an error with the Data Links File Manager log manager.
  2. There is an error when accessing system shared resources.
  3. There is an operating system error.
  4. The Data Links File Manager log file is corrupt.

- Action:** Perform the following steps:
1. See the error information in the db2diag.log.
  2. Restart the Data Links File Manager (if needed).
  3. Report the problem to IBM Service.

### **QueryGroups**

- Cause:** Possible causes are:

1. There is an error with the Data Links File Manager log manager.
2. There is an error when accessing system shared resources.
3. There is an operating system error.
4. The Data Links File Manager log file is corrupt.

**Action:** Perform the following steps:

1. See the error information in the db2diag.log.
2. Restart the Data Links File Manager (if needed).
3. Report the problem to IBM Service.

### QueryGroupsExt

**Cause:** Possible causes are:

1. There is an error with the Data Links File Manager log manager.
2. There is an error when accessing system shared resources.
3. There is an operating system error.
4. The Data Links File Manager log file is corrupt.

**Action:** Perform the following steps:

1. See the error information in the db2diag.log.
2. Restart the Data Links File Manager (if needed).
3. Report the problem to IBM Service.

### QueryGroupsExt2

**Cause:** Possible causes are:

1. There is an error with the Data Links File Manager log manager.
2. There is an error when accessing system shared resources.
3. There is an operating system error.
4. The Data Links File Manager log file is corrupt.

**Action:** Perform the following steps:

1. See the error information in the db2diag.log.
2. Restart the Data Links File Manager (if needed).
3. Report the problem to IBM Service.

### QueryPreparedTxns

**Cause:** Possible causes are:

1. There is an error with the Data Links File Manager log manager.

2. There is an error when accessing system shared resources.
3. There is an operating system error.
4. The Data Links File Manager log file is corrupt.

**Action:** Perform the following steps:

1. See the error information in the `db2diag.log`.
2. Restart the Data Links File Manager (if needed).
3. Report the problem to IBM Service.

#### **ReleaseDelete**

**Cause:** The Data Links File Manager is unable to delete the file in its current state. The file was deleted or modified by an administrative user while it was being managed by the Data Links File Manager.

**Action:** Check the status of this file and delete it if necessary.

#### **ReleaseRestore**

**Cause:** The Data Links File Manager is unable to restore the original owners and permissions of this file. The file was deleted or modified by an administrative user while it was being managed by the Data Links File Manager.

**Action:** Check the status of this file and change its ownership and permission attributes or delete the file, as appropriate.

#### **ResolvePrefixId**

**Cause:** Possible causes are:

1. There is an error with the Data Links File Manager log manager.
2. There is an error when accessing system shared resources.
3. There is an operating system error.
4. The Data Links File Manager log file is corrupt.

**Action:** Perform the following steps:

1. See the error information in the `db2diag.log`.
2. Restart the Data Links File Manager (if needed).
3. Report the problem to IBM Service.

#### **ResolvePrefixName**

**Cause:** The prefix for the given file is not registered with the Data Links File Manager.



**Action:** The DB2 Data Links Manager Administrator should register the prefix for the given file with the Data Links server.

### TakeOver

**Cause:** The Data Links File Manager is unable to initiate the management of a file. The file was deleted or modified before the Data Links File Manager could take control of the file.

**Action:** Check the status of this file and change its ownership and permission attributes, as appropriate.

### UnlinkFile

**Cause:** Possible causes are:

1. There is an error with the Data Links File Manager log manager.
2. There is an error when accessing system shared resources.
3. There is an operating system error.
4. The Data Links File Manager log file is corrupt.

**Action:** Perform the following steps:

1. See the error information in the `db2diag.log`.
2. Restart the Data Links File Manager (if needed).
3. Report the problem to IBM Service.

---

## DLFM701E

Invalid number of arguments (<num-args>) specified on "<command>".  
Reason code = "<reason-code>".

**Cause:** Too many or too few arguments were specified with the given *command*. *num-args* is the number of arguments actually specified.

**Action:** Correct the number of arguments specified with the given command. Use the command **dlfm help** to obtain more information about the correct syntax of *command*.

---

## DLFM703E

Unable to get network host entry using given server name "<server-name>".  
Return code = "<return-code>".

**Cause:** The "gethostbyname" system function was unable to resolve the given *server-name*, and failed with the given *return-code*.

**Action:** Verify that the given *server-name* is correctly specified, and is defined in your local Domain Name Server or system hosts file.

---

## DLFM704E

Specified control file name is not associated with this DLFM server.

**Cause:** The "dl\_fm\_export" utility requires that the "control file name" passed to it as the first parameter must match the hostname of the current DLFM server.

**Action:** Verify that the given "control file name" is correctly specified, and matches the hostname (or IP address) of the DLFM server.

---

## DLFM706W

No files to be archived from this DLFM server.

**Cause:** The "dl\_fm\_export" utility has read the contents of the input "control file", but could not process any of its URL entries. This is normally due to there not being any URL's in the file which reference the current DLFM server. As a result, there are no files to be written to the output archive file.

**Action:** Verify that at least one URL in the control file has a hostname matching that of the current DLFM server. If not, then there are indeed no files that need to be written to an archive for this DLFM server, and no further action is required on this server.

---

## DLFM707I

"<command>" completed successfully.

**Cause:** The given *command* completed its processing successfully.

**Action:** None required.

---

## DLFM721E

Invalid argument ("<arg>") supplied for command "<command>".

**Cause:** The given *arg* value is not a valid argument for the given *command*.

**Action:** Check the correct syntax for the given *command* and correct the argument entered accordingly.

---

## DLFM741E

Invalid backup directory name "<dir-name>" specified in DLFM\_BACKUP\_DIR\_NAME registry variable.

**Cause:** The value of the DLFM\_BACKUP\_DIR\_NAME registry variable, *dir-name* is invalid. The directory specified in this variable must be an absolute path (i.e. beginning with a slash character).

**Action:** Verify that the *dir-name* value is an absolute path of the correct format, and which exists on the current system. (This directory is used by the Data Links File Manager for file backups.) Use the **db2set** command to correct the value of the DLFM\_BACKUP\_DIR\_NAME registry variable accordingly.

---

## DLFM742E

Unable to access backup directory name "<dir\_name>" specified in DLFM\_BACKUP\_DIR\_NAME registry variable.

**Cause:** The value of the DLFM\_BACKUP\_DIR\_NAME registry variable, *dir-name* is invalid. The directory specified in this variable must be an absolute path of an existing directory. The given *dir-name* could not be found.

**Action:** Verify that the *dir-name* value is an absolute path which exists on the current system. (This directory is used by the Data Links File Manager for file backups.) Use the **db2set** command to correct the value of the DLFM\_BACKUP\_DIR\_NAME registry variable accordingly.

---

## DLFM743E

Unable to find file "<file-name>" stored in local disk backup.

**Cause:** The given *file-name* was expected to be found in the local disk backup, but was not present.

**Action:** Perform the following steps:

1. See the error information in the `db2diag.log`.
2. Follow any corrective actions described there.
3. Use the **db2set DLFM\_BACKUP\_DIR\_NAME** command to verify that the DLFM\_BACKUP\_DIR\_NAME registry variable points to the directory used by this Data Links File Manager for file backups.

4. If not set correctly, use the **db2set** command to correct the value of the `DLFM_BACKUP_DIR_NAME` registry variable accordingly.
5. Verify that the given *file-name* does exist in the backup directory.

---

## DLFM744E

Unable to access file "<file-name>" in local disk backup.

**Cause:** The given *file-name* was found in the local disk backup, but could not be accessed.

**Action:** Perform the following steps:

1. See the error information in the `db2diag.log`.
2. Follow any corrective actions described there.
3. Use the **db2set** `DLFM_BACKUP_DIR_NAME` command to verify that the `DLFM_BACKUP_DIR_NAME` registry variable points to the directory used by this Data Links File Manager for file backups.
4. If not set correctly, use the **db2set** command to correct the value of the `DLFM_BACKUP_DIR_NAME` registry variable accordingly.
5. Verify that the given *file-name* does exist in the backup directory, and has at least READ permission by the file owner.

---

## DLFM746E

No records found for requested <item-type> list.

**Cause:** There are no items of the type *item-type* found in the Data Links File Manager's database matching any criteria already specified.

**Action:** You might want to re-run this command with different arguments, or by specifying a different set of choices to see if you can retrieve the backup file information you want to locate.

---

## DLFM747E

An SQL error occurred while fetching <item-type> list. Return code = "<return-code>".

**Cause:** An unexpected database error occurred while trying to retrieve a list of items of type *item-type*, according to the

criteria previously specified. The *return-code* value indicates the SQL error code associated with the database action.

- Action:** Perform the following steps:
1. See the error information in the `db2diag.log`.
  2. Follow any corrective actions described there.
  3. If the problem persists, report it to IBM Service.

---

## DLFM748I

Enter the number corresponding to the *<item-type>* you want to choose.

**Cause:** Following this message is a list of valid *item-type* choices. In order for the command to continue processing, you must choose one of the items in this list.

**Action:** Enter the number corresponding to the item you want to choose.

---

## DLFM749E

An SQL error occurred while processing *<item-type>* list. Return code = "*<return-code>*".

**Cause:** An unexpected database error occurred while trying to retrieve a list of items of type *item-type*, according to the criteria previously specified. The *return-code* value indicates the SQL error code associated with the database action.

- Action:** Perform the following steps:
1. See the error information in the `db2diag.log`.
  2. Follow any corrective actions described there.
  3. If the problem persists, report it to IBM Service.

---

## DLFM750I

Using DLFM database "*<dbname>*".

**Cause:** The Data Links File Manager environment indicates that *dbname* is the name of its private DB2 database. This is the database which will be used to retrieve information about the DLFM's backup files.

**Action:** None required.

---

## DLFM751E

Unable to connect to database "*<dbname>*". Return code = "*<return-code>*".

**Cause:** An unexpected database error occurred while trying to connect to the *dbname* database. The *return-code* value indicates the SQL error code associated with the DB2 CONNECT statement.

**Action:** Perform the following steps:

1. See the error information in the `db2diag.log`.
2. Follow any corrective actions described there.
3. Verify that *dbname* is the correct name of the DLFM database. (The default name of this database is "DLFM\_DB".)
4. If the problem persists, report it to IBM Service.

---

## DLFM801I

Starting DLFM migration. Please wait for processing to complete.

**Cause:** The Data Links File Manager migration program has started. This program will convert a previous version's DLFM environment to that of the currently installed DLFM version.

**Action:** Allow the migration to complete uninterrupted. Depending upon the amount of file information being maintained by the Data Links File Manager, this may take several minutes or even quite a bit longer to complete.

---

## DLFM802I

Binding the migration package.

**Cause:** The Data Links File Manager migration program is binding itself to the existing DLFM\_DB database in order to proceed with migration to the currently installed version of the DLFM.

**Action:** Allow the migration to complete uninterrupted. Depending upon the amount of file information being maintained by the Data Links File Manager, this may take several minutes or even quite a bit longer to complete.

---

## DLFM803I

Backing up the DLFM database.

**Cause:** The Data Links File Manager migration program is backing up the existing DLFM\_DB database in order to proceed with migration to the currently installed version of the DLFM.

**Action:** Allow the migration to complete uninterrupted. Depending upon the amount of file information being maintained by the

Data Links File Manager, this may take several minutes or even quite a bit longer to complete.

---

#### DLFM804I

Following database actions are being performed:

**Cause:** The Data Links File Manager migration program is performing numerous database actions against the existing DLFM\_DB database in order to migrate to the currently installed version of the DLFM. The list of database actions and results follows this message.

**Action:** Allow the migration to complete uninterrupted. Depending upon the amount of file information being maintained by the Data Links File Manager, this may take several minutes or even quite a bit longer to complete.

---

#### DLFM805I

"<database-action>": SQL code = "<sql-code>", Return code = "<return-code>".

**Cause:** The Data Links File Manager migration program has performed the indicated *database-action* against the existing DLFM\_DB database in order to migrate to the currently installed version of the DLFM. The *sql-code* and *return-code* indicate the result of the given *database-action*.

**Action:** Allow the given database action and the overall migration program to complete uninterrupted. Depending upon the amount of file information being maintained by the Data Links File Manager, this may take several minutes or even quite a bit longer to complete.

---

#### DLFM806I

"<database-action>": SQL code = "<sql-code>", Return code = "<return-code>", Reason code = "<reason-code>".

**Cause:** The Data Links File Manager migration program has performed the indicated *database-action* against the existing DLFM\_DB database in order to migrate to the currently installed version of the DLFM. The *sql-code*, *return-code*, and *reason-code* indicate the result of the given *database-action*.

**Action:** Allow the given database action and the overall migration program to complete uninterrupted. Depending upon the

amount of file information being maintained by the Data Links File Manager, this may take several minutes or even quite a bit longer to complete.

---

### DLFM807I

Exporting the "<data-name>" data into file "<file-name>".

**Cause:** The Data Links File Manager migration program is exporting the indicated *data-name* into a work file, *file-name*. This export is being performed using the existing DLFM\_DB database in order to migrate to the currently installed version of the DLFM.

**Action:** Allow the export processing and the overall migration program to complete uninterrupted. Depending upon the amount of file information being maintained by the Data Links File Manager, this may take several minutes or even quite a bit longer to complete.

---

### DLFM808I

Number of rows of "<data-name>" data exported = <num-rows>.

**Cause:** The Data Links File Manager migration program has exported *num-rows* of the indicated *data-name* into a work file. This export is being performed using the existing DLFM\_DB database in order to migrate to the currently installed version of the DLFM.

**Action:** Allow the export processing and the overall migration program to complete uninterrupted. Depending upon the amount of file information being maintained by the Data Links File Manager, this may take several minutes or even quite a bit longer to complete.

---

### DLFM809I

Loading the "<data-name>" data from file "<file-name>".

**Cause:** The Data Links File Manager migration program is loading the indicated *data-name* from a work file, *file-name*. This load is being performed into the existing DLFM\_DB database in order to migrate to the currently installed version of the DLFM.

**Action:** Allow the load processing and the overall migration program to complete uninterrupted. Depending upon the amount of



file information being maintained by the Data Links File Manager, this may take several minutes or even quite a bit longer to complete.

---

### DLFM810I

Number of rows of "<data-name>" data loaded = <num-rows1>, number of rows committed = <num-rows2>.

**Cause:** The Data Links File Manager migration program has loaded *num-rows1* of the indicated *data-name* from a work file, and committed *num-rows2* of these rows. This load is being performed into the existing DLFM\_DB database in order to migrate to the currently installed version of the DLFM.

**Action:** Allow the load processing and the overall migration program to complete uninterrupted. Depending upon the amount of file information being maintained by the Data Links File Manager, this may take several minutes or even quite a bit longer to complete.

---

### DLFM811E

The current DLFM database could not be backed up. Return code = "<return-code>". DLFM migration is stopping.

**Cause:** The Data Links File Manager migration program has failed in backing up the existing DLFM\_DB database. *return-code* is the return code from the DB2 Backup utility program.

**Action:** Perform the following steps:

1. See the error information in the db2diag.log.
2. Follow any corrective actions described there.
3. If the problem persists, report it to IBM Service.

---

### DLFM812I

Binding the main DLFM packages.

**Cause:** The Data Links File Manager migration program is binding the currently installed version of the DLFM to the migrated DLFM\_DB database.

**Action:** Allow the bind processing and the overall migration program to complete uninterrupted. Depending upon the amount of file information being maintained by the Data Links File Manager, this may take several minutes or even quite a bit longer to complete.

---

## DLFM813I

All steps of DLFM migration completed successfully.

**Cause:** The Data Links File Manager migration program has completed conversion of the existing DLFM\_DB database to match the currently installed version of the DLFM. All steps have completed successfully.

**Action:** None required.

---

## DLFM814I

DLFM migration ended.

**Cause:** The Data Links File Manager migration program has ended.

**Action:** Take note of any preceding messages indicating success or failure of the DLFM migration program. If successful, no further action is required. If unsuccessful, follow the instructions associated with the error message before attempting to re-run the migration program.

---

## DLFM900I

DLFM server has stopped.

**Cause:** The Data Links File Manager was stopped normally or abnormally.

**Action:** If the DLFM was stopped abnormally, issue the commands **dlfm shutdown** and then **dlfm start** to restart the DLFM.

---

## DLFM901E

A system error occurred. Return code = "<return-code>". The current command cannot be processed. Refer to the db2diag.log file for additional information.

**Cause:** An unexpected system environment or processing error has occurred. Possible causes include the following:

1. Unable to perform an automatic back of the DLFM\_DB database.
2. Unable to obtain file information from the file system.
3. Unable to create a child process.
4. Unable to execute a system command.
5. Unable to get database instance information.
6. An unexpected SQL error occurred.

7. Unable to perform a file system operation, e.g. change directory.
8. Unable to get the system time.
9. An internal processing error occurred.
10. Unable to initialize the db2diag.log file.
11. Unable to generate a unique internal prefix identifier.

**Action:**

Perform the following steps:

1. See the error information in the db2diag.log.
2. Follow any corrective actions described there.
3. If the problem persists, report it to IBM Service.

---

## DLFM908E

Repeated failures occurred while performing "<file-action>" action on file "<file-name>". DLFM is being stopped.

**Cause:**

Data Links Manager failed to take over files from or release files to the file system. One or more files being taken over may have failed in being backed up to the archive. These were files inserted into, or deleted from, a DATALINK column. Before commit processing could complete, one of the following may have occurred:

- The file system containing the file went offline.
- The file was deleted by an administrative user.
- The Archive server or storage is temporarily unavailable.

**Action:**

Perform the following steps:

1. Make sure the file system is available, and check for the existence of the file.
2. If a file was being taken over, but could not be backed up, verify that the Archive server and archive storage are available.
3. If the file was not deleted and is now available, restart the Data Links Manager. This should release the file, and the Data Links Manager should become usable.

If the file was deleted or if the problem persists, report it to IBM Service.

---

## DLFM909E

Repeated failures occurred while replacing file "<existing-file>" with file "<replacement-file>". DLFM is being stopped.

- Cause:** Data Links Manager failed to replace an existing file with a replacement file. Before commit processing could complete, one of the following may have occurred:
- The file system containing the files went offline.
  - The replacement file was deleted by an administrative user.

- Action:** Perform the following steps:
1. Make sure the file system is available, and check for the existence of the replacement file.
  2. If the replacement file was not deleted and is now available, restart the Data Links Manager. Restarting should release the file, and the Data Links Manager should become usable.

If the file was deleted or if the problem persists, report it to IBM Service.

---

## Appendix A. Data Links File Manager commands

This appendix lists the Data Links File Manager (DLFM) commands, and describes the purpose and syntax of each command.

---

### dlfm command

**Requirement:** You must execute this command from the Data Links server using the Data Links Manager Administrator user ID.

▶▶—dlfm—————▶▶

Displays the syntax of all Data Links File Manager commands that apply on the current platform and environment.

To display command descriptions as well as command syntax, use the **dlfm ?** or the **dlfm help** commands.

**Related reference:**

- “dlfm ? command” on page 210
- “dlfm help command” on page 194

---

### dlfm add\_db command

**Requirement:** You must execute this command from the Data Links server using the Data Links Manager Administrator user ID.

▶▶—dlfm add\_db—*database\_name*—*instance\_name*—*node\_name*—————▶▶

Registers a database with the Data Links File Manager (DLFM).

Execute this command **before** defining this DLFM to a DB2 database with the DB2 ADD DATALINKS MANAGER command.

*database\_name*

The name of the database.

*instance\_name*

The name of the database instance.

*node\_name*

The DB2 server node name.

Example:

- `dlfm add_db sampledatabase myinstance myhostname.com`

**Related tasks:**

- “Enabling and registering file systems with DLFM (AIX, Solaris Operating Environment)” on page 65
- “Registering a drive with DLFF (Windows operating systems)” on page 66

**Related reference:**

- “`dlfm list registered prefixes` command” on page 196

---

## dlfm add\_prefix command

**Requirement:** You must execute this command from the Data Links server using the Data Links Administrator user ID.

►—`dlfm add_prefix—prefix_name`—◄

Registers a *prefix* with the Data Links File Manager (DLFM). A prefix is the absolute path of a Data Links File System (DLFS) mount point (on UNIX) or a drive sharename (on Windows) under which linked files are stored.

Execute this command **after** defining the DLFS in which this prefix resides, but **before** referencing a file in that file system in a DATALINK column.

**Attention:** Once you add a prefix you cannot remove it.

*prefix\_name*

Specifies the prefix name. Do not use an underscore ( `_` ) character in the name. **Important:** The file system name is case sensitive for AIX and the Solaris Operating environment. Case is insensitive for Windows file system names.

Example:

- `dlfm add_prefix /myfilesystem`

**Related tasks:**

- “Enabling and registering file systems with DLFM (AIX, Solaris Operating Environment)” on page 65
- “Registering a drive with DLFF (Windows operating systems)” on page 66

- “Listing registered file systems (AIX, Solaris Operating Environment)” on page 68
- “Listing registered drives (Windows operating systems)” on page 68

---

## dlfm bind command

**Requirement:** You must execute this command from the Data Links server using the Data Links Manager Administrator user ID.

▶▶—dlfm bind—————▶▶

Binds the executables used by the Data Links File Manager to the DLFM\_DB database. This command also updates the DB2 statistics for the DLFM\_DB database.

Use this command after applying fixes (for example, from a FixPak) to the Data Links Manager product.

---

## dlfm create command

**Requirement:** You must execute this command from the Data Links server using the Data Links Manager Administrator user ID.

▶▶—dlfm create—————▶▶

Creates all the DLFM\_DB tables used by the Data Links File Manager. Use this command only when installing or rebuilding a new Data Links File Manager.

This command gets invoked automatically as part of the normal Data Links Manager installation process.

**Related concepts:**

- “Actions performed by the DB2 Setup wizard (AIX)” in the *Quick Beginnings for Data Links Manager*
- “Actions performed by the DB2 Setup wizard (Windows)” in the *Quick Beginnings for Data Links Manager*
- “Actions performed by the DB2 Setup wizard (Solaris Operating Environment)” in the *Quick Beginnings for Data Links Manager*

---

## dlfm create\_db command

**Requirement:** You must execute this command from the Data Links server using the Data Links Manager Administrator user ID.

►—dlfm create\_db—◄

Creates and configures the DLFM\_DB database. After the database is created, the archive logging feature is turned on and an offline backup copy of the database is made.

Use this command only when installing or rebuilding a new Data Links File Manager. This command gets invoked automatically as part of the normal Data Links Manager installation process.

Execute this command before issuing a **dlfm start**.

### Related concepts:

- “Actions performed by the DB2 Setup wizard (AIX)” in the *Quick Beginnings for Data Links Manager*
- “Actions performed by the DB2 Setup wizard (Windows)” in the *Quick Beginnings for Data Links Manager*
- “Actions performed by the DB2 Setup wizard (Solaris Operating Environment)” in the *Quick Beginnings for Data Links Manager*

---

## dlfm deny command

### Requirements:

- You must execute this command from the Data Links server using the Data Links Server Administrator ID.
- For any link authorization changes to be effective, the file link security feature must be on (activated). You activate file link security with the **dlfm set link security on** command.

►—dlfm deny—

—write privilege—	—on dir—	—directory_name—	—to—	—user—	—db2_authorization_id—
—link privilege—				—group—	
—all privileges—				—public—	

◄

►—for db—db\_name—inst—db\_inst\_name—node—server\_node\_name—◄

Enables you to deny link and write access privileges to the files stored on a Data Links server for specific DB2 database users. The link privilege enables users to link to files. The write privilege enables users to write to (update)



linked files. You can deny a single privilege, or both privileges, to individual database users, a group of database users, or to all users of a particular database.

**Restriction:** You can only control the write privilege when a DB2 host table DATALINK column is set to WRITE PERMISSION ADMIN.

Table 6 shows all the possible write permission attributes of a DB2 host table DATALINK column, and compares the privileges that you can grant, deny, and revoke based on those attributes.

Table 6. DATALINK column attributes and applicable DLM security privileges

	Link privilege?	Write privilege?	Both Link and Write privileges?
WRITE PERMISSION ADMIN	Yes	Yes	Yes
WRITE PERMISSION FS	Yes	No	No
WRITE PERMISSION BLOCKED	Yes	No	No

**Important:** The execution of this command does not impact any *existing* connections to the Data Links Server from the DB2 host. *Subsequent* connections will reflect all privilege changes.

**link privilege | write privilege | all privileges**

Specifies the privilege that you want to deny: link, write, or both privileges (**all privileges**).

*directory\_name*

Specifies a Data Links server directory. The directory must be on a registered Data Links File System (DLFS), within or beneath a registered prefix. The denied privilege or privileges apply to all files in a specified directory **and** any subdirectories.

You can also specify all directories with an asterisk ( \* ). On AIX and the Solaris Operating Environment, you must enter a backward slash ( \ ) before the asterisk to avoid UNIX command substitution.

**user | group | public**

Specifies the DB2 host users for whom you want to deny the privilege or privileges: a particular user (**user**), a user group (**group**), or all DB2 host users (**public**).

*db2\_authorization\_id*

Specifies the authorization id value of a user or group.

*db\_name*

The name of the remote DB2 UDB database.

*db\_inst* The name of the instance under which the database resides.

*server\_node\_name*

The DB2 server node name.

**Example:**

```
dlfm deny link privilege on dir /dlfs/test to user SHERRYG
for db EMP_TEST inst HRaccess node olympus.sanjose.bigco.com
```

Result: DB2 server user SHERRYG is not allowed to link to files in the /dlfs/test directory, and any of its subdirectories, only when SHERRYG is attempting to link those files to DB2 server database EMP\_TEST under the HRaccess instance on the olympus.sanjose.bigco.com node.

**Related concepts:**

- “Advanced file management security rules and guidelines” on page 101

**Related reference:**

- “dlfm grant command” on page 189
- “dlfm revoke command” on page 203
- “dlfm set link security command” on page 206

---

## **dlfm drop\_db command**

**Requirement:** You must execute this command from the Data Links server using the Data Links Manager Administrator user ID.

```
►—dlfm drop_db—◄
```

Drops the DLFM\_DB database.

**Attention:** Use this command only if none of your DB2 host table DATALINK columns defined with the FILE LINK CONTROL attribute reference any files on this Data Links server. Running this command will completely disable your Data Links server.

---

## dlfm drop\_dlm command

**Requirement:** You must execute this command from the Data Links server using the Data Links Manager Administrator user ID.

►►—dlfm drop\_dlm—*database\_name*—*instance\_name*—*host\_name*—►►

Removes a DB2 database registration from the Data Links File Manager by initiating an asynchronous deletion of all information for a particular database.

Use this command only after dropping a Data Links Manager from a particular database on the DB2 host.

*database\_name*

Specifies the name of the remote DB2 UDB database.

*instance\_name*

Specifies the name of the instance under which the database resides.

*host\_name*

Specifies the host name of the DB2 UDB server on which the database resides.

Example:

- dlfm drop\_dlm sampledatabase myinstance myhostname.com

### Related tasks:

- “Removing information about a DB2 database from Data Links Manager” on page 75

### Related reference:

- “DROP DATALINKS MANAGER Command” in the *Command Reference*

---

## dlfm grant command

### Requirements:

- You must execute this command from the Data Links server using the Data Links Manager Administrator ID.
- For any link authorization changes to be effective, the file link security feature must be on (activated). You activate file link security with the **dlfm set link security on** command.

►►—dlfm grant—

┌ write privilege ├ link privilege └ all privileges	—on dir—	<i>directory_name</i>	—to	user	— <i>db2_authorization_id</i> —►►
				group	
				public	

Enables you to grant link and write access privileges to the files stored on a Data Links server to specific DB2 database users.

- Granting the link privilege enables the specified users to link to files.
- Granting the write privilege enables the specified users to generate and obtain a write token. Users then provide the write token, which is embedded in the filename, to perform an open-for-write operation. Without a valid write token, users are *prohibited* from updating linked files.

You can grant a single privilege, or both privileges, to individual database users, a group of database users, or to all users of a particular database.

**Restriction:** You can only control the write privilege when a DB2 host table DATALINK column is set to WRITE PERMISSION ADMIN.

Table 7 shows all the possible write permission attributes of a DB2 host table DATALINK column, and compares the privileges that you can grant, deny, and revoke based on those attributes.

Table 7. DATALINK column attributes and applicable DLM security privileges

	Link privilege?	Write privilege?	Both Link and Write privileges?
WRITE PERMISSION ADMIN	Yes	Yes	Yes
WRITE PERMISSION FS	Yes	No	No
WRITE PERMISSION BLOCKED	Yes	No	No

**Important:** The execution of this command does not impact any *existing* connections to the Data Links server from the DB2 host. *Subsequent* connections will reflect all privilege changes.

**link privilege | write privilege | all privileges**

Specifies the privilege that you want to grant: link, write, or both privileges (**all privileges**).

*directory\_name*

Specifies a Data Links server directory. The directory must be on a registered Data Links File System (DLFS), within or beneath a registered prefix. The granted privilege or privileges apply to all files in a specified directory **and** any subdirectories.

You can also specify all directories with an asterisk ( \* ). On AIX and the Solaris Operating Environment, you must enter a backward slash ( \ ) before the asterisk to avoid UNIX command substitution.

**user | group | public**

Specifies the DB2 host users for whom you want to grant the privilege or privileges: a particular user (**user**), a user group (**group**), or all DB2 host users (**public**).

*db2\_authorization\_id*

Specifies the authorization id value of a user or group.

*db\_name*

The name of the remote DB2 UDB database.

*db\_inst* The name of the instance under which the database resides.

*server\_node\_name*

The DB2 server node name.

**Example:**

```
dlfm grant link privilege on dir /dlfs/test to user SHERRYG for
db EMP_TEST inst HRaccess node olympus.sanjose.bigco.com
```

Result: DB2 server user SHERRYG is allowed to link files in the /dlfs/test directory only when SHERRYG is linking those files to DB2 server database EMP\_TEST under the HRaccess instance on the olympus.sanjose.bigco.com node.

**Related concepts:**

- “Advanced file management security rules and guidelines” on page 101

**Related tasks:**

- “Approach 3 configuration tasks” on page 118

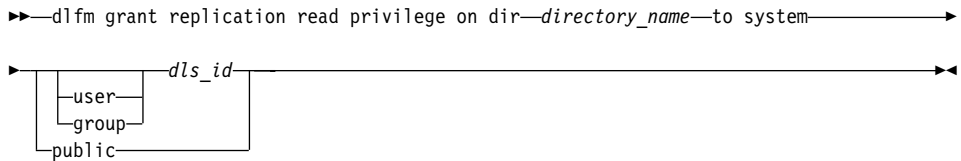
**Related reference:**

- “dlfm deny command” on page 186
- “dlfm revoke command” on page 203
- “dlfm set link security command” on page 206

---

## **dlfm grant replication read command**

**Requirement:** You must execute this command from the Data Links server using the Data Links Manager Administrator user ID.



This command enables you to support Replication users who require access to linked files to replicate linked data.

The **dlfm grant replication read privilege** command grants specific Data Links server system users (individuals, groups, or anyone who is authorized to access the server) the authority to access (read) all of the linked files in a specific directory on the Data Links server.

Requirement:

*directory\_name*

Specifies a Data Links server directory. The directory must be on a registered Data Links File System (DLFS), within a registered prefix (directory hierarchy). The granted privilege applies to all files in a specified directory **and** any subdirectories.

**user | group | public**

Specifies the Data Links server system users to whom you want to grant the authority: an individual user (**user**), a user group (**group**), or all registered Data Links server system users (**public**).

*dls\_id* Specifies the ID value of a Data Links server system user or group.

**Requirement:** The user or group must be local. On Windows, you cannot specify another domain user as the ID value.

Example:

- dlfm grant replication read privilege to system user mohans

**Related concepts:**

- “Basic linked file security controls” on page 97

**Related tasks:**

- “Replicating special data types” in the *Replication Guide and Reference*

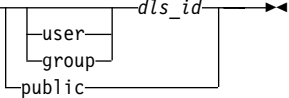
**Related reference:**

- “dlfm grant replication write command” on page 193
- “dlfm list registered replication access control command” on page 196
- “dlfm revoke replication command” on page 205

---

## dlfm grant replication write command

**Requirement:** You must execute this command from the Data Links server using the Data Links Manager Administrator user ID.

►►—dlfm grant replication write request privilege to system—

Grants specific Data Links server system users (individuals, groups, or anyone who is authorized to access the server) the authority to issue a file write request to the Data Links Manager Replication file-copy daemon, DLFM\_ASNCOPYD. DLFM\_ASNCOPYD communicates with the ASNDLCOPY exit routine to replicate linked data. ASNDLCOPY is part of the DB2 Data Propagator product.

**Important:** This command does not impact whether the specified users can actually create and write to files on the Data Links server. Data Links server users can create and write to files only if the native file system grants the appropriate authorities.

### **user | group | public**

Specifies the Data Links server system users to whom you want to grant the authority: an individual user (**user**), a user group (**group**), or all registered Data Links server system users (**public**).

*dls\_id* Specifies the ID value of a Data Links server system user or group.

**Requirement:** The user or group must be local. On Windows, you cannot specify another domain user as the ID value.

Example:

- dlfm grant replication write request privilege to system user mohans

### **Related concepts:**

- “Basic linked file security controls” on page 97

### **Related tasks:**

- “Replicating special data types” in the *Replication Guide and Reference*

### **Related reference:**

- “dlfm grant replication read command” on page 191
- “dlfm list registered replication access control command” on page 196
- “dlfm revoke replication command” on page 205

---

## dlfm help command

**Requirement:** You must execute this command from the Data Links server using the Data Links Manager Administrator user ID.

▶▶dlfm help◀◀

Displays the syntax and a brief description of all Data Links File Manager commands that apply on the current platform and environment. Same as the **dlfm ?** command.

To display command syntax only, use the **dlfm** command.

**Related reference:**

- “dlfm command” on page 183
- “dlfm ? command” on page 210

---

## dlfm list registered databases command

**Requirement:** You must execute this command from the Data Links server using the Data Links Manager Administrator user ID.

▶▶dlfm list registered databases◀◀

Lists all the databases registered with the Data Links File Manager.

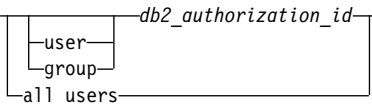
**Related tasks:**

- “Listing databases registered with the Data Links server” on page 74

---

## dlfm list registered directories command

**Requirement:** You must execute this command from the Data Links server using the Data Links Manager Administrator user ID.

▶▶dlfm list registered directories for db2\_authorization\_id◀◀

▶▶on db—db\_name—inst—db\_inst—node—server\_node◀◀



Lists all Data Links server directories and file access privileges (link, write, all, or none) for the specified user, user group, or for all privileged users on a particular DB2 database, database instance, and database server node.

**user | group | all users**

Specifies for whom you want to list directories and access privileges: a specific user (**user**), a user group (**group**), or all privileged users (**all users**).

*db2\_authorization\_id*

Specifies the authorization id value of a user or group.

*db\_name*

The name of the remote DB2 UDB database.

*db\_inst* The name of the instance under which the database resides.

*server\_node*

The DB2 server node name.

**Example:**

Issuing this command:

```
dlfm list registered directories for all users on db drawings inst acct2
node storage.ak1.bigco.com
```

might provide output similar to the following:

TYPE	AUTHID	PRIVILEGE	DIRECTORY
user	app1	write	/localfs/files/chips/
user	srgordon	link	/localfs/files/widgets/

**Related concepts:**

- “Advanced file management security rules and guidelines” on page 101

**Related tasks:**

- “Approach 3 maintenance tasks” on page 120

**Related reference:**

- “dlfm grant command” on page 189
- “dlfm set link security command” on page 206

---

## dlfm list registered prefixes command

**Requirement:** You must execute this command from the Data Links server using the Data Links Manager Administrator user ID.

▶—dlfm list registered prefixes—▶

Lists all prefixes within a Data Links File System (DLFS) that are registered with the Data Links File Manager.

### Related tasks:

- “Listing registered file systems (AIX, Solaris Operating Environment)” on page 68
- “Listing registered drives (Windows operating systems)” on page 68

### Related reference:

- “dlfm add\_prefix command” on page 184

---

## dlfm list registered replication access control command

**Requirement:** You must execute this command from the Data Links server using the Data Links Server Administrator user ID.

▶—dlfm list registered replication access control for—▶

▶—*dls\_id*—▶  
┌──system user──┐  
├──system group──┤  
└──all──┘

Lists the operations (read request, write request, or both requests) that Data Links server users can perform through the Data Links Manager Replication daemon, DLFM\_ASNCOPYD.

### system user | system group | all

Specifies for whom you want to display a list of operations: a particular Data Links server system user (**system user**), a Data Links server system user group (**system group**), or all Data Links server users (**all**).

*dls\_id* Specifies the id value of a Data Links server system user or group.

### Example:

Issuing this command:

dlfm list registered replication access control for all

might produce output similar to the following:

TYPE	ID	ACCESS	DIRECTORY
group	system	read	/localfs/files/chips/
public	--	read	/localfs/files/widgets/
user	app1	write	/localfs/files/chips/

**Related tasks:**

- “Replicating special data types” in the *Replication Guide and Reference*

**Related reference:**

- “dlfm grant replication read command” on page 191
- “dlfm grant replication write command” on page 193

---

## dlfm list registered users command

**Requirement:** You must execute this command from the Data Links server using the Data Links Manager Administrator user ID.

►—dlfm list registered users for directory—*dir\_name*—————►  
►—on db—*db\_name*—inst—*db\_inst*—node—*server\_node*—————►◀

Lists all users on a particular DB2 database, database instance, and database server node and their file access privileges (link, write, all, or none) for a specific Data Links server directory.

*dir\_name*

Specifies the Data Links server directory for which you want to list all users and their file access privileges.

*db\_name*

The name of the remote DB2 UDB database.

*db\_inst* The name of the instance under which the database resides.

*server\_node*

The DB2 server node name.

**Example:**

Issuing this command:

```
dlfm list registered users for directory /localfs/dbfiles/photos on db
employee inst acct1 node storage.ca2.bigco.com
```

might produce output similar to the following:

TYPE	AUTHID	PRIVILEGE
user	JDOE	write
user	MOHAN	link

**Related concepts:**

- “Advanced file management security rules and guidelines” on page 101

**Related tasks:**

- “Approach 3 maintenance tasks” on page 120

**Related reference:**

- “dlfm grant command” on page 189
- “dlfm set link security command” on page 206

---

## dlfm list upd\_in\_progress files for db command

**Requirement:** You must execute this command from the Data Links server using the Data Links Manager Administrator user ID.

```
▶—dlfm list upd_in_progress files for—————▶
▶—db—db_name—inst—db_inst—node—server_node—————▶
   └tid—table_id┘
▶—┬tsid—tablespace_id┘ ┬user—dlsuser_name┘—————▶
```

Displays all linked files that are currently in an update in progress state for a particular database and, optionally, for a particular table, table space, and Data Links server user. The command output also shows the write token values for each linked file.

**Attention:** The token values shown in the output might be invalid if the **dlfm refresh key** command has been executed just before or anytime after you invoke the **dlfm list upd\_in\_progress files for db** command.

*db\_name*

The name of the remote DB2 UDB database.

*db\_inst*

The name of the instance under which the database resides.

*server\_node*

The DB2 server node name.

*table\_id*

This optional parameter enables you to specify the ID number of the table for which you want to display all linked files that are currently in an update-in-progress state.

You can obtain the table ID number from the TABLEID column of the DB2 catalog table, SYSCAT.TABLES, on the corresponding DB2 server.

*tablespace\_id*

This optional parameter enables you to specify the number of the table space for which you want to display all linked files that are currently in an update-in-progress state.

You can obtain the table space ID number from the TBSPACEID column of the DB2 catalog table, SYSCAT.TABLES, on the corresponding DB2 server.

*dlsuser\_name*

This optional parameter enables you to specify the name of a particular Data Links server user. Important: This should be the value of a Data Links server user, not an authid from the DB2 host server.

Example:

- `dlfm list upd_in_progress files for db EMPLOYEE inst DB2 node storage.ca2.bigco.com tid 10 tsid 5 user joshua`

**Related concepts:**

- “Advanced file management security rules and guidelines” on page 101
- “Approach 3: Update linked files/customize file access privileges” on page 116

**Related tasks:**

- “Approach 3 maintenance tasks” on page 120
- “Recovering missing write tokens” on page 129

**Related reference:**

- “dlfm grant command” on page 189
- “dlfm refresh key command” on page 201

---

## **dlfm list upd\_in\_progress files for prefix command**

**Requirement:** You must execute this command from the Data Links server using the Data Links Manager Administrator user ID.

►—`dlfm list upd_in_progress files for prefix—prefix_name—pathname—pathname_pattern`—►

Displays all linked files that are currently in an update-in-progress state for a specific prefix. A *prefix* is the absolute path of a Data Links File System (DLFS) mount point (on UNIX) or a drive sharename (on Windows) under which linked files are stored.

The command output also shows the write token values for each linked file.

**Attention:** The token values shown in the output might be invalid if the **dlfm refresh key** command has been executed just before or anytime after you invoke the **dlfm list upd\_in\_progress files for prefixes** command.

*prefix\_name*

Specifies the prefix name.

To see a list of valid prefixes, use the **dlfm list registered prefixes** command.

*pathname\_pattern*

Specifies a matching expression for the rest of the pathname, except the prefix. An asterisk ( \* ) can be used to represent an arbitrary matching string in the pathname.

Example:

- `dlfm list upd_in_progress files for prefix /dlfs/test pathname "SELLIOT/*"`

Invoking this command will list all update-in-progress files under the directory `/dlfs/test/SELLIOT`, and under any of its subdirectories. Notice that the pathname pattern `"SELLIOT/*"` is enclosed in double quotes to avoid UNIX command substitution for the asterisk character.

**Related concepts:**

- “Advanced file management security rules and guidelines” on page 101
- “Approach 3: Update linked files/customize file access privileges” on page 116

**Related tasks:**

- “Approach 3 maintenance tasks” on page 120
- “Recovering missing write tokens” on page 129

**Related reference:**

- “dlfm grant command” on page 189

---

## dlfm refresh key command

**Requirement:** You must execute this command from the Data Links server using the Data Links Administrator user ID.

▶▶—dlfm refresh key—————▶▶

Refreshes the key used to generate access control tokens for DATALINK columns with the attributes READ PERMISSION DB and WRITE PERMISSION ADMIN. Invoking this command will invalidate all existing access control tokens, including any read and write tokens that are currently in use.

After issuing this command, execute the **dlfm restart** command to restart the Data Links Manager.

All applications that are connected to DB2 hosts that point to files on the Data Links server must disconnect before the new key will be used on those DB2 hosts. You can force the applications to disconnect from their DB2 databases using the DB2 FORCE APPLICATION command. Use the DB2 FORCE APPLICATION command only if it is appropriate to do so in your environment.

**Important:** If a file is in an update-in-progress state when this command is invoked, its assigned write token will become invalid and the update operation will be unable to proceed. The user or application can do either of the following should this situation occur:

- Commit the existing changes by issuing an SQL UPDATE statement that uses the DLNEWCOPY scalar function.
- If the file is referenced from a DATALINK column defined with the RECOVERY YES attribute, back out the existing changes by issuing an SQL UPDATE statement that uses the DLPREVIOUSCOPY scalar function.

**Related concepts:**

- “Approach 3: Update linked files/customize file access privileges” on page 116
- “Usage considerations for write tokens” on page 128

**Related reference:**

- “FORCE APPLICATION Command” in the *Command Reference*

---

## dlfm restart command

**Requirement:** You must execute this command from the Data Links server using the Data Links Administrator user ID.

►—dlfm restart—◄

Stops and then starts the Data Links File Manager. This command is equivalent to issuing the **dlfm stop** command followed by the **dlfm start** command.

**Related concepts:**

- “Basic operations: starting, stopping, and restarting the DLFM” on page 61

**Related tasks:**

- “Restarting the DLFM after an abnormal termination” on page 62

**Related reference:**

- “dlfm start command” on page 208
- “dlfm stop command” on page 209

---

## dlfm retrieve command

**Requirement:** You must execute this command from the Data Links server using the Data Links Manager Administrator user ID.

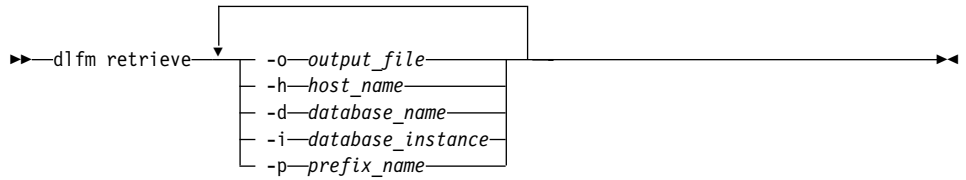
►—dlfm retrieve—◄

**Important:** This command is equivalent to the **retrieve\_query** command, which was documented in previous releases.

Displays the archival status of a specified set of files managed by the Data Links File Manager. This command initiates an interactive dialog that prompts you for the hostname, database name, instance name, and prefix name. Based on your selection criteria, this command then lists the status of all linked and unlinked files currently tracked by the Data Links File Manager.

Alternatively, you can bypass the interactive dialog and invoke this command with the parameters that you want directly from the command line, as follows.





**Example:**

```
dlfm retrieve -o result.txt -h dev1.lhuron.bigco.com -d SAMPLE -i
DB2INST -p /localdlfstest
```

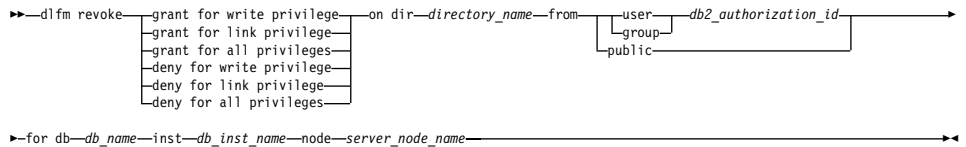
**Related tasks:**

- “Retrieving archive server backup file information” on page 93

**dlfm revoke command**

**Requirements:**

- You must execute this command from the Data Links server using the Data Links Server Administrator ID.
- For any link authorization changes to be effective, the file link security feature must be on (activated). You activate file link security with the **dlfm set link security on** command.



Enables you to revoke (remove) the effects of the **dlfm grant** or **dlfm deny** commands.

A **dlfm revoke** command only revokes a rule that exactly corresponds, parameter for parameter to the original **dlfm grant** or **dlfm deny** command.

**Important:** The **dlfm revoke** command goes into effect only when all existing DB2 host connections to the Data Links File Manager (DLFM) on the Data Links server are disconnected.

**grant for link privilege | grant for write privilege | grant for all privileges | deny for link privilege | deny for write privilege | deny for all privileges**

Specifies the command and the privilege or privileges that you want to revoke.

*directory\_name*

Specifies a Data Links server directory. The directory must be on a registered Data Links File System (DLFS), within or beneath a registered prefix. The revoked privilege or privileges apply to all files in a specified directory **and** any subdirectories.

You can also specify all directories with an asterisk ( \* ). On AIX and the Solaris Operating Environment, you must enter a backward slash ( \ ) before the asterisk to avoid UNIX command substitution.

**user | group | public**

Specifies the DB2 host users from whom you are revoking the specified command and the privileges: a particular user (**user**), a user group (**group**), or all DB2 host users (**public**).

*db2\_authorization\_id*

Specifies the authorization id value of a user or group.

*db\_name*

The name of the remote DB2 UDB database.

*db\_inst\_name*

The name of the instance under which the database resides.

*server\_node\_name*

The DB2 server node name.

**Example:**

```
dlfm revoke grant for link privilege on dir /dlfs/test from user SHERRYG
for db EMP_TEST inst HRaccess node olympus.sanjose.bigco.com
```

Result: DB2 server user SHERRYG is no longer allowed to link to files in the /dlfs/test directory, and all of its subdirectories, only when SHERRYG is attempting to link those files to DB2 server database EMP\_TEST under the HRaccess instance on the olympus.sanjose.bigco.com node.

**Related concepts:**

- “Advanced file management security rules and guidelines” on page 101

**Related reference:**

- “dlfm deny command” on page 186
- “dlfm grant command” on page 189
- “dlfm set link security command” on page 206

---

## dlfm revoke replication command

**Requirement:** You must execute this command from the Data Links server using the Data Links Server Administrator user ID.

```
▶▶—dlfm revoke replication—————▶▶
```

```
▶—read privilege on dir—directory_name—▶ from system—▶ user—▶ dls_id—▶▶
```

```
  |—write request privilege—▶                    |—group—▶
```

```
                                           |—public—▶
```

This command enables you to revoke (remove) the read privilege or the write request privilege for user accounts that no longer need to access linked files for replication purposes.

The **dlfm revoke replication command** can be used to revoke the authority to access (read) all of the linked files in a specific directory, or to revoke the authority to issue a file write request to the Data Links Manager Replication file-copy daemon, DLFM\_ASNCOPYD. You revoke the authorities from specific Data Links server user accounts (individuals, groups, or anyone who is authorized to access the server).

**Important:** You cannot revoke both the read privilege and the write request privilege for a particular user or group of users at the same time. You must issue the **dlfm revoke replication command** twice, specifying the particular privilege you want to remove each time.

### read privilege on dir | write request privilege

Specifies whether you want to revoke the read (access) privilege or the write request privilege.

#### *directory\_name*

Specifies a Data Links server directory. The directory must be on a registered Data Links File System (DLFS), within a registered prefix (directory hierarchy). This value is required only if the **read privilege on dir** option is specified.

### user | group | public

Specifies the Data Links server system users from whom you want to revoke the read or write request privilege: an individual user (**user**), a user group (**group**), or all registered Data Links server system users (**public**).

*dls\_id* Specifies the ID value of a Data Links server system user or group.

**Requirement:** The user or group must be local. On Windows, you cannot specify another domain user as the ID value.

Examples:

- `dlfm revoke replication write request privilege from system user dprop`
- `dlfm revoke replication read privilege on dir \ddrive\files\resumes from system group rplktor`

**Related tasks:**

- “Replicating special data types” in the *Replication Guide and Reference*

**Related reference:**

- “`dlfm grant replication read command`” on page 191
- “`dlfm grant replication write command`” on page 193
- “`dlfm list registered replication access control command`” on page 196

---

## **dlfm see command (AIX and Solaris Operating Environment)**

**Requirement:** You must execute this command from the Data Links server using the Data Links Administrator user ID.

►—dlfm see—◄

Displays all DLFM processes currently running on the system. Using this command is a simple way to check whether the DLFM is up and running.

**Related tasks:**

- “Monitoring the DLFM background processes (AIX, Solaris Operating Environment)” on page 63

---

## **dlfm set link security command**

**Requirement:** You must execute this command from the Data Links server using the Data Links Manager Administrator ID.

►—dlfm set link security— on  
 off—◄

Activates and deactivates the file link security feature for a Data Links server. File link security enables you to control who can link to files in specific directories, and from specific databases. You must restart the Data Links server using **dlfm restart** command for the changes to become effective.

**Attention:** By default, file link security is activated (on) when you install Data Links Manager.

- To grant linking privileges to the files on a Data Links server, use the **dlfm grant** command.
- To remove or deny linking privileges to the files on a Data Links server, use the **dlfm revoke** or **dlfm deny** commands.

**Important:** All file link security access privileges are stored on the Data Links server. If you deactivate the file link security feature, then reactivate it at a later time, all previous file link security access privileges also get reactivated.

**on | off**

Activates (**on**) or deactivates (**off**) file link security.

**Related concepts:**

- “Advanced file management security rules and guidelines” on page 101
- “Approach 3: Update linked files/customize file access privileges” on page 116

**Related reference:**

- “dlfm deny command” on page 186
- “dlfm drop\_dlm command” on page 189
- “dlfm grant command” on page 189
- “dlfm revoke command” on page 203

## dlfm setup command

**Requirement:** You must execute this command from the Data Links server using the Data Links Manager Administrator ID.

▶▶—dlfm setup—————▶▶

Sets up the DLFM\_DB database repository. This command is equivalent to issuing the following sequence of commands:

1. **dlfm startdbm**
2. **dlfm create\_db**
3. **dlfm create**
4. **dlfm stopdbm**

This command only needs to be run when installing or rebuilding a new Data Links File Manager.

The **dlfm setup** command is automatically invoked as part of the normal Data Links Manager installation process.

**Related reference:**

- “dlfm create command” on page 185
- “dlfm create\_db command” on page 186
- “dlfm stopdbm command” on page 210
- “dlfm startdbm command” on page 209

---

## dlfm shutdown command

**Requirement:** You must execute this command from the Data Links server using the Data Links Manager Administrator user ID.

►—dlfm shutdown—◄◄

Stops the Data Links File Manager and removes all Inter Process Communication (IPC) resources. This command attempts a clean shutdown of the DLFM. If a clean shutdown is not possible, this command kills the DLFM processes. A full shutdown might take from 30 to 40 seconds to complete.

**Attention:** If you are running other processes under this same account, those processes will probably terminate. This command removes *all* shared resources, including all IPCs, for the account that you are using (in this situation, the Data Links Manager Administrator account).

**Related concepts:**

- “Basic operations: starting, stopping, and restarting the DLFM” on page 61

**Related tasks:**

- “Restarting the DLFM after an abnormal termination” on page 62

---

## dlfm start command

**Requirement:** You must execute this command from the Data Links server using the Data Links Manager Administrator user ID.

►—dlfm start—◄◄

Starts the Data Links File Manager. Execute the **dlfm setup** (automatically run during install), **dlfm add\_db** and **dlfm add\_prefix** commands *before* executing **dlfm start**. Issue the **dlfm see** command after executing **dlfm start** to verify that the DLFM processes are running.

**Windows Operating Systems only:** If you have DLFM started as a service, use the Services panel, which is accessed from the Control Panel, to start the DLFM service.

**Related concepts:**

- “Basic operations: starting, stopping, and restarting the DLFM” on page 61

**Related reference:**

- “dlfm see command (AIX and Solaris Operating Environment)” on page 206

---

## dlfm startdbm command

**Requirement:** You must execute this command from the Data Links server using the Data Links Manager Administrator user ID.

▶▶—dlfm startdbm—▶▶

Starts the DB2 database manager for DLFM’s private database, DLFM\_DB. This command is the same as the DB2 Universal Database **db2start** command.

**Related concepts:**

- “Basic operations: starting, stopping, and restarting the DLFM” on page 61

**Related reference:**

- “db2start - Start DB2 Command” in the *Command Reference*

---

## dlfm stop command

**Requirement:** You must execute this command from the Data Links server using the Data Links Manager Administrator user ID.

▶▶—dlfm stop—▶▶

Stops the Data Links File Manager. This command ends all processes that comprise the Data Links File Manager.

Use the **dlfm see** command to verify that all DLFM processes have ended. If some processes still remain after about 30 seconds, you can use the **dlfm shutdown** command to ensure that the DLFM is completely terminated.

The **dlfm stop** command might not always remove all IPC resources used by the Data Links File Manager. Use the **dlfm shutdown** command to ensure that all IPCs are removed.

**Windows Operating Systems only:** If you have DLFM started as a service, use the Services panel, which is accessed from the Control Panel, to stop the DLFM service.

**Related concepts:**

- “Basic operations: starting, stopping, and restarting the DLFM” on page 61

**Related reference:**

- “dlfm see command (AIX and Solaris Operating Environment)” on page 206
- “dlfm shutdown command” on page 208

---

## dlfm stopdbm command

**Requirement:** You must execute this command from the Data Links server using the Data Links Manager Administrator user ID.

▶—dlfm stop—▶

Stops the DB2 database manager for DLFM’s private database, DLFM\_DB. This command is the same as the DB2 Universal Database **db2stop** command.

**Attention:** Do *not* execute the **dlfm stopdbm** command while the Data Links File Manager is running.

**Related concepts:**

- “Basic operations: starting, stopping, and restarting the DLFM” on page 61

**Related reference:**

- “db2stop - Stop DB2 Command” in the *Command Reference*

---

## dlfm ? command

**Requirement:** You must execute this command from the Data Links server using the Data Links Manager Administrator user ID.

▶—dlfm ?—▶



Displays the syntax and a brief description of all Data Links File Manager commands that apply on the current platform and environment. Same as the **dlfm help** command.

To display command syntax only, use the **dlfm** command.

**Related reference:**

- “dlfm command” on page 183
- “dlfm help command” on page 194



---

## Appendix B. Data Links Filesystem Filter commands

This appendix lists the Data Links Filesystem Filter (DLFF) commands, and describes the purpose and syntax of each command. These commands apply to Windows Operating systems only.

---

### dlff add command (Windows operating system)

**Requirement:** You must execute this command from the Data Links server using the DLFM administrator user ID.

This command applies to Windows NT and Windows 2000 operating systems only.

►—dlff add—*drive letter:*—►

Places a file system under control of the Data Links Filesystem Filter (DLFF). Run this command before issuing a **dlfm add\_prefix** command. The **dlfm add\_prefix** command registers the file system with the DLFM (Data Links File Manager) on a Data Links server.

If a drive that is being added does not exist, or is not of a supported file system type, the command will generate an error message.

*drive letter:*

The letter of the drive that you want to place under the control of the DLFF. Include a colon after the drive letter value. You can specify multiple drive letters, separated by semicolons ( ; ).

Examples:

- dlff add d:
- dlff add d::e:

### Error messages

The following error messages can occur when invoking this command.

Drive x: is not a DLFS supported type. Failing to add drive x:

**Cause:** This message gets displayed at a command prompt when an attempt is made to add a drive that is not of the NTFS file system type. *x* stands for the drive letter in error.

**Action:** Ensure that the drive you want to add is an NTFS-formatted drive. If the drive is not an NTFS-formatted drive, you can either convert an existing file system to NTFS, or create a new NTFS partition.

- To convert an existing file system to NTFS, enter the following command: `convert x:/fs:ntfs`, where `x:` is the drive that you want to convert to NTFS. Run this command from a command prompt window on a drive other than the one that you are converting to NTFS.
- To create an NTFS partition, use the Windows NT Disk Administrator tool or Windows 2000 Disk Management.

Drive `x:` is already in the list

**Cause:** This message gets displayed at a command prompt when an attempt is made to add a drive that is already under DLFF control. `x:` stands for the drive letter in error.

**Action:** Ensure that the drive you want to add is really under DLFF control by invoking the **dlff list** command.

**Related tasks:**

- “Registering a drive with DLFF (Windows operating systems)” on page 66

**Related reference:**

- “dlfm add\_prefix command” on page 184
- “dlff list command (Windows operating system)” on page 215

---

## dlff get dlfmaccount command (Windows operating system)

**Requirement:** You must execute this command from the Data Links server using the Data Links Manager Administrator user ID.

This command applies to Windows NT and Windows 2000 operating systems only.

►—dlff get dlfmaccount—◄

Use this command to obtain the current `dlfmuser` account name for the Data Links Manager. Use the **dlff set dlfmaccount** command to change the `dlfmuser` account name.

Example:

- `dlff get dlfmaccount`

**Related reference:**

- “dlff set dlfmaccount command (Windows operating system)” on page 219

---

## **dlff get loglevel command (Windows operating system)**

**Requirement:** You must execute this command from the Data Links server using the Data Links Manager Administrator user ID.

This command applies to Windows NT and Windows 2000 operating systems only.

▶—dlff get loglevel—————▶

Use this command to get the current message severity level for logs generated by the DLFS. The different levels are as follows:

- 0 - success
- 1- informational
- 2 - warning
- 3 - error

Use the **dlff set loglevel** command to change the current message severity level.

Example:

- dlff get loglevel

**Related tasks:**

- “Modifying logging for Data Links Filesystem Filter (DLFF) processing (Windows operating systems)” on page 80

**Related reference:**

- “dlff set loglevel command (Windows operating system)” on page 221

---

## **dlff list command (Windows operating system)**

**Requirement:** You must execute this command from the Data Links server using the DLFM administrator user ID.

This command applies to Windows NT and Windows 2000 operating systems only.

►—dlff list—◄

Lists all drives under the control of the Data Links Filesystem Filter (DLFF).

If no drives are under DLFF control NIL displays.

Example:

- dlff list

**Related tasks:**

- “Listing registered drives (Windows operating systems)” on page 68

**Related reference:**

- “dlff add command (Windows operating system)” on page 213
- “dlfm list registered prefixes command” on page 196

---

## **dlff refreshtrace command (Windows operating system)**

**Requirement:** You must execute this command from the Data Links server using the Data Links Manager Administrator user ID.

This command applies to Windows NT and Windows 2000 operating systems only.

►—dlff refreshtrace—◄

Use this command to refresh trace settings for all drives under the control of a Data Links Filesystem Filter (DLFF).

You invoke this command after resetting the message severity level with the **dlff set loglevel** command.

Example:

- dlff refreshtrace

**Related tasks:**

- “Modifying logging for Data Links Filesystem Filter (DLFF) processing (Windows operating systems)” on page 80

**Related reference:**

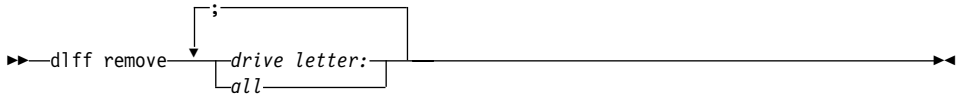
- “dlff get loglevel command (Windows operating system)” on page 215
- “dlff set loglevel command (Windows operating system)” on page 221

---

## dlff remove command (Windows operating system)

**Requirement:** You must execute this command from the Data Links server using the DLFM administrator user ID.

This command applies to Windows NT and Windows 2000 operating systems only.



Removes drives from Data Links Filesystem Filter (DLFF) control.

After invoking this command, you must reboot the system for the changes to take effect.

*drive letter:*

The letter of the drive from which you want to remove the DLFF. Include a colon after the drive letter value. You can specify multiple drive letters, separated by semicolons ( ; ).

*all* Use this optional keyword to remove *all* drives that are currently under DLFF control in one operation. Do not specify drive letters if you use this keyword.

After the remove all operation completes, the message No drives in the list LogicalDrives = (NIL) displays.

Examples:

- dlff remove d:
- dlff remove d;;e:
- dlff remove all

### Error messages

The following error message can occur when invoking this command.

Drive x: is not in the list

**Cause:** This message gets displayed at a command prompt when an attempt is made to remove a drive from under DLFF control, but that drive is not actually under DLFF control. *x:* stands for the drive letter in error.

**Action:** Ensure that the drive you want to remove is really under DLFF control by invoking the **dlff list** command.

**Related tasks:**

- “Registering a drive with DLFF (Windows operating systems)” on page 66
- “Registering, querying and de-registering a DLFF (Windows operating system)” on page 71

**Related reference:**

- “dlff list command (Windows operating system)” on page 215

---

**dlff set dlfm\_write\_group command (Windows operating system)**

**Requirement:** You must execute this command from the Data Links server using the Data Links Manager Administrator user ID.

This command applies to Windows NT and Windows 2000 operating systems only.

► dlff set dlfm\_write\_group *dlfmgroupname* ◄

Use this command to notify the DLFF if you have changed special write group to which the dlfmuser account belongs.

Run this command *after* you have actually changed the group membership of the dlfmuser account in Windows.

After you run **dlff set dlfm\_write\_group** command, you must either run the **dlff refreshtrace** command, or reboot the machine for the changes to take effect.

**Attention:** If you want to change the group in which the dlfmuser account is located, you must make the change *before* you start linking files. Once you have begun linking files, you cannot *ever* change the group where the dlfmuser account belongs.

*dlfmgroupname*

The name of the special write group to which the dlfmuser account now belongs.

Example:

- dlff set dlfm\_write\_group dlfxgrp

**Related concepts:**

- “Before you install DB2 Data Links Manager (Windows)” in the *Quick Beginnings for Data Links Manager*



**Related tasks:**

- “Installing DB2 Data Links Manager (Windows)” in the *Quick Beginnings for Data Links Manager*

**Related reference:**

- “dlff refreshtrace command (Windows operating system)” on page 216
- “dlff set dlfmaccount command (Windows operating system)” on page 219

---

## dlff set dlfmaccount command (Windows operating system)

**Requirement:** You must execute this command from the Data Links server using the Data Links Manager Administrator user ID.

This command applies to Windows NT and Windows 2000 operating systems only.

```
►► dlff set dlfmaccount [account_name]
```

Use this command to notify the DLFF of a password or name change for the dlfmuser account. Run this command *after* you have actually changed the dlfmuser account password or name value in Windows.

**Attention:** Do not *ever* change the name of the dlfmuser account after you have started linking files. You run the risk of making *all* linked files inaccessible to the new dlfmuser account.

If you attempt to change the dlfmuser account name with this command, a warning message displays, alerting you to the risks of making the name change. You will be prompted to acknowledge these risks before the command runs. After you change the dlfmuser account name, you must either run the **dlff refreshtrace** command, or reboot the system for the change to take effect.

If you are just changing the password, changes go into effect immediately after running the command.

*account\_name*

An optional parameter that specifies the changed name of the dlfmuser account. You only need to specify this value if you want to change the name of the dlfmuser account. You will be prompted to input the account’s password.

If you only want to change the dlfmuser account password, do not specify this parameter. You will be prompted to input the new

password value. If DLFF cannot locate the dlfmaccount name for some reason, you will be prompted to input the name.

Examples:

- `dlfm set dlfmaccount`
- `dlfm set dlfmaccount dlfm`

## Error messages

The following error messages can occur when invoking this command.

LookupAccountName failed for xxxx: No mapping between account names and security IDs was done. Failed to set parameter DlfmAccountName to xxxx. Dlfscfgset failed for parameter dlfmaccount.

**Cause:** This message gets displayed at a command prompt when an attempt is made to set an account name, but the operation failed. xxxx stands for the account name value in error.

**Action:** Ensure that the the dlfmuser account name being set exists as a user name on the system.

LogOnUser failed: Logon failure: unknown user name or bad password. Maximum Retries Exceeded the limit - quitting. Failed to set parameter DlfmAccountName to xxxx. Dlfscfgset failed for parameter dlfmaccount.

**Cause:** This message gets displayed at a command prompt when an attempt is made to set an account name, but the password supplied for the account at the command prompt is incorrect. xxxx stands for the account name value in error.

**Action:** Ensure that the dlfmaccount itself has the same password as the one you supply at the command prompt.

### Related concepts:

- “Before you install DB2 Data Links Manager (Windows)” in the *Quick Beginnings for Data Links Manager*

### Related tasks:

- “Installing DB2 Data Links Manager (Windows)” in the *Quick Beginnings for Data Links Manager*

### Related reference:

- “dlff refreshtrace command (Windows operating system)” on page 216
- “dlff set dlfm\_write\_group command (Windows operating system)” on page 218

---

## dlff set loglevel command (Windows operating system)

**Requirement:** You must execute this command from the Data Links server using the Data Links Manager Administrator user ID.

This command applies to Windows NT and Windows 2000 operating systems only.

►►—dlff set loglevel—*log\_level*—————►►

Use this command to set the message severity level for message logs generated by DLFF. The messages get placed into the Windows system log.

After you reset the message severity level with this command, you must run the **dlff refreshtrace** command for the changes to take effect.

*log\_level*

A number between 0 and 3 that specifies the message severity level of the errors you want to log. The different levels are as follows:

- 0 - success
- 1- informational
- 2 - warning
- 3 - error

Example:

- dlff set loglevel 1

### Related tasks:

- “Modifying logging for Data Links Filesystem Filter (DLFF) processing (Windows operating systems)” on page 80

### Related reference:

- “dlff get loglevel command (Windows operating system)” on page 215
- “dlff refreshtrace command (Windows operating system)” on page 216



---

## Appendix C. Data Links File System (DLFS) errors

This appendix contains a list of the potential error messages that are issued by the Data Links Filesystem Filter (DLFF). The error messages display through your operating environment's file system. For each command or operation that could result in an error message, this appendix lists:

- The error message text that displays
- The possible causes of the error message
- The suggested actions for you to take

The appendix contains the following sections:

- "DLFS errors from commands (AIX)"
- "DLFS errors from commands (Solaris Operating Environment)" on page 229
- "DLFS errors from commands (Windows)" on page 236
- "DLFS errors from files referenced from columns with WRITE PERMISSION ADMIN" on page 240

---

### DLFS errors from commands (AIX)

This topic contains the potential error messages that you can receive on AIX-based file systems as a result of Data Links Filesystem Filter (DLFF) actions.

The AIX commands that can result in errors are organized into two sections:

- "DLFS errors from specific AIX commands"
- "DLFS errors from AIX commands that open a file" on page 226

#### DLFS errors from specific AIX commands

This section contains information about error messages that you might receive from the Data Links Filesystem Filter (DLFF) that can result from certain AIX commands. It contains an alphabetical listing of the common commands. After each command name, this section lists:

- The error message text that displays
- The possible causes of the error message
- The suggested actions for you to take

This section lists the following commands:

- "chmod nnn filename command" on page 224

- “ls filename command”
- “mkdir command” on page 225
- “mount command” on page 225
- “mv dir11 newdir11 command” on page 225
- “mv filename newfilename command” on page 225
- “rm filename command” on page 225
- “strload command” on page 226

### **chmod nnn filename command**

*If your file is linked to a DATALINK column that is defined with READ PERMISSION DB:*

chmod: filename: The file access permissions do not allow the specified action.

**Cause:** This is a system message that displays when you attempt to change the attributes of a file that is linked under Data Links Manager control as a READ PERMISSION DB table column.

**Action:** No action required. Data Links Manager does not allow non-root users to change the attributes of files that are linked under its control as READ PERMISSION DB table columns.

*If your file is linked to a DATALINK column that is defined with READ PERMISSION FS and WRITE PERMISSION BLOCKED:*

chmod: filename: Operation not permitted.

**Cause:** This is a system message that displays when you attempt to change the attributes of a file that is linked under Data Links Manager control as a READ PERMISSION FS and WRITE PERMISSION BLOCKED table column.

**Action:** No action required. Data Links Manager does not allow non-root users to change the attributes of files that are linked under its control as WRITE PERMISSION BLOCKED table columns.

### **ls filename command**

ls: 0653-345 filename: Permission denied.

**Cause:** This is a system message that displays when you attempt to list a file that is linked under Data Links Manager control from a READ PERMISSION DB table column.

**Action:** Data Links Manager does not allow non-root users to list files that are linked under Data Links Manager control from READ

PERMISSION DB table columns. To see the listing (or see its attributes), get a valid token from the database, and then re-issue the command.

### **mkdir command**

0653-358 Cannot create "...". "...": The specified device does not exist.

**Cause:** This is a system message that displays when you attempt to create a directory in a Data Links File System (DLFS), but the Data Links File Manager (DLFM) is not started.

**Action:** Ensure that the DLFM is running by issuing the **dlfm see** command. If the DLFM is not running, start it by issuing the **dlfm start** command, and then try creating the directory again.

### **mount command**

dlfs mount Error: Function not implemented dlfs mount helper: Mount Unsuccessful Unmount the base file system

**Cause:** The Data Links File System (DLFS) driver (dlfsdrv) must be loaded when you try to mount a DLFS.

**Action:** Ensure that the dlfsdrv is loaded.

### **mv dir11 newdir11 command**

mv: 0653-401 Cannot rename dir11 to newdir11: Operation not permitted.

**Cause:** This is a system message that displays when you attempt to rename a directory in a Data Links File System (DLFS).

**Action:** No action required. Data Links Manager does not allow you to rename a directory in a DLFS.

### **mv filename newfilename command**

mv: 0653-401 Cannot rename filename to newfilename: The file access permissions do not allow the specified action.

**Cause:** This is a system message that displays when you attempt to rename a file that is linked under Data Links Manager control.

**Action:** No action required. Data Links Manager does not allow non-root users to rename files that are under its control.

### **rm filename command**

rm: filename: Cannot remove filename. The file access permissions do not allow the specified action.

**Cause:** This is a system message that displays when you attempt to remove a file that is linked under Data Links Manager control.

**Action:** No action required. Data Links Manager does not allow non-root users to remove files that are under its control.

**strload command**

strload: cannot terminate dlfsdrv: The requested resource is busy.

**Cause:** Either Data Links File Manager (DLFM) is running, or a Data Links File System (DLFS) is mounted and is using this driver.

**Action:** Stop DLFM if it is running, and then try to unload the driver. If that does not work, check to see whether any mounted DLFS is already using this driver. Unmount that DLFS and try to unload the driver again.

**DLFS errors from AIX commands that open a file**

This section contains information about error messages that you might receive from the Data Links Filesystem Filter (DLFF) when you issue a command to open a file that is linked to a DATALINK column that is defined with READ PERMISSION DB. These errors only apply for non-root users. They are categorized by the types of files that you can open. Under each type of file, the section lists:

- Examples of common commands that you might use to open each type of file
- The error message text that displays
- The possible causes of the error message
- The suggested actions for you to take

This section lists the following types of open operations:

- “Opening a file without a token”
- “Opening a file with an expired token” on page 227
- “Opening a file with an invalid token” on page 228

**Opening a file without a token**

For a file that is linked to a DATALINK column that is defined with READ PERMISSION DB, the DLFF can issue many error messages when you open a file without a token. Table 8 on page 227 contains a partial list of AIX commands that you might use to open a file without a token, and the error message text that is associated with each command.

The list in Table 8 on page 227 does not include all of the possible commands that you can use in an open operation. The commands that are listed represent the types of commands that you might use when you open the file.



Table 8. Partial list of AIX commands that you might use to open a file without a token and the associated error message.

Command name	Error message
<code>cat filename</code>	cat: 0652-050 Cannot open filename.
<code>cp filename newfilename</code>	cp: filename: The file access permissions do not allow the specified action.
<code>diff filename filenameefs</code>	diff: filename: The file access permissions do not allow the specified action.
<code>grep "str" filename</code>	grep: 0652-033 Cannot open filename.
<code>head filename</code>	filename: The file access permissions do not allow the specified action.
<code>tail filename</code>	filename: The file access permissions do not allow the specified action.

When you issue a command to open a file without a token, the returned error message has the following cause and suggested action:

**Cause:** This is a system message that displays when you attempt to read the contents of a file that is linked under Data Links Manager as a READ PERMISSION DB table column.

**Action:** To read the contents of the file, get a valid token from the database, and then re-issue the command.

### Opening a file with an expired token

For a file that is linked to a DATALINK column that is defined with READ PERMISSION DB, the DLFF can issue many error messages when you open a file with an expired token. Table 9 contains a partial list of AIX commands that you might use to open a file with an expired token, and the error message text that is associated with each command.

The list in Table 9 does not include all of the possible commands that you can use in an open operation. The commands that are listed represent the types of commands that you might use when you open the file.

Table 9. Partial list of AIX commands that you might use to open a file with an expired token and the associated error message.

Command name	Error message
<code>cat "04E2_DEGnck_JD8hFHOy6JU;filename"</code>	cat: 0652-050 Cannot open 04E2_DEGnck_JD8hFHOy6JU;filename.
<code>cp "04E2_DEGnck_JD8hFHOy6JU;filename" newfilename</code>	cp: 04E2_DEGnck_JD8hFHOy6JU;filename: The file access permissions do not allow the specified action.

Table 9. Partial list of AIX commands that you might use to open a file with an expired token and the associated error message. (continued)

Command name	Error message
<code>diff "04E2_DEGnck__JD8hFHOy6JU;filename" filenamefs</code>	diff: 04E2_DEGnck__JD8hFHOy6JU;filename: The file access permissions do not allow the specified action.
<code>grep "str" "04E2_DEGnck__JD8hFHOy6JU;filename"</code>	grep: 0652-033 Cannot open 04E2_DEGnck__JD8hFHOy6JU;filename.
<code>head "04E2_DEGnck__JD8hFHOy6JU;filename"</code>	04E2_DEGnck__JD8hFHOy6JU;filename: The file access permissions do not allow the specified action.
<code>more "04E2_DEGnck__JD8hFHOy6JU;filename"</code>	04E2_DEGnck__JD8hFHOy6JU;filename: The file access permissions do not allow the specified action.
<code>tail "04E2_DEGnck__JD8hFHOy6JU;filename"</code>	04E2_DEGnck__JD8hFHOy6JU;filename: The file access permissions do not allow the specified action.

When you issue a command to open a file with an expired token, the returned error message has the following cause and suggested action:

**Cause:** This is a system message that displays when you attempt to read the contents of a file that is linked under Data Links Manager control as a READ PERMISSION DB table column, and the token expired.

**Action:** Get a valid token from the database, and then re-issue the command.

### Opening a file with an invalid token

For a file that is linked to a DATALINK column that is defined with READ PERMISSION DB, the DLFF can issue many error messages when you open a file with an invalid token. Table 10 on page 229 contains a partial list of AIX commands that you might use to open a file with an invalid token, and the error message text that is associated with each command.

The list in Table 10 on page 229 does not include all of the possible commands that you can use in an open operation. The commands that are listed represent the types of commands that you might use when you open the file.

Table 10. Partial list of AIX commands that you might use to open a file with an invalid token and the associated error message.

Command name	Error message
cat "04E2_AAAAAAAAAAAAAAH0y6JU;filename"	cat: 0652-050 Cannot open 04E2_AAAAAAAAAAAAAAH0y6JU;filename.
cp "04E2_AAAAAAAAAAAAAAH0y6JU;filename" newfilename	cp: 04E2_AAAAAAAAAAAAAAH0y6JU;filename: A file or directory in the path name does not exist.
diff "04E2_AAAAAAAAAAAAAAH0y6JU;filename" filenamefs	diff: 04E2_AAAAAAAAAAAAAAH0y6JU;filename: A file or directory in the path name does not exist.
grep "str" "04E2_AAAAAAAAAAAAAAH0y6JU;filename"	grep: 0652-033 Cannot open 04E2_AAAAAAAAAAAAAAH0y6JU;filename.
head "04E2_AAAAAAAAAAAAAAH0y6JU;filename"	04E2_AAAAAAAAAAAAAAH0y6JU;filename: A file or directory in the path name does not exist.
more "04E2_AAAAAAAAAAAAAAH0y6JU;filename"	04E2_AAAAAAAAAAAAAAH0y6JU;filename: A file or directory in the path name does not exist.
tail "04E2_AAAAAAAAAAAAAAH0y6JU;filename"	04E2_AAAAAAAAAAAAAAH0y6JU;filename: A file or directory in the path name does not exist.

When you issue a command to open a file with an invalid token, the returned error message has the following cause and suggested action:

**Cause:** This is a system message that displays when you attempt to read the contents of a file that is linked under Data Links Manager control as a READ PERMISSION DB table column, and the token is invalid.

**Action:** Get a valid token from the database, and then re-issue the command.

**Related reference:**

- "DLFS errors from files referenced from columns with WRITE PERMISSION ADMIN" on page 240

---

## DLFS errors from commands (Solaris Operating Environment)

This topic contains the potential error messages that you can receive on Solaris Operating Environment file systems as a result of Data Links Filesystem Filter (DLFF) actions.

The commands in the Solaris Operating Environment that can result in errors are organized into two sections:

- “DLFS errors from specific commands in the Solaris Operating Environment”
- “DLFS errors from commands in the Solaris Operating Environment that open a file” on page 233

## **DLFS errors from specific commands in the Solaris Operating Environment**

This section contains information about error messages that you might receive from the Data Links Filesystem Filter (DLFF) that can result from certain commands that you can use in the Solaris Operating Environment. It contains an alphabetical listing of the common commands. After each command name, this section lists:

- The error message text that displays
- The possible causes of the error message
- The suggested actions for you to take

This section lists the following commands:

- “chmod nnn filename command”
- “ls filename command” on page 231
- “mkdir command” on page 231
- “mount command” on page 231
- “mvdire dir11 newdir11” on page 231
- “mv filename newfilename command” on page 232
- “rem\_drv command” on page 232
- “rm filename command” on page 232

### **chmod nnn filename command**

*If your file is linked to a DATALINK column that is defined with READ PERMISSION DB:*

chmod: WARNING: can't access filename

**Cause:** This is a system message that displays when you attempt to change the attributes of a file that is linked under Data Links Manager control as a READ PERMISSION DB table column.

**Action:** No action required. Data Links Manager does not allow non-root users to change the attributes of files that are linked under Data Links Manager control as READ PERMISSION DB table columns.

*If your file is linked to a DATALINK column that is defined with READ PERMISSION FS and WRITE PERMISSION BLOCKED:*

chmod: WARNING: can't change filename

**Cause:** This is a system message that displays when you attempt to change the attributes of a file that is linked under Data Links Manager control as a READ PERMISSION FS and WRITE PERMISSION BLOCKED table column.

**Action:** No action required. Data Links Manager does not allow non-root users to change the attributes of files that are linked under Data Links Manager control as WRITE PERMISSION BLOCKED table columns.

### **ls filename command**

filename: Permission denied

**Cause:** This is a system message that displays when you attempt to list a file that is linked under Data Links Manager control from a READ PERMISSION DB table column.

**Action:** Data Links Manager does not allow non-root users to list files that are linked under Data Links Manager control from READ PERMISSION DB table columns. Check if the Data Links File Manager (DLFM) is running by issuing the **dlfm see** command. If the DLFM is not running, start it by issuing the **dlfm start** command, and then try creating the directory again.

### **mkdir command**

Failed to make directory "..."; No such device

**Cause:** This is a system message that displays when you attempt to create a directory in a Data Links File System (DLFS), but the Data Links File Manager (DLFM) is not started.

**Action:** Ensure that the DLFM is running by issuing the **dlfm see** command. If the DLFM is not running, start it by issuing the **dlfm start** command, and then try creating the directory again.

### **mount command**

dlfs mount Error : Invalid argument dlfs mount helper: Mount Unsuccessful  
Unmount the base file system /usr/sbin/umount /dlfstest umount: warning:  
/dlfstest not in mnttab

**Cause:** The Data Links File System (DLFS) driver (dlfsdrv) must be loaded when you try to mount a DLFS.

**Action:** Ensure that the dlfsdrv is loaded.

### **mkdir dir11 newdir11**

mv: dir11 is a directory

- Cause:** This is a system message that displays when you attempt to rename a directory in a Data Links File System (DLFS).
- Action:** No action required. Data Links Manager does not allow you to rename a directory in a DLFS.

**mv filename newfilename command**

*If your file is linked to a DATALINK column that is defined with READ PERMISSION DB:*

mv: cannot access filename

- Cause:** This is a system message that displays when you attempt to rename a file that is linked under Data Links Manager control as a READ PERMISSION DB table column.
- Action:** No action required. Data Links Manager does not allow non-root users to rename files that are under its control.

*If your file is linked to a DATALINK column that is defined with READ PERMISSION FS and WRITE PERMISSION BLOCKED:*

mv: cannot rename filename: Permission denied

- Cause:** This is a system message that displays when you attempt to rename a file that is linked under Data Links Manager control as a READ PERMISSION FS and WRITE PERMISSION BLOCKED table column.
- Action:** No action required. Data Links Manager does not allow non-root users to rename files that are under its control.

**rem\_drv command**

Device busy Cannot unload module: dlfsdrv Will be unloaded upon reboot.

- Cause:** Either Data Links File Manager (DLFM) is running, or a Data Links File System (DLFS) is mounted and is using this driver.
- Action:** Stop DLFM if it is running, and then try to unload the driver. If that does not work, check to see whether any mounted DLFS is already using this driver. Unmount that DLFS and try to unload the driver again.

**rm filename command**

rm: filename: override protection 444 (yes/no)? y

rm: filename not removed: Permission denied

- Cause:** This is a system message that displays when you attempt to remove a file that is linked under Data Links Manager control.

**Action:** No action required. Data Links Manager does not allow non-root users to remove files that are under its control.

## **DLFS errors from commands in the Solaris Operating Environment that open a file**

This section contains information about error messages that you might receive from the Data Links Filesystem Filter (DLFF) when you issue a command to open a file that is linked to a DATALINK column that is defined with READ PERMISSION DB. These errors only apply for non-root users. They are categorized by the types of files that you can open. Under each type of file, the section lists:

- Examples of common commands that you might use to open each type of file
- The error message text that displays
- The possible causes of the error message
- The suggested actions for you to take

This section lists the following types of open operations:

- “Opening a file without a token”
- “Opening a file with an expired token” on page 234
- “Opening a file with an invalid token” on page 235

### **Opening a file without a token**

For a file that is linked to a DATALINK column that is defined with READ PERMISSION DB, the DLFF can issue many error messages when you open a file without a token. Table 11 contains a partial list of commands that you might use in the Solaris Operating Environment to open a file without a token, and the error message text that is associated with each command.

The list in Table 11 does not include all of the possible commands that you can use in an open operation. The commands that are listed represent the types of commands that you might use when you open the file.

*Table 11. Partial list of commands that you might use in the Solaris Operating Environment to open a file without a token and the associated error message.*

<b>Command name</b>	<b>Error message</b>
<b>cat filename</b>	cat: cannot open filename
<b>cp filename newfilename</b>	cp: cannot access filename
<b>diff filename filenamefs</b>	diff: filename: Permission denied
<b>grep "str" filename</b>	grep: can't open filename
<b>head filename</b>	filename: Permission denied
<b>more filename</b>	filename: Permission denied

Table 11. Partial list of commands that you might use in the Solaris Operating Environment to open a file without a token and the associated error message. (continued)

Command name	Error message
tail filename	tail: cannot open input

When you issue a command to open a file without a token, the returned error message has the following cause and suggested action:

- Cause:** This is a system message that displays when you attempt to read the contents of a file that is linked under Data Links Manager control as a READ PERMISSION DB table column.
- Action:** Get a valid token from the database, and then re-issue the command.

### Opening a file with an expired token

For a file that is linked to a DATALINK column that is defined with READ PERMISSION DB, the DLFF can issue many error messages when you open a file with an expired token. Table 12 contains a partial list of commands that you might use in the Solaris Operating Environment to open a file with an expired token, and the error message text that is associated with each command.

The list in Table 12 does not include all of the possible commands that you can use in an open operation. The commands that are listed represent the types of commands that you might use when you open the file.

Table 12. Partial list of commands that you might use in the Solaris Operating Environment to open a file with an expired token and the associated error message.

Command name	Error message
cat "04E2_DEJ3FE__21WJSqB38XM;filename"	cat: cannot open 04E2_DEJ3FE__21WJSqB38XM;filename
cp "04E2_DEJ3FE__21WJSqB38XM;filename" newfilename	cp: cannot access 04E2_DEJ3FE__21WJSqB38XM;filename
diff "04E2_DEJ3FE__21WJSqB38XM;filename" filenamefs	diff: 04E2_DEJ3FE__21WJSqB38XM;filename: No such file or directory
grep "str" "04E2_DEJ3FE__21WJSqB38XM;filename"	grep: can't open 04E2_DEJ3FE__21WJSqB38XM;filename
head "04E2_DEJ3FE__21WJSqB38XM;filename"	04E2_DEJ3FE__21WJSqB38XM;filename: No such file or directory
more "04E2_DEJ3FE__21WJSqB38XM;filename"	04E2_DEJ3FE__21WJSqB38XM;filename: No such file or directory
tail "04E2_DEJ3FE__21WJSqB38XM;filename"	tail: cannot open input



When you issue a command to open a file with an expired token, the returned error message has the following cause and suggested action:

- Cause:** This is a system message that displays when you attempt to read the contents of a file that is linked under Data Links Manager control as a READ PERMISSION DB table column, and the token expired.
- Action:** Get a valid token from the database, and then re-issue the command.

### Opening a file with an invalid token

For a file that is linked to a DATALINK column that is defined with READ PERMISSION DB, the DLFF can issue many error messages when you open a file with an invalid token. Table 13 contains a partial list of commands that you might use in the Solaris Operating Environment to open a file with an invalid token, and the error message text that is associated with each command.

The list in Table 13 does not include all of the possible commands that you can use in an open operation. The commands that are listed represent the types of commands that you might use when you open the file.

*Table 13. Partial list of commands that you might use in the Solaris Operating Environment to open a file with an invalid token and the associated error message.*

Command name	Error message
cat "04E2AAAAAAAAAAAAASqB38XM;filename"	cat: cannot open 04E2AAAAAAAAAAAAASqB38XM;filename
cp "04E2AAAAAAAAAAAAASqB38XM;filename" newfilename	cp: cannot access 04E2AAAAAAAAAAAAASqB38XM;filename
diff "04E2AAAAAAAAAAAAASqB38XM;filename" filenamefs	diff: 04E2AAAAAAAAAAAAASqB38XM;filename: No such file or directory
grep "str" "04E2AAAAAAAAAAAAASqB38XM;filename"	grep: can't open 04E2AAAAAAAAAAAAASqB38XM;filename
head "04E2AAAAAAAAAAAAASqB38XM;filename"	04E2AAAAAAAAAAAAASqB38XM;filename: No such file or directory
more "04E2AAAAAAAAAAAAASqB38XM;filename"	04E2AAAAAAAAAAAAASqB38XM;filename: No such file or directory
tail "04E2AAAAAAAAAAAAASqB38XM;filename"	tail: cannot open input

When you issue a command to open a file with an invalid token, the returned error message has the following cause and suggested action:

**Cause:** This is a system message that displays when you attempt to read the contents of a file that is linked under Data Links Manager control as a READ PERMISSION DB table column, and the token is invalid.

**Action:** Get a valid token from the database, and then retry the operation.

**Related reference:**

- “DLFS errors from files referenced from columns with WRITE PERMISSION ADMIN” on page 240

---

## DLFS errors from commands (Windows)

This topic contains the potential error messages that you can receive on a Windows NTFS as a result of Data Links Filesystem Filter (DLFF) actions.

The commands on a Windows NTFS that can result in errors are organized into two sections:

- “DLFS errors from specific Windows commands”
- “DLFS errors from Windows commands that open a file” on page 238

### DLFS errors from specific Windows commands

This section contains information about error messages that you might receive from the Data Links Filesystem Filter (DLFF) that can result from certain Windows commands. It contains an alphabetical listing of the common commands. After each command name, this section lists:

- The error message text that displays
- The possible causes of the error message
- The suggested actions for you to take

This section lists the following commands:

- “attrib + r filename”
- “del “filename”” on page 237
- “mkdir command” on page 237
- “move command” on page 237
- “ren “d:\test\filename” “newfilename”” on page 237
- “rmdir or delete command from Windows Explorer” on page 238

#### **attrib + r filename**

Access is denied.

**Cause:** This is a system message that displays when you attempt to

change the attributes of a file that is linked under Data Links Manager control as a READ PERMISSION DB table column.

**Action:** No action required. Data Links Manager does not allow users other than dlmadmin to change the attributes of files that are linked under its control as READ PERMISSION DB table columns.

**del "filename"**

Access is denied.

**Cause:** This is a system message that displays when you attempt to delete a file that is linked under Data Links Manager control.

**Action:** No action required. Data Links Manager does not allow users other than dlmadmin to remove files that are under its control.

**mkdir command**

The device is not ready.

**Cause:** These are system messages that get displayed when you attempt to create a directory in a Data Links File System (DLFS), but the Data Links File Manager (DLFM) is not started.

**Action:** Ensure that the DLFM is running by issuing the **net start** command. This command shows a list of Windows services that are started. If the DLFM is not running, start it using the **dlfm start** command, and then try creating the directory again.

**move command**

A duplicate file name exists, or the file cannot be found.

**Cause:** This is a system message that displays when you attempt to rename a directory in a Data Links File System (DLFS).

**Action:** No action required. Data Links Manager does not allow you to rename a directory in a DLFS.

**ren "d:\testfilename" "newfilename"**

Access is denied.

**Cause:** This is a system message that displays when you attempt to rename a file that is linked under Data Links Manager control.

**Action:** No action required. Data Links Manager does not allow users other than dlmadmin to rename files that are linked under its control.

### **rmdir or delete command from Windows Explorer**

Access is denied. Or a dialog box displaying the message "Error Deleting File".

**Cause:** This is a system message that displays when users other than dlmadmin attempt to delete a directory in a Data Links File System (DLFS).

**Action:** Ensure that the directory that you are deleting does not have files that are linked under "DB2 Data Links File Manager" in its directory tree. The DLFS does not allow you to delete directories that have files that are linked under "DB2 Data Links File Manager".

### **DLFS errors from Windows commands that open a file**

This section contains information about error messages that you might receive from the Data Links Filesystem Filter (DLFF) when you issue a command to open a file that is linked to a DATALINK column that is defined with READ PERMISSION DB. These errors only apply for users other than dlmadmin. They are categorized by the types of files that you can open. Under each type of file, the section lists:

- Examples of common commands that you might use to open each type of file
- The error message text that displays
- The possible causes of the error message
- The suggested actions for you to take

This section lists the following types of open operations:

- "Opening a file without a token"
- "Opening a file with an expired token" on page 239
- "Opening a file with an invalid token" on page 240

#### **Opening a file without a token**

For a file that is linked to a DATALINK column that is defined with READ PERMISSION DB, the DLFF can issue many error messages when you open a file without a token. Table 14 on page 239 contains a partial list of Windows commands that you might use to open a file without a token, and the error message text that is associated with each command.

The list in Table 14 on page 239 does not include all of the possible commands that you can use in an open operation. The commands that are listed represent the types of commands that you might use when you open the file.

Table 14. Partial list of Windows commands that you might use to open a file without a token and the associated error message.

Command name	Error message
<code>more &lt;"d:\test\filename"</code>	Access is denied.
<code>type "filename"</code>	Access is denied.

When you issue a command to open a file without a token, the returned error message has the following cause and suggested action:

- Cause:** This is a system message that displays when a you attempt to display the contents of a file that is linked under Data Links Manager control as a READ PERMISSION DB table column.
- Action:** To read the contents of the file, get a valid token from the database, and then re-issue the command with the token.

### Opening a file with an expired token

For a file that is linked to a DATALINK column that is defined with READ PERMISSION DB, the DLFF can issue many error messages when you open a file with an expired token. Table 15 contains a partial list of Windows commands that you might use to open a file with an expired token, and the error message text that is associated with each command.

The list in Table 15 does not include all of the possible commands that you can use in an open operation. The commands that are listed represent the types of commands that you might use when you open the file.

Table 15. Partial list of Windows commands that you might use to open a file with an expired token and the associated error message.

Command name	Error message
<code>more "d:\test\04E2_D3iwQk_H5G1CXMzq_g;filename"</code>	The system cannot find the file specified.
<code>type "04E2_D3iwQk_H5G1CXMzq_g;filename"</code>	Access is denied.

When you issue a command to open a file with an expired token, the returned error message has the following cause and suggested action:

- Cause:** This is a system message that displays when you attempt to read the contents of a file that is linked under Data Links Manager control as a READ PERMISSION DB table column, and the token expired.
- Action:** Get a new token from the database, and then re-issue the command.

### Opening a file with an invalid token

For a file that is linked to a DATALINK column that is defined with READ PERMISSION DB, the DLFF can issue many error messages when you open a file with an invalid token. Table 16 contains a partial list of Windows commands that you might use to open a file with an invalid token, and the error message text that is associated with each command.

The list in Table 16 does not include all of the possible commands that you can use in an open operation. The commands that are listed represent the types of commands that you might use when you open the file.

*Table 16. Partial list of Windows commands that you might use to open a file with an invalid token and the associated error message.*

Command name	Error message
<code>more "d:\test\24E2_D3iwQk_H5G1CXMzq_g;filename"</code>	The system cannot find the file specified.
<code>type "24E2_D3iwQk_H5G1CXMzq_g;filename"</code>	Access is denied.

When you issue a command to open a file with an invalid token, the returned error message has the following cause and suggested action:

**Cause:** This is a system message that displays when you attempt to read the contents of a file that is linked under Data Links Manager control as a READ PERMISSION DB table column, and the token is invalid.

**Action:** Get a new token from the database, and then re-issue the command.

---

## DLFS errors from files referenced from columns with WRITE PERMISSION ADMIN

This topic contains the potential error messages that you can receive on both UNIX-based file systems and Windows NTFS when you attempt to open a file that is linked to a DATALINK column that is defined with WRITE PERMISSION ADMIN. These errors apply only for users other than dlmadmin (on Windows) or non-root users (on UNIX). They are a result of Data Links Filesystem Filter (DLFF) actions.

When an open operation fails, the DLFF can return a number of different error messages, depending on platform. There can be several possible causes for the error. This section lists:

- The action that you were trying to perform when the open operation fails
- The possible error messages that the DLFF can return for that condition
- The possible causes of the error message

- The suggested actions for you to take

## Opening a file for read operations

When you attempt to open a file to read the contents or to retrieve the file attribute and the open operation fails, your operating environment's file system can display several error messages for both UNIX and Windows platforms.

If your file system is UNIX-based, here are the possible error codes that the DLFS can return and an explanation of what each error code means:

### ENOENT

No such file or directory.

### EPERM

Operation not permitted.

### EACCES

Permission denied.

If your file system is Windows NTFS, here are the possible error codes that the DLFS can return and an explanation of what each error code means:

### ERROR\_FILE\_NOT\_FOUND

The system cannot find the file specified.

### ERROR\_ACCESS\_DENIED

Access is denied.

*Table 17. Possible causes and suggested actions for errors that result from opening a file for read operations.*

Cause	Action
The token is not valid (for example, an expired token or a token that is not well-formed), or no token is provided.	Re-generate a valid token by issuing an SQL SELECT statement on the corresponding table row and column.
The file is no longer linked.	Depending on the file permissions, try to open the file without a token.
You are currently modifying a file with a token, but you are not using the same token to read or write to that file.	When you begin to modify a file, you must continue to use the same token for either the read or write operation.

## Opening a file for write operations

When you attempt to open a file for write operations and the open operation fails, your operating environment's file system can display several error messages for both UNIX and Windows platforms.

If your file system is UNIX-based, here are the possible error codes that the DLFS can return and an explanation of what each error code means:

**ENOENT**

No such file or directory.

**EPERM**

Operation not permitted.

**EACCES**

Permission denied.

If your file system is Windows NTFS, here is the error code that the DLFS can return and an explanation of what that error code means:

**ERROR\_ACCESS\_DENIED**

Access is denied.

*Table 18. Possible causes and suggested actions for errors that result from opening a file for write operations.*

<b>Cause</b>	<b>Action</b>
The token is not valid (for example, an expired token or a token that is not well-formed), or no token is provided.	Re-generate a valid write token by issuing an SQL SELECT statement with either DLCOMPLETEWRITE or DLPATHWRITE scalar functions on the corresponding table row and column.
The file is linked to a column with RECOVERY YES and the file archiving process is not yet complete.	Wait a few seconds, and then retry the operation.
The file is currently in the update-in-progress state and is modified by using a different token or by a different user.	Use the <b>dlfm list upd_in_progress files</b> command to see who is currently updating the file. Then wait until the current update operation finishes, and retry the same operation.
Someone unlinked the file or issued the SQL UPDATE command to complete the file update operation, but the DB2 transaction did not yet commit.	Wait a few seconds, and then retry the operation.
The file is no longer linked.	Depending on the file permissions, try to open the file without a token.

**Related reference:**

- “DLFS errors from commands (AIX)” on page 223
- “DLFS errors from commands (Solaris Operating Environment)” on page 229



---

## DB2 Data Links Manager Glossary

**absolute path.** The full path name of an object. Absolute path names begin at the highest level, or "root" directory (which is identified by the forward slash (/) or backward slash (\) character).

**ADSM.** ADSTAR Distributed Storage Manager. See Tivoli Storage Manager.

**backup.** A copy of a database or table space that can be stored on a different medium and used to restore the database or table space in the event of failure or damage to the original.

**call level interface (CLI).** A callable API for database access that is an alternative to an embedded SQL API. In contrast to embedded SQL, the CLI does not require precompiling or binding to a database, but instead provides a standard set of functions to process SQL statements and related services at run time. See also DB2 Call Level Interface.

**daemon.** A system process that provides a specific service to remote applications or users.

**DB2 Call Level Interface (CLI).** An application that uses a standard set of functions to process SQL statements and related services at run time. It does not have to be precompiled or bound.

**DATALINK.** An SQL data type that enables logical references from the database to a file stored outside the database.

**Data Links Manager Administrator.** Refers to both the person and the user ID that is responsible for administering the DB2 Data Links Manager and its associated environment. Sometimes also referred to as "DLFM User," because when DB2 Data Links Manager is installed, an account with the default user ID "dlfm" is set up for use by the Data Links Manager Administrator. The Data Links Manager Administrator id also owns all of the resources used by the DLFM component, for example: (1) the DB2 instance containing the "DLFM\_DB" database, (2) linked files referenced in a READ PERMISSION DB DATALINK type column, and (3) the user ID under which the DLFM Server itself runs. For Windows NT and Windows 2000 environments only, an additional "dlmadmin" user account is created at installation time, but this account is intended to be a superuser account that the Data Links Administrator uses to perform any necessary advanced administration operations. The dlmadmin account is equivalent to the "root" user on UNIX. See also dlmadmin account, superuser.

**dlmadmin account.** In DB2 Data Links Manager, an account that gets created at install time on Windows NT and Windows 2000 environments only. By default, the account is named "dlmadmin." This account has advanced user privileges, and is intended to be equivalent to the "root" user in UNIX environments. Its purpose is to act as a superuser to perform any necessary advanced administration operations by both the DLFM component and the Data Links Manager Administrator on the Data Links server. However, unlike the "DLFM User" account, dlmadmin does not own any of the DLFM resources. The Data Links Manager Administrator uses this account in addition to the "DLFM User" account.

**Data Links Filesystem Filter (DLFF).** A DB2 Data Links Manager component. A file system filter program that enforces data integrity by ensuring valid and controlled access to linked files. See also linked files.

**Data Links File Manager (DLFM).** A component of the DB2 Data Links Manager that enables a DB2 database to manage files that are outside of the database.

**Data Links File System (DLFS).** A file system that is under the control of the Data Links Filesystem Filter (DLFF).

**Data Links server.** A machine that contains these DB2 Data Links Manager components: a Data Links File Manager (DLFM), a Data Links Filesystem Filter (DLFF) controlling a Data Links File System (DLFS), and DB2 (used as the Logging Manager).

**DB2 Data Links Manager.** A separately orderable DB2 feature that enables your applications to manipulate data residing in both unstructured files and in the relational database management system (RDBMS). DB2 Data Links Manager enables DB2 Universal Database to manage unstructured files as though they were directly stored in the database and provides the integration between the RDBMS and the external file systems through extensions to DB2 Universal Database.

**DB2 DataPropagator.** A product that provides DB2 replication for OS/390, z/OS, OS/400, z/VM, VM, and VSE operating-system environments. For UNIX and Windows operating system environments, replication is integrated with DB2 and does not require a separate license. See also replication.

**db2\_recon\_aid utility.** A utility that identifies database tables containing DATALINK type columns, and optionally runs the DB2 Reconcile utility against these tables.

**DLFM\_ASNCOPYD file-copy daemon (Data Links Manager Replication daemon).** The DLFM process which enables replication of DB2 Data Links Manager files (in conjunction with the associated DB2 relational data) in support of the DataPropagator product.

**DLFF.** See Data Links Filesystem Filter.

**DLFM.** See Data Links File Manager.

**DLFS.** See Data Links File System.

**DLFM\_DB database.** A DB2 database that acts as a logging manager for the Data Links server.

**DLM.** See DB2 Data Links Manager.

**Data Link Reconcile Pending (DRP).** The state of a DB2 table in which one or more DATALINK type columns contain file references whose integrity might be in doubt (for example, as the result of restoring a database without rolling forward through the database logs).

**Data Link Reconcile DLM Pending (DRDP).** The state of a DB2 table in which one or more DATALINK type columns contain file references whose integrity could not be verified (for example, as the result of a Data Links Manager being unavailable while running the DB2 RECONCILE utility).

**Data Link Reconcile Not Possible (DRNP).** The state of a DB2 table in which one or more DATALINK type columns contain file references whose integrity is violated (for example, as the result of restoring a database without being able to restore the files referenced therein).

**file access token .** See read token.

**file attributes.** The set of all characteristics describing a file, maintained by the operating system (in the containing file system). Includes such characteristics as the file owner, access permissions, last update timestamp, size, etc.

**file update operations** . All actions that are involved when modifying a file, especially in the case where the file is referenced in a DATALINK type column and is under the control of a DB2 Data Links Manager. See also linked file.

**File System Migrator (FSM)**. The virtual file system whose space usage is controlled by the Tivoli Space Manager. DB2 Data Links Manager supports use of this file system in the AIX operating environment.

**High Availability Cluster Multiprocessor (HACMP)**. Any hardware environment with multiple CPU nodes supporting the takeover of operations on one CPU by another. In a DB2 Data Links Manager configuration, both the DB2 host and the Data Links server can be configured in an HACMP environment.

**DB2 host**. In a DB2 Data Links Manager configuration, a DB2 database that contains a DATALINK column, on a DB2 server.

**Inter-Process Communication (IPC)**. A mechanism of an operating system that allows processes to communicate with each other within the same computer or over a network.

**JDBC (Java Database Connectivity)**. A set of database APIs for use in the Java programming language.

**JFS (Journaled File System)**. The native file system in the AIX operating system.

**link**. The action that the DB2 Data Links Manager takes to control a file that is referenced in a table that contains a DATALINK column. A file can be linked as the result of such database actions as an SQL UPDATE, INSERT, IMPORT or LOAD.

**linked file**. In DB2 Data Links Manager, a file that is referenced in a table DATALINK column defined with FILE LINK CONTROL. A linked file is maintained under the control of the DLFF component, in order to guarantee referential integrity.

**NTFS**. One of the native file systems in the Windows NT and follow-on operating environments.

**prefix**. In a DB2 Data Links Manager environment, the absolute path of a DLFS mount point (on UNIX) or a drive sharename (on Windows) under which linked files are stored.

**read token**. The authorization key embedded in a READ PERMISSION DB DATALINK column value, returned as a simple column value or by using the scalar functions DLURLCOMPLETE or DLURLPATH. A read token is required for reading a file referenced in a READ PERMISSION DB DATALINK column.

**RECONCILE**. A DB2 utility used to validate and repair references to files in the DATALINK columns of a table.

**RESTORE**. A DB2 utility that is used to rebuild a damaged or corrupted database or tablespace from a backup image produced with the BACKUP utility.

**replacement file**. In DB2 Data Links Manager, a file whose contents are intended to take the place of an existing linked file.

**replication**. The process of maintaining a defined set of data in more than one location. It involves copying designated changes for one location (a source) to another (a target), and synchronizing the data in both locations.

**retrieve\_query utility.** A standalone DLFM utility that displays the backup or archival status of all files managed by a particular Data Links server. The `retrieve_query` utility can also be invoked with the `dlfm retrieve` command.

**superuser.** A user that has various system control authorities above and beyond that of the ordinary user. In UNIX environments, the standard superuser is "root".

**Tivoli Space Manager.** A feature of the Tivoli Storage Manager product that moves files in and out of a secondary storage medium based upon actual file accesses in the primary native file system. This feature can be used with DB2 Data Links Manager to enable DATALINK files to be stored in a virtually infinitely sized file system

**Tivoli Storage Manager (TSM).** A client/server product that provides storage management and data access services in a heterogeneous environment. TSM supports various communication methods, provides administrative facilities to manage the backup and storage of files, and provides facilities for scheduling backups.

**UFS (UNIX File System).** The native file system in the Solaris Operating Environment.

**unlink.** The action that DB2 Data Links Manager takes to give up control of a file that is no longer referenced in a table that contains a DATALINK column. A file may be unlinked as the result of such database actions as an SQL UPDATE, DELETE, or DROP TABLE.

**unlinked file.** In a DB2 Data Links Manager environment, a file that the native file system on an operating system controls. By contrast, a linked file is controlled by the DLFF component.

**update-in-place.** In a DB2 Data Links Manager environment, the process of making changes to a linked file while a DATALINK column value in a database is pointing to that file. Any changes to linked files during an update-in-place operation become visible to database users immediately. However, depending on how you have defined the WRITE PERMISSION attribute of a particular DATALINK column, any changes to linked files might not actually become final until the DB2 host is notified that the update operations are complete.

**update-in-progress state.** The logical state of a file under the control of a DB2 Data Links Manager that is in the process of being updated. A linked file enters this state once it has been opened using a write token, and is no longer in this state when the DB2 host has been notified that the update is complete.

**updating.** In a DB2 Data Links Manager environment, the act of modifying a linked file.

**write token.** The authorization key embedded in a WRITE PERMISSION ADMIN DATALINK column value, returned by using the scalar function `DLURLCOMPLETEWRITE` or `DLURLPATHWRITE`. A write token is used for updating a file referenced in a WRITE PERMISSION ADMIN DATALINK column.

**XBSA.** An industry-standard API set for backup and restore utilities. XBSA is one of the archive area options available for use in maintaining backup copies of linked files in the DB2 Data Links Manager environment. The XBSA option is specified with the `DLFM_BACKUP_TARGET` registry variable.

---

## Notices

IBM may not offer the products, services, or features discussed in this document in all countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country/region or send inquiries, in writing, to:

IBM World Trade Asia Corporation  
Licensing  
2-31 Roppongi 3-chome, Minato-ku  
Tokyo 106, Japan

**The following paragraph does not apply to the United Kingdom or any other country/region where such provisions are inconsistent with local law:** INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions; therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make

improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product, and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licenses of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information that has been exchanged, should contact:

IBM Canada Limited  
Office of the Lab Director  
8200 Warden Avenue  
Markham, Ontario  
L6G 1C7  
CANADA

Such information may be available, subject to appropriate terms and conditions, including in some cases payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems, and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements, or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information may contain examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious, and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

#### COPYRIGHT LICENSE:

This information may contain sample application programs, in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Each copy or any portion of these sample programs or any derivative work must include a copyright notice as follows:

© (*your company name*) (*year*). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. *\_enter the year or years\_*. All rights reserved.

---

## Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both, and have been used in at least one of the documents in the DB2 UDB documentation library.

ACF/VTAM	LAN Distance
AISPO	MVS
AIX	MVS/ESA
AIXwindows	MVS/XA
AnyNet	Net.Data
APPN	NetView
AS/400	OS/390
BookManager	OS/400
C Set++	PowerPC
C/370	pSeries
CICS	QBIC
Database 2	QMF
DataHub	RACF
DataJoiner	RISC System/6000
DataPropagator	RS/6000
DataRefresher	S/370
DB2	SP
DB2 Connect	SQL/400
DB2 Extenders	SQL/DS
DB2 OLAP Server	System/370
DB2 Universal Database	System/390
Distributed Relational Database Architecture	SystemView
DRDA	Tivoli
eServer	VisualAge
Extended Services	VM/ESA
FFST	VSE/ESA
First Failure Support Technology	VTAM
IBM	WebExplorer
IMS	WebSphere
IMS/ESA	WIN-OS/2
iSeries	z/OS
	zSeries

The following terms are trademarks or registered trademarks of other companies and have been used in at least one of the documents in the DB2 UDB documentation library:

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.



Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.



---

# Index

## A

- abnormal termination
  - Data Links File Manager (DLFM) 62
- archive server 93, 202
  - using a local directory for in Data Links Manager 30
  - using Tivoli Storage Manager in Data Links Manager
    - AIX 24
    - Solaris Operating Environment 26
    - Windows 28

## B

- backing up
  - file systems
    - JFS on AIX 143
    - on Windows NT 143
    - UFS on Solaris Operating Environment 143

## C

- commands
  - See DLFF commands 71
  - See DLFM commands 68
- crash recovery 141

## D

- Data Links File Manager (DLFM) 6
  - error messages 155
  - listing
    - databases registered with 194
    - directories registered with 194
    - prefixes registered with 196
  - monitoring 206
  - registering databases with 183
  - restarting 61
    - after abnormal termination 62
  - starting 61
  - stopping 61
- Data Links Filesystem Filter (DLFF) 6
  - error messages
    - AIX 223

- Data Links Filesystem Filter (DLFF) (*continued*)
  - error messages (*continued*)
    - Solaris Operating Environment 229
    - Windows 236
- Data Links Manager
  - See DB2 Data Links Manager 189
- Data Links server
  - moving a DLFS to another hard disk 80
    - AIX 82
    - Solaris Operating Environment 86
    - Windows 89
- databases
  - dropping
    - Data Links File Manager from 189
  - registered with DLFM, listing 194
- DATALINK data type
  - overview 12
- DB2 clients
  - with Data Links Manager 8
- DB2 Data Links Manager
  - component descriptions 5
  - DB2 server 8
  - file system paradigm 9
  - overview 4
  - prefixes 184
  - replication 191, 193
  - supported systems 5
  - typical setups 15
  - working with applications 11
- DB2 Universal Database servers
  - use by DB2 Data Links Manager 8
- db2\_recon\_aid command 146
- DFS Client
  - overview 8
- DFS Client Enabler
  - overview 8
- directories
  - registered with DLFM, listing 194
- DLFF (Data Links Filesystem Filter)
  - adding in JFS environment 65

- DLFF (Data Links Filesystem Filter) (*continued*)
  - error messages
    - AIX 223
  - listing or adding registered file systems
    - AIX 68
    - Solaris Operating Environment 68
  - loading
    - JFS environments 69
    - UFS environments 70
  - querying
    - JFS environments 69
    - UFS environments 70
    - Windows 71
  - registering on Windows 71
  - unloading
    - JFS environments 69
    - UFS environments 70
  - unregistering on Windows 71
- dlff add command
  - syntax 213
- DLFF commands
  - dlff add 213
  - dlff get dlfmaccount command 214
  - dlff get loglevel 215
  - dlff list 215
  - dlff refreshtrace 216
  - dlff remove 217
  - dlff set dlfm\_write\_group 218
  - dlff set dlfmaccount 219
  - dlff set loglevel 221
- dlff get dlfmaccount command
  - syntax 214
- dlff get loglevel command
  - syntax 215
- dlff list command
  - syntax 215
- dlff refreshtrace command
  - syntax 216
- dlff remove command
  - syntax 217
- dlff set dlfm\_write\_group command
  - syntax 218
- dlff set dlfmaccount command
  - syntax 219

- dlff set loglevel command
  - syntax 221
- dlfm ? command
  - syntax 210
- DLFM (Data Links File Manager)
  - backup procedures 143
  - error messages 155
  - failure recovery scenarios 148
  - listing
    - databases registered with 74, 194
    - directories registered with 194
    - prefixes registered with 196
  - monitoring 206
  - processes
    - AIX 63
    - Solaris Operating Environment 63
    - Windows 64
  - registering databases 73, 183
  - restarting 61
  - restarting after abnormal termination 62
  - setup procedures 143
  - starting 61
  - stopping 61
- dlfm add\_db command
  - syntax 183
- dlfm add\_prefix command
  - syntax 184
- dlfm bind command
  - syntax 185
- dlfm command
  - syntax 183
- DLFM commands
  - dlfm 183
  - dlfm ? 210
  - dlfm add\_db 183
  - dlfm add\_prefix 184
  - dlfm bind 185
  - dlfm create 185
  - dlfm create\_db 186
  - dlfm deny 186
  - dlfm drop\_db 188
  - dlfm drop\_dlm 189
  - dlfm grant 189
  - dlfm grant replication read 191
  - dlfm grant replication write 193
  - dlfm help 194
  - dlfm list registered databases 194
  - dlfm list registered directories 194
  - dlfm list registered prefixes 196
- DLFM commands (*continued*)
  - dlfm list registered replication access 196
  - dlfm list registered users 197
  - dlfm list upd\_in\_progress files for db 198
  - dlfm list upd\_in\_progress files for prefix 199
  - dlfm refresh key 201
  - dlfm restart 202
  - dlfm retrieve 202
  - dlfm revoke 203
  - dlfm revoke replication 205
  - dlfm see 206
  - dlfm set link security 206
  - dlfm setup 207
  - dlfm shutdown 208
  - dlfm start 208
  - dlfm startdbm 209
  - dlfm stop 209
  - dlfm stopdbm 210
  - getting help with 183, 194, 210
  - dlfm create command
    - syntax 185
  - dlfm create\_db command
    - syntax 186
  - dlfm deny command
    - examples 101
    - syntax 186
  - dlfm drop\_db command
    - syntax 188
  - dlfm drop\_dlm command
    - syntax 189
  - dlfm grant command
    - examples 101
    - syntax 101, 189
  - dlfm grant replication read command
    - syntax 191
  - dlfm grant replication write command
    - syntax 193
  - dlfm help command
    - syntax 194
  - dlfm list registered databases command
    - examples 101
    - syntax 194
  - dlfm list registered directories command
    - syntax 194
  - dlfm list registered prefixes command
    - syntax 196
- dlfm list registered replication access control command
  - syntax 196
- dlfm list registered users command
  - examples 101
  - syntax 197
- dlfm list upd\_in\_progress files for db command
  - syntax 198
- dlfm list upd\_in\_progress files for prefix command
  - syntax 199
- dlfm refresh key command
  - syntax 201
- dlfm restart command
  - syntax 202
- dlfm retrieve command
  - syntax 202
- dlfm revoke command
  - syntax 203
- dlfm revoke replication command
  - syntax 205
- dlfm see command
  - syntax 206
- dlfm set link security command
  - syntax 206
- dlfm setup command
  - syntax 207
- dlfm shutdown command
  - syntax 208
- dlfm start command
  - for restarting 62
  - syntax 208
- dlfm startdbm command
  - syntax 209
- dlfm stop command
  - syntax 209
- dlfm stopdbm command
  - syntax 210
- DLFM\_ASNCOPYD file-copy daemon
  - enabling in DB2 Data Links Manager 54
  - granting authority to in DB2 Data Links Manager 193
  - listing granted operations for in DB2 Data Links Manager 196
  - revoking access privileges to 205
- DLFM\_DB
  - creating and configuring 186
  - creating initial tables for 185
  - dropping 188
  - ensuring log space for 57
  - setting up 207

DLFM\_DB (*continued*)  
starting the database manager  
for 209  
stopping the database manager  
for 210  
updating DB2 statistics for 185  
DLFS-DMAPP  
overview 6

## E

error messages  
changing diagnostic level 76  
Data Links File Manager 155  
Data Links Filesystem Filter  
(DLFF)  
AIX 223  
Solaris Operating  
Environment 229  
Windows 236

## F

failure recovery 141  
overview 141  
scenarios 148  
file system directory hierarchy 144  
file systems  
Data Links Manager enabled 80  
FORCE APPLICATION  
command 201

## I

isochronous delivery 9

## J

JFS  
increasing size 72

## L

linked files  
authorizing  
file write requests for  
replication 193  
replication users to  
access 191  
denying write access to 186  
granting write access to 189  
listing  
updates in progress, by  
prefix 199  
users with access to 197  
read operation security 105  
revoking access privileges  
from 203  
security, turning on and off 206  
showing updates-in-  
progress 198

linked files (*continued*)  
updating  
by replacing file contents 131  
manually 113  
native file system defines  
access 115  
using Data Links server  
authentication 116  
write operation security 105  
Logging Manager (DLM) 6

## M

migrating  
DLFF-enabled file system to  
another hard disk 80

## P

prefix  
adding, in Data Links  
Manager 184  
listing registered, in Data Links  
Manager 196  
listing updates in progress for  
linked files 199

## R

Read operation security  
in DB2 Data Links Manager 105  
RECONCILE command  
db2\_recon\_aid utility 146  
recovering  
write tokens in Data Links  
Manager 198  
replication  
enabling DLFM\_ASNCOPYD 54  
listing granted operations in Data  
Links Manager 196  
replicating linked files 191, 193  
revoking privileges for in Data  
Links Manager 205  
restarting a DLFM  
after abnormal termination 62  
description 61  
restoring  
file systems  
JFS on AIX 143  
UFS on Solaris Operating  
Environment 143  
Windows NT 143  
retrieving archive server backup files  
in Data Links Manager 202

## S

security  
file link 186, 189, 197, 203, 206  
rules and guidelines 101

security (*continued*)  
read operation  
DB2 Data Links  
Manager 105  
write operation  
in DB2 Data Links  
Manager 105

## T

Tivoli Storage Manager (TSM)  
using as an archive server in  
Data Links Manager  
AIX 24  
Solaris Operating  
Environment 26  
Windows 28  
tokens  
write 128  
Data Links Manager 201  
recovering in Data Links  
Manager 198

## U

UFS  
increasing size 72

## W

Windows 2000 x  
Windows NT x  
write operation security  
in DB2 Data Links Manager 105  
write tokens  
invalidating in Data Links  
Manager 201  
recovering in Data Links  
Manager 198  
usage considerations in DB2 Data  
Links Manager 128



---

## Contacting IBM

In the United States, call one of the following numbers to contact IBM:

- 1-800-237-5511 for customer service
- 1-888-426-4343 to learn about available service options
- 1-800-IBM-4YOU (426-4968) for DB2 marketing and sales

In Canada, call one of the following numbers to contact IBM:

- 1-800-IBM-SERV (1-800-426-7378) for customer service
- 1-800-465-9600 to learn about available service options
- 1-800-IBM-4YOU (1-800-426-4968) for DB2 marketing and sales

To locate an IBM office in your country or region, check IBM's Directory of Worldwide Contacts on the web at [www.ibm.com/planetwide](http://www.ibm.com/planetwide)

---

## Product information

Information regarding DB2 Universal Database products is available by telephone or by the World Wide Web at [www.ibm.com/software/data/db2/udb](http://www.ibm.com/software/data/db2/udb)

This site contains the latest information on the technical library, ordering books, client downloads, newsgroups, FixPaks, news, and links to web resources.

If you live in the U.S.A., then you can call one of the following numbers:

- 1-800-IBM-CALL (1-800-426-2255) to order products or to obtain general information.
- 1-800-879-2755 to order publications.

For information on how to contact IBM outside of the United States, go to the IBM Worldwide page at [www.ibm.com/planetwide](http://www.ibm.com/planetwide)



Part Number: CT19RNA

Printed in U.S.A.

SC27-1221-00



(1P) P/N: CT19RNA





Spine information:



IBM® DB2 Universal Database™

Data Links Manager Administration Guide and  
Reference

Version 8