IBM® DB2® Universal Database™

Data Links Manager Administration Guide and Reference

Version 8.2

SC27-1221-02

IBM® DB2® Universal Database™

Data Links Manager Administration Guide and Reference

Version 8.2

SC27-1221-02

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

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
- To find your local IBM representative, go to the IBM Directory of Worldwide Contacts at 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.

Contents

Tables

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.

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

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 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, “DB2 Data Links Manager concepts and operations,” on page 13 describes DB2 Data Links Manager concepts, such as how DB2 Data Links Manager interacts with DB2. This chapter also describes DB2 Data Links Manager operations, such as how DB2 Data Links Manager controls access to linked files.

- Chapter 3, “Prerequisites,” on page 27 lists all of the hardware and software requirements for running DB2 Data Links Manager.
- Chapter 4, “System management options,” on page 33 describes the DB2 Data Links Manager system options, and how to configure them.
- Chapter 5, “Working with the Data Links File Manager,” on page 67 discusses how to use the Data Links File Manager (DLFM) component.
- Chapter 6, “Security,” on page 95 lists the different types of security features that DB2 Data Links Manager provides and describes how to use them.
- Chapter 7, “Updating linked files,” on page 109 explains the various ways in which you can update linked files.
- Chapter 8, “Replicating linked files with DB2 DataPropagator,” on page 131 discusses how to configure DB2 Data Links Manager to work with DB2 DataPropagator to replicate linked data.
- Chapter 9, “Data Links File Manager server availability issues,” on page 155 describes the operations that you can and cannot perform when the Data Links File Manager (DLFM) component is not running on a Data Links server.
- Chapter 10, “Server recovery,” on page 175 explains how to recover a Data Links server in case of failure.
- Chapter 11, “Troubleshooting problems,” on page 187 is a troubleshooting reference.
- Appendix A, “Data Links File Manager commands,” on page 209 lists all Data Links File Manager (DLFM) commands.
- Appendix B, “Data Links Filesystem Filter commands,” on page 233 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 241 lists operations and potential error messages for a Data Links File System (DLFS).
- “DB2 Data Links Manager Glossary” on page 257 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

What's new in DB2 Data Links Manager Version 8

Enhancements delivered in Version 8.2:

- Data Links Manager online help and error message functionality

  This release enhances existing and adds new online help functions for the Data Links File Manager (DLFM) component of Data Links Manager: The **db2 ? message** command can now be used with DLFM error message numbers to get online help about DLFM error messages. The **DLFM ?** command can now produce online command help in multiple languages.

  Additionally, all DLFM error messages are now available in multiple languages.

Enhancements delivered in Version 8.1 (including all FixPaks and modification levels):

- 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 supports AIX® 5L Version 5.2
- DB2 Data Links Manager is available on the Windows 2000 operating system, in addition to Windows NT, AIX, and the Solaris Operating Environment.
- DB2 Data Links Manager supports both the Version 8 and Version 7 Solaris Operating Environments.
- You can 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® and Windows® 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:

- "DB2 Data Links environment" on page 5

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® Data Links environment is composed of:

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

You can install DB2 Data Links Manager on:

- AIX® systems to manage files stored in a Journaled File System (JFS).
- Solaris™ Operating Environments to manage files stored in a UNIX® File System (UFS).
- Windows® 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 12
- "Data Links server components" on page 5
- "DB2 clients in the Data Links Manager environment" on page 7
- "DB2 servers in the Data Links Manager environment" on page 7

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® (Logging Manager)

Data Links File Manager (DLFM)

The DLFM tracks all 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 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 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)
DLFS file systems include FSM (File System Migrator; the filter file system component of Tivoli® Space Manager), JFS, NTFS, or UFS environments.

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 share name 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, and so on.

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. 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 12
- "DB2 clients in the Data Links Manager environment" on page 7
- "DB2 servers in the Data Links Manager environment" on page 7

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.

Related concepts:

- "DB2 Data Links environment" on page 5
- "DB2 Data Links Manager typical setups" on page 12
- "Data Links server components" on page 5
- "DB2 servers in the Data Links Manager environment" on page 7

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. You can register more than one Data Links Manager with a given DB2 database.

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

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 12
- "Data Links server components" on page 5
- "DB2 clients in the Data Links Manager environment" on page 7

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 might 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 could be captured in high volumes, and you might 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 could 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 could 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 could 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 9

DB2 Data Links Manager and your applications

DB2® 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 7

DATALINK data type

Data Links technology uses the DATALINK data type, implemented as an SQL data type in DB2® 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® 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.

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.
- 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 7
- "DB2 Data Links Manager and your applications" on page 9
- "Updating approaches overview" on page 109

Related reference:

• "Updating approaches summary" on page 127

DB2 Data Links Manager typical setups

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

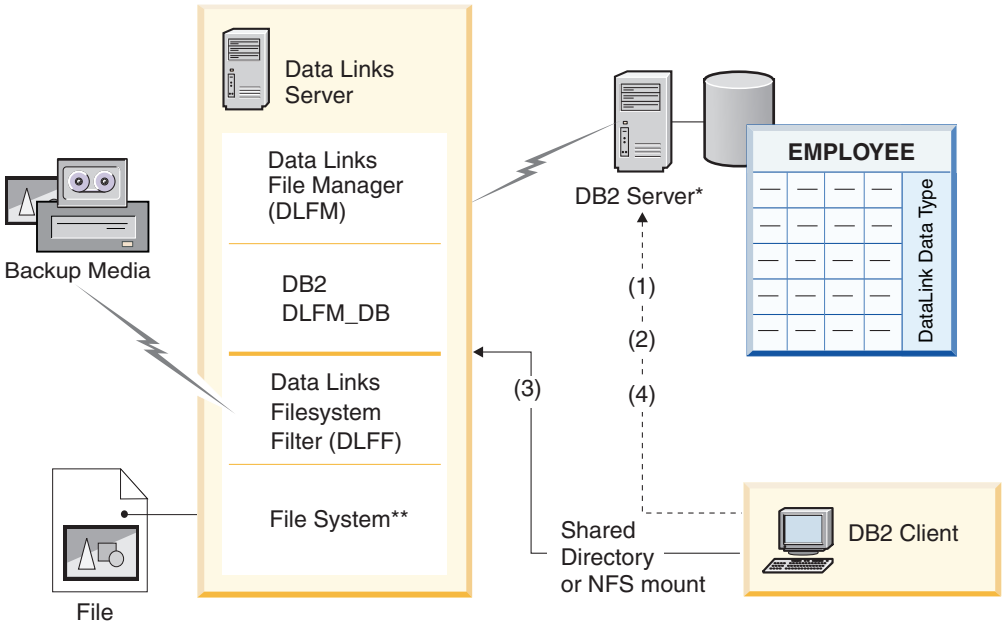


Figure 1. Overview of Data Links Manager Processing (FSM on AIX, NTFS, JFS, or UFS environments)

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(dl1) into :var_dl1 from EMPLOYEE
   ```

3. The application then copies the :var_dl1 file to the new_version file over a shared drive on Windows® or an NFS mount on AIX 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 dl1=dlvalue(:new_version)
   commit
   ```

Related concepts:

- "DB2 Data Links Manager" on page 4
- "DB2 Data Links environment" on page 5

Chapter 2. DB2 Data Links Manager concepts and operations

This chapter provides an overview of and information about essential DB2 Data Links Manager concepts, operations, functions, and features.

DB2 Data Links Manager file control

DB2® Data Links Manager extends the reach of the RDBMS to the operating system by providing control of files that are referenced in DATALINK data type columns of a database. Working with the DB2 environment, DB2 Data Links Manager provides four key elements of file control:

Referential integrity of external files
DB2 Data Links Manager ensures that end-users cannot delete, move, or rename any external file that is referenced in a database.

RDBMS data-value-based access to external files
DB2 grants or denies end-users the ability to read a referenced external file, and determines the level of control that DB2 Data Links Manager can enforce for write operations on referenced external files. DB2 Data Links Manager grants or denies end-users the ability to reference (link) external files in a database, and the ability to write to (update) a referenced external file. All of these access controls are optional, and you can customize them as needed.

Automatic and coordinated backup-and-restore capability within transactional environments
DB2 Data Links Manager enables you to coordinate backup and recovery of external data in synchronization with the associated database. This coordination is optional, and can be customized as needed.

Transaction consistency
With DB2 Data Links Manager, you can ensure that changes that affect both the database and external files are executed within a transactional context. Transaction consistency helps to preserve the logical integrity and consistency of the external files.

Related concepts:

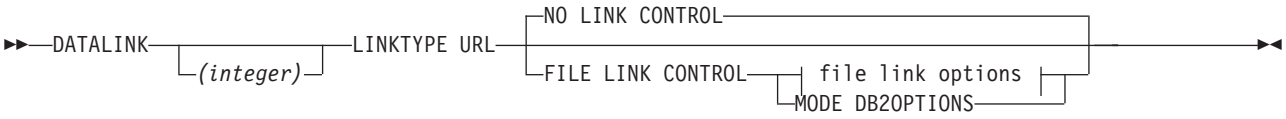
- "DATALINK column configuration" on page 14
- "Data Links servers" on page 15
- "File system prefixes" on page 17
- "Linked and unlinked files" on page 17
- "DB2 database and DB2 Data Links Manager file access controls" on page 19
- "Backup and recovery of linked files" on page 24
- "Examples of using UDFs to access linked files" on page 25
- "Introduction to Data Links Manager security" on page 95
- "Built-in security features" on page 95
- "Data access security features" on page 96
- "Updating approaches overview" on page 109

DATALINK column configuration

The attributes that you use to define a DATALINK column in a table determine how much control, if any, DB2® Data Links Manager has over the files that are to be referenced in the column.

Important: DB2 does not allow you to alter or drop existing DATALINK type columns after you create them. Before you create DATALINK columns in your tables, you must plan the column configuration to produce the behavior of DB2 Data Links Manager that you want.

You define DATALINK columns in a table by using the SQL statement CREATE TABLE or ALTER TABLE ADD COLUMN. Figure 2 illustrates the syntax of the portion of these two SQL statements that is used to define DATALINK columns.



file link options:

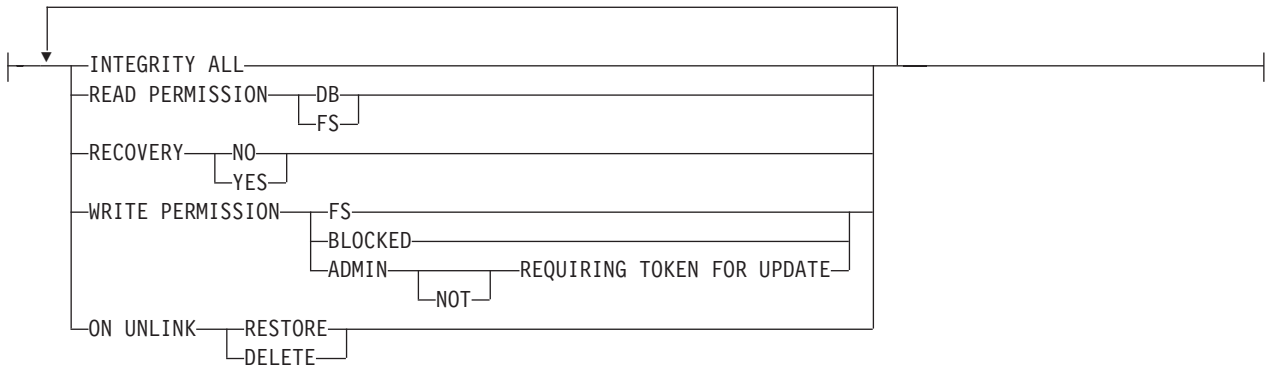


Figure 2. DATALINK column definition syntax

The LINK CONTROL attribute that you choose determines whether DB2 Data Links Manager has control over the files that are referenced in the column.

NO LINK CONTROL
Specifies that no check is made to determine that the referenced file exists. Only the syntax of the URL is checked. DB2 Data Links Manager has no control over the referenced file.

You might want to use this attribute when referential integrity of the referenced data is not a requirement. For example, you can use the NO LINK CONTROL attribute in the following situations:

- While using a development environment as a starting point for building a production environment.
- If you are using an application that already provides referential integrity.

You can also define DATALINK columns with the NO LINK CONTROL attribute if you need to reference data on a file system that DB2 Data Links Manager does not currently support.

FILE LINK CONTROL
Specifies that a check be made for the existence of the referenced file. You can use the additional options shown in the file link options clause to specify the DB2 Data Links Manager controls. You can either specify the individual file link attributes, or use the MODE DB2OPTIONS attribute, which defines a set of default file link options.

You must specify the FILE LINK CONTROL attribute to enable DB2 Data Links Manager to control the referenced files.

The MODE DB2OPTIONS attribute defines the following file link attributes:

- INTEGRITY ALL
- READ PERMISSION FS
- RECOVERY NO
- WRITE PERMISSION FS

The ON UNLINK attribute is not available when you use MODE DB2OPTIONS.

Requirement: Any DB2 database that uses one or more DATALINK columns must increase the current size of the APP_CTL_HEAP_SZ configuration parameter by 128. For example, if you are currently using the default value of 128, you must increase the value to 256. If you already increased the APP_CTL_HEAP_SZ value to be greater than the default value, you must increase the existing value by 128.

Related concepts:

- "DATALINK data type" on page 10
- "Data Links servers" on page 15
- "DB2 database and DB2 Data Links Manager file access controls" on page 19

Related reference:

- "app_ctl_heap_sz - Application control heap size configuration parameter" in the *Administration Guide: Performance*
- "ALTER TABLE statement" in the *SQL Reference, Volume 2*
- "CREATE TABLE statement" in the *SQL Reference, Volume 2*

Data Links servers

External files that are to be referenced by a database are stored on one or more computers where DB2® Data Links Manager is installed. Those computers are called *Data Links servers*. A single DB2 database can communicate with up to 16 Data Links servers. A Data Links server has the following components:

Data Links File Manager (DLFM)
Tracks all of the files on a Data Links server that are referenced by one or more DB2 databases.

Data Links Filesystem Filter (DLFF)
Enforces data integrity by ensuring valid and controlled access to linked files. A file system that uses the DLFF program is called a Data Links File System (DLFS). On a DLFS, all file and directory operations are routed

through the DLFF, regardless of who initiates the actions (for example, a local machine user or a remote client). Because of this routing, a small overhead is introduced for certain file operations, such as open and query permissions.

DB2 run-time environment
A DB2 database acts as a logging manager for a Data Links server. This database is named DLFM_DB by default at installation time. The logging manager database maintains registration information about DB2 host databases that can connect to the Data Links server, and file systems that are managed by DLFF. A file system is either a mount point in UNIX® environments, or the share name of a drive in Windows® environments.

To enable a DB2 database to reference the files on a specific Data Links server, you must do the following tasks:

1. Ensure that the database contains tables with one or more properly configured DATALINK data type columns.

   Requirement: To enable DB2 Data Links Manager to control referenced files, the DATALINK column must use the FILE LINK CONTROL attribute.
2. Enable Data Links Manager-to-DB2 communications by registering the DB2 database with the Data Links server.
3. Enable DB2-to-Data Links Manager communications by registering the Data Links server with the DB2 server where the database instance resides.

After you establish communications between a DB2 database and a Data Links server, you must configure the DB2 Data Links Manager file link security feature. With this feature, you can specify who can reference the files on a Data Links server.

Related concepts:
- "DATALINK data type" on page 10
- "Data Links server components" on page 5
- "DATALINK column configuration" on page 14
- "File system prefixes" on page 17
- "Linked and unlinked files" on page 17
- "DB2 database and DB2 Data Links Manager file access controls" on page 19
- "Backup and recovery of linked files" on page 24
- "Introduction to Data Links Manager security" on page 95
- "Built-in security features" on page 95
- "Data access security features" on page 96

Related tasks:
- "Creating additional Data Links File Systems on a Data Links server" on page 47
- "Adding DLFMs to a specific DB2 database on the DB2 host" on page 56
- "Adding DB2 databases to a DLFM on a Data Links server" on page 54

File system prefixes

When you create a new Data Links File System (DLFS) on a Data Links server, one of the first configuration tasks that you must perform is to determine the directory, or the directory path, where the referenced files are to be stored. After you determine a storage directory for the files, you register a *prefix* corresponding to that location with the Data Links File Manager (DLFM) on a Data Links server.

A prefix is the absolute path of a file system mount point (on UNIX® systems) or a drive share name (on Windows® systems) under which files are stored. Absolute path names begin at the highest level, or "root" directory, which, depending on your operating system, is identified by the forward slash (/) or backward slash (\) character. For example, an absolute path on a UNIX system might be /files/employees, and an absolute path on a Windows system might be \files\employees.

You register prefix values with the DLFM on a Data Links server using the **dlfm add_prefix** command. You can specify only one prefix per DLFS. However, you can create subdirectories of a prefix at any time. You do not have to register or define the subdirectories of an existing prefix to the Data Links Filesystem Filter (DLFF) or DLFM.

You must register prefixes with a DLFM before you can begin linking to the files stored on a Data Links server.

Related concepts:

- "Data Links servers" on page 15
- "DB2 database and DB2 Data Links Manager file access controls" on page 19

Related tasks:

- "Identifying DLFS paths to contain linked files" on page 48
- "Enabling and registering file systems with DLFM (AIX, Solaris Operating Environment)" on page 70
- "Listing registered file systems (AIX, Solaris Operating Environment)" on page 73
- "Listing registered drives (Windows operating systems)" on page 73
- "Registering, querying and de-registering a DLFF (Windows operating system)" on page 75

Related reference:

- "dlfm add_prefix command" on page 210
- "dlff add command (Windows operating system)" on page 233
- "dlff list command (Windows operating system)" on page 235
- "dlfm list registered prefixes command" on page 219

Linked and unlinked files

Linking is the action that the DB2® Data Links Manager takes to control a file that is referenced in a DATALINK column that was defined using the FILE LINK CONTROL attribute. A file can be linked as the result of database actions such as an SQL UPDATE, INSERT, IMPORT, or LOAD operation.

A *linked file* is a file that is referenced in a table DATALINK column that was defined using the FILE LINK CONTROL attribute. The Data Links Filesystem Filter (DLFF) on a Data Links server controls the operations that can and cannot be performed on a linked file.

When a file is linked, it is always maintained under the control of the DLFF to guarantee referential integrity. Through the DLFF, DB2 Data Links Manager controls all access to the file, regardless of who is trying to access the file (for example, a person or application), and regardless of how the file is accessed (for example, through a database user defined function or by accessing the local file system).

Important: You cannot link a file that is already in a linked state, because a file can be controlled by only one DB2 Data Links Manager at a time. In technical terms, a file can be linked in only one DATALINK column defined with the FILE LINK CONTROL attribute at any given time. Multiple DATALINK columns defined with the FILE LINK CONTROL attribute cannot link to the same file. However, you can point to a linked file from other DATALINK columns defined with the NO LINK CONTROL attribute. Even though the linked file is being referenced from multiple DATALINK columns, the file is still being controlled by only *one* DB2 host DATALINK reference — because only one of the DATALINK columns pointing to the file uses the FILE LINK CONTROL attribute.

End-users cannot *ever* delete, move, or rename any linked file. You can also optionally restrict who can write to (update) a linked file.

You regulate who can *link* to any file on a Data Links server by using the DB2 Data Links Manager file link control security feature. You configure the file link security control feature with a series of **dlfm** commands.

Important: The file link security control feature is turned on (activated) by default at DB2 Data Links Manager installation time. When the file link security control feature is activated, no one, not even a Data Links server superuser, can link to the files on a Data Links server until the Data Links Manager Administrator explicitly grants the linking privilege.

In contrast to linking, *unlinking* is the action that DB2 Data Links Manager takes to give up control of a file. A file can be unlinked as the result of database actions such as an SQL UPDATE, DELETE, or DROP TABLE operation. A file that was changed from a linked state to an unlinked state is referred to as an *unlinked file*. An unlinked file remains stored within the Data Links File System (DLFS), but reverts to the control of the native file system on the current operating system. Any file access permissions that were defined by the native file system are restored. DB2 Data Links Manager does not track or control the unlinked files that are stored on a Data Links server.

You can relink an unlinked file at any time. You might need or want to unlink a file for brief periods of time, then relink it again. For example, moving a linked file between tables requires unlinking the file from the original table, then relinking the file in the new table. Another example: you might want to change (update) files only when they are unlinked, then relink the files when the changes are complete.

Related concepts:

- "DATALINK column configuration" on page 14
- "DB2 database and DB2 Data Links Manager file access controls" on page 19

- "Backup and recovery of linked files" on page 24
- "Data Links server file backups" on page 33
- "Basic linked file security controls" on page 97
- "Advanced file management security features" on page 98
- "DB2 utilities and the Data Links File Manager" on page 155
- "Database recovery using DB2 utilities with the Data Links File Manager" on page 160
- "Updating approaches overview" on page 109
- "Approach 1: Unlink/update/relink files" on page 111

DB2 database and DB2 Data Links Manager file access controls

Linking a file from a properly configured table DATALINK column places that file under control of DB2® Data Links Manager. The file remains under control of DB2 Data Links Manager for as long as it remains in a linked state. Because the acts of linking and unlinking affect file integrity, you need to regulate who can perform file link and file unlink actions.

DB2 and DB2 Data Links Manager features work together to enable you to control who can link files, who can remove files from a linked state, and who can perform specific actions on linked files.

- DB2 host environment access permissions include access to SQL statements (for example, UPDATE) and database commands (for example, LOAD), which regulate who can access specific databases to perform file link actions.
- The DB2 Data Links Manager file link security control feature is activated and configured on individual Data Links servers by using a series of **dlfm** commands.
- DB2 Data Links Manager file access permissions enable you to regulate who can perform read and write actions on linked files.
- Only DB2 host environment access permissions regulate who can unlink files. DB2 Data Links Manager does not regulate who can perform file unlink actions.

Figure 3 on page 20 shows the DB2 host environment access permissions and the DB2 Data Links Manager file link security commands that control file linking, and how they work together to regulate a file link action.

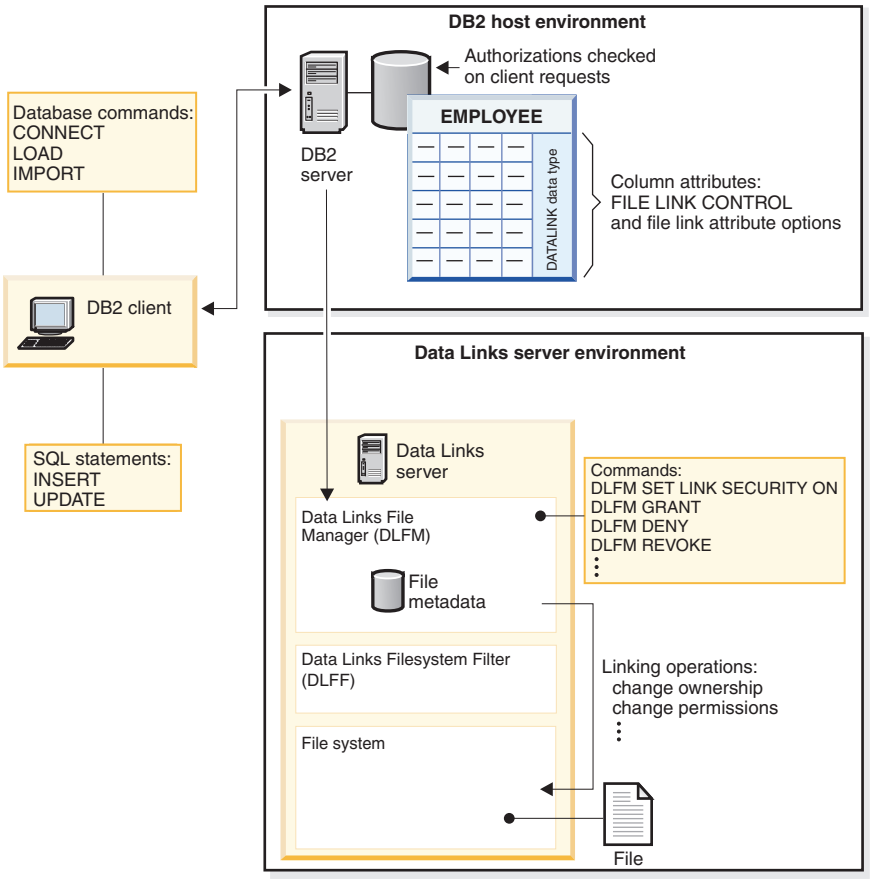


Figure 3. File linking actions and controls

Table 1 lists the required DB2 host environment access permissions and the DB2 Data Links Manager commands that enable you to control file linking and unlinking actions.

Table 1. File link and unlink operation requirements

Operation	Required DB2 database commands	Required SQL statements	Controlling DLFM commands
Link	Both: • CONNECT • LOAD or IMPORT	INSERT or UPDATE	Either: • Both – dlfm set link security on – dlfm grant • dlfm set link security off
Prohibit link	None	REVOKE INSERT or REVOKE UPDATE	All of these commands: • dlfm set link security on • dlfm deny • dlfm revoke

Table 1. File link and unlink operation requirements (continued)

Operation	Required DB2 database commands	Required SQL statements	Controlling DLFM commands
Unlink	CONNECT	Any of these statements: • DELETE • UPDATE • DROP TABLE • DROP TABLESPACE • DROP DATABASE	None

After a file is linked, access to the file is regulated by a combination of DB2 host database permissions and DB2 Data Links Manager permissions.

Only a superuser on a Data Links file server machine can *delete* or *rename* a file that is linked. Other actions, such as read and write, can be regulated through a combination of DB2 host environment access permissions and DB2 Data Links Manager file access permissions. The specific DB2 Data Links Manager file access permissions that you can use will vary, and depend on the file link attributes of the DATALINK column from which a file is linked.

You configure DB2 Data Links Manager file access permissions on individual Data Links servers by using a series of **dlfm** commands.

Figure 4 on page 22 shows how the DB2 host database access permissions and the DB2 Data Links Manager interact to regulate read and write operations for linked files.

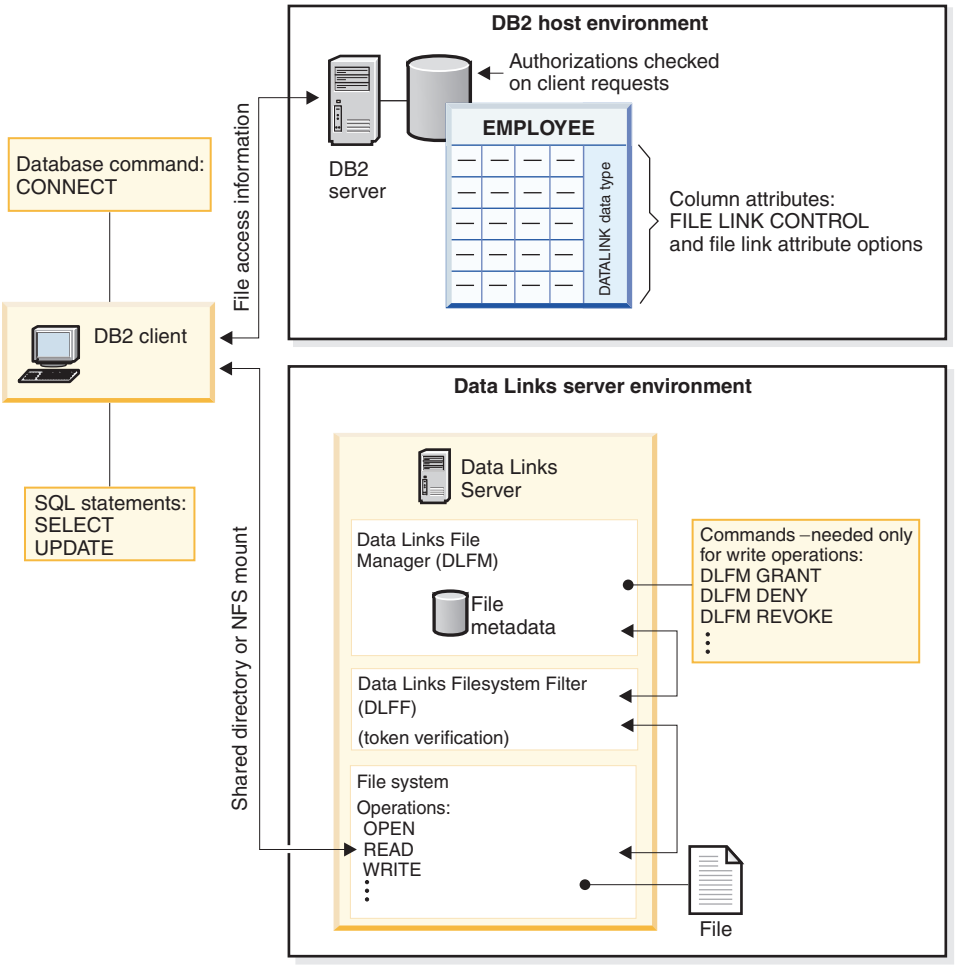


Figure 4. Accessing linked files for read or write actions

Table 2 lists file system actions that you can perform on linked files and their associated Data Links server directories, the DB2 host database requirements that enable the action to be restricted, and the DB2 Data Links Manager commands that enable you to restrict each file system action.

Table 2. File system operations and their configuration requirements

		DB2 host database requirements			
Action	**Applies to**	**DATALINK column file link attributes**	**DB2 database commands**	**SQL statements**	**Controlling DLFM commands**
Read: access controlled by database[1]	Files	READ PERMISSION DB	CONNECT	SELECT	None
Read: access controlled by file system[1]	Files	READ PERMISSION FS	CONNECT[2]	SELECT[2]	None
Write: prohibited	Files	WRITE PERMISSION BLOCKED	None	None	None
Write: access controlled by file system[1]	Files	WRITE PERMISSION FS	CONNECT[2]	SELECT[2]	None

Table 2. File system operations and their configuration requirements (continued)

		DB2 host database requirements			
Action	**Applies to**	**DATALINK column file link attributes**	**DB2 database commands**	**SQL statements**	**Controlling DLFM commands**
Write: access controlled by DB2 Data Links Manager[1]	Files	WRITE PERMISSION ADMIN [NOT] REQUIRING TOKEN FOR UPDATE	CONNECT	SELECT UPDATE	Any of these commands: • dlfm grant • dlfm deny • dlfm revoke
Create	Files and directories	The file system controls this action independently of DB2 and Data Links Manager.			
Copy	Files and directories	Uses the permissions of the source and target file system for the associated read and write activities.			
Move	Files and directories	Prohibited for all users except Data Links server superusers.			
Delete	Files and directories	Prohibited for all users except Data Links server superusers.			
Rename	Files and directories	Prohibited for all users except Data Links server superusers.			
DB2 replication: read[3]	Directories	None	None	None	dlfm grant replication read
DB2 replication: write[3]	Directories	None	None	None	dlfm grant replication write request

Notes:

1. Includes open-for-read or open-for-write operations.
2. This authority is not always required, because it is possible to access a file without obtaining its URL.
3. The application or user that performs the replication activity must be authorized to access the Data Links server machines where the linked files are stored.

Related concepts:

- "DATALINK column configuration" on page 14
- "Linked and unlinked files" on page 17
- "Backup and recovery of linked files" on page 24
- "Data access security features" on page 96
- "Basic linked file security controls" on page 97
- "Advanced file management security features" on page 98
- "Read operation security" on page 103
- "Write operation security" on page 103
- "Updating approaches overview" on page 109

Related reference:

- "dlfm deny command" on page 211
- "dlfm grant replication read command" on page 216
- "dlfm grant replication write request command" on page 217
- "dlfm grant command" on page 214
- "dlfm revoke command" on page 225

- "dlfm set link security command" on page 228

Backup and recovery of linked files

Data Links Manager automatically takes backups of linked files referenced in DATALINK columns defined with the RECOVERY YES attribute. These backups are automatically stored in the *archive area* for a Data Links server. An archive area can be on a local disk or on a separate computer. The number of file backups that are kept depends on the value of the DB2® host database configuration parameter NUM_DB_BACKUPS.

Data Links Manager uses *copy daemon* processes to copy linked files to the archive area. 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.

If reconcile operations are performed as part of database recovery, any valid backed-up versions of files that exist in a Data Links server archive area will be used. A backed-up version of a file is always used for reconciliation purposes, regardless of whether a linked *or* unlinked version of that same file already exists on the Data Links server. The contents of any existing file to be replaced are preserved in a file of the same name, with the extension MOD. The restored version of the file is relinked and placed under DB2 Data Links Manager control.

For example, if the backup of a file named emp50100.gif is used, the contents of the existing emp50100.gif file will be copied to a file called emp50100.gif.MOD. Then, the restored emp50100.gif file is relinked and placed under DB2 Data Links Manager control.

Related concepts:

- "Linked and unlinked files" on page 17
- "Data Links server file backups" on page 33
- "Failure and recovery overview" on page 175
- "Bringing the file system directory hierarchy to the current point in time" on page 178
- "DB2 Data Links Manager recovery scenarios" on page 182
- "Database recovery using DB2 utilities with the Data Links File Manager" on page 160

Related tasks:

- "Setting the number of copy processes" on page 65

Related reference:

- "Data Links server machine failure" on page 175
- "DB2 Data Links Manager system setup and backup recommendations" on page 176
- "File system backup and restore recommendations" on page 178
- "db2_recon_aid utility" on page 180

Examples of using UDFs to access linked files

User-defined functions (UDFs) are extensions or additions to the existing built-in functions of the SQL language. A user-defined function can be a scalar function, which returns a single value each time it is called; a column function, which is passed a set of like values and returns a single value for the set; a row function, which returns one row; or a table function, which returns a table.

UDFs are especially useful when combined with Data Links technology. With a UDF, you can extract data from files referenced in DATALINK columns, regardless of where the referenced files are physically located. Creating and using UDFs enables you to extract data from linked files at any time, without the overhead of keeping extra data stored in a database.

You must consider the following points before you create a UDF to access data from files referenced in DATALINK columns:

- The route to the Data Links server machine or machines that the UDF can use to access the referenced files.

  You can access the required files remotely, either by using an NFS mount on UNIX® systems, or through a shared drive on Windows®. Or you can access the required files directly, by going through the native file system on the Data Links server machine itself (for example, through an FTP or HTTP server).
- The read access permissions associated with the DATALINK column or columns that the UDF needs to access.
  - If the column to be accessed uses the READ PERMISSION DB attribute, the UDF can use the embedded token in the URL to read the file.
  - If the column to be accessed uses the READ PERMISSION FS attribute, the UDF must have the appropriate file permissions on the native file system of the associated Data Links server where the file resides.

You might find the following examples of UDFs that access data from linked files helpful in creating your own UDFs.

- A scalar UDF called LASTUPDATE retrieves the last modification timestamp of a linked file in a Data Links File System (DLFS).

  Assume that the LASTUPDATE function accepts a DATALINK argument and returns a TIMESTAMP value. The body of the function takes the specified URL and remotely accesses the specified Data Links server to obtain information about the file from the native file system. The last modification date and time will be formatted as an SQL TIMESTAMP value for output.

  ```
  SELECT EMPNO, LASTUPDATE(RESUME)
  FROM EMPLOYEE
  WHERE EMPNO = 123456
  ```

  This example UDF is useful for DATALINK columns defined with the WRITE PERMISSION FS attribute, because the files can be modified without interacting with the DB2® host database. However, you can use this UDF with any sort of DATALINK column.

  Alternative: For DATALINK columns defined with the WRITE PERMISSION ADMIN attribute, you could store the last modification time in another column of the same table, because all file updates also require DB2 host database updates.
- A table UDF called HTMLREFS returns a set of URLs that are referenced within a specified HTML file. The HTML file is linked to the DB2 host database through a DATALINK column. The following query produces a list of server

names referenced within the HOMEPAGE HTML file (referenced from a DATALINK column in the EMPLOYEE table) for a specific employee.

```
SELECT e.EMPNO, dlurlserver(h.URLREF)
FROM EMPLOYEE e, TABLE( HTMLREFS(e.HOMEPAGE) ) h
WHERE e.EMPNO = 123456
GROUP BY e.EMPNO, dlurlserver(h.URLREF)
```

Assume that the HTMLREFS function accepts a DATALINK argument and returns a table with one column of DATALINK values. The column is named URLREF in the CREATE FUNCTION statement. The body of the function takes the given URL reference (to an HTML file), then reads that file from its remote location on the corresponding Data Links server, and looks for all URL references in that file. Finally, the function returns a row with a single DATALINK value for each URL found. In this example, the built-in scalar function DLURLSERVER is used to select only the server portion of the URL.

Related concepts:

- "User-defined functions" in the *SQL Reference, Volume 1*
- "Creating a user-defined function (UDF) or method" in the *Administration Guide: Implementation*
- "DB2 User-Defined Functions and Methods" in the *Application Development Guide: Programming Client Applications*

Movement of Data Links Manager data

You can use the DB2® export, import, and load utilities to move Data Links Manager data. Together, these three utilities are referred to as the *DB2 data movement* utilities. You can perform the following tasks using the DB2 data movement utilities:

- Move linked files between DB2 databases, and between DB2 and other applications (for example, spreadsheets)
- Load data into DATALINK columns in DB2 tables, from operating system files
- Archive linked files

The DB2 data movement utilities accommodate both the database data (for example, information about tables), and the data files on the corresponding Data Links servers.

Related concepts:

- "Moving DB2 Data Links Manager Data Using Export - Concepts" in the *Data Movement Utilities Guide and Reference*

Related tasks:

- "Using export to move DB2 Data Links Manager data" in the *Data Movement Utilities Guide and Reference*
- "Using import to move DB2 Data Links Manager data" in the *Data Movement Utilities Guide and Reference*
- "Using load to move DB2 Data Links Manager data" in the *Data Movement Utilities Guide and Reference*

Chapter 3. Prerequisites

This chapter describes the prerequisites for DB2 Data Links Manager.

Minimum hardware and software requirements

This topic lists the minimum hardware and software requirements for DB2® Data Links Manager on all supported operating systems.

You can install DB2 Data Links Manager on the following operating systems:

- AIX®
- Solaris™ Operating Environment
- Windows® NT
- Windows 2000

DB2 Data Links Manager is a 32-bit application. Ensure that your operating system is enabled to run in 32-bit mode, even if the hardware that you are using supports more than 32 bits.

Table 3 lists the minimum memory, storage space, and software that you need for DB2 Data Links Manager on each supported operating system.

Table 3. Minimum hardware and software for DB2 Data Links Manager

Operating system	Hardware[1]	Software
AIX[2]	• IBM® RISC/6000; or an IBM eServer™ pSeries™ • 256 MB RAM • 255 MB total storage space [3], including: – 170 MB storage space in the /usr/opt directory – 85 MB storage space in the DB2 Data Links Manager Administrator's home directory	• AIX Operating System Version 4.3.3 or later; AIX Version 5L (Version 5.1) for PPC; or AIX Version 5.2 • IBM DB2 Universal Database™ Version 8.1 [4] • Tivoli® Space Manager Version 4.2 (optional) • A supported storage manager program (optional). These programs include: – Tivoli Storage Manager Version 4.2.0 or later – An XBSA-compliant storage application, such as Legato NetWorker
Solaris Operating Environment	• Solaris UltraSPARC-based computer • 256 MB RAM • 422 MB total storage space [3], including: – 372 MB storage space in the /opt/IBM/db2 directory – 50 MB storage space in the DB2 Data Links Manager Administrator's home directory	• Solaris Operating Environment Version 7 or Version 8 with 32-bit kernel • IBM DB2 Universal Database Version 8.1 [4] • A supported storage manager program (optional). These programs include: – Tivoli Storage Manager Version 4.2.0 or later – An XBSA-compliant storage application, such as Legato NetWorker

Table 3. Minimum hardware and software for DB2 Data Links Manager (continued)

Operating system	Hardware[1]	Software
Windows NT®	• Intel Pentium® or Pentium-compatible CPU • 192 MB RAM • 205 MB total storage space [3]	• Windows NT Version 4.0 with Service Pack 6 or later • IBM DB2 Universal Database Version 8.1 [4] • A supported storage manager program (optional). These programs include: – Tivoli Storage Manager Version 4.2.0 or later – An XBSA-compliant storage application, such as Legato NetWorker
Windows 2000	• Intel Pentium or Pentium-compatible CPU • 192 MB RAM • 205 MB total storage space [3]	• Windows 2000 (5.00.2195) with Service Pack 1 or later, RC 1.1 • IBM DB2 Universal Database Version 8.1 [4] • A supported storage manager program (optional). These programs include: – Tivoli Storage Manager Version 4.2.0 or later – An XBSA-compliant storage application, such as Legato NetWorker

Notes:

1. You might require more memory and storage space depending on the configuration of your system and the processing requirements at your site. Also, the system requirements for your DB2 database might periodically change if your use of DB2 Data Links Manager increases over time.
2. You will need more storage and memory if you plan to use a High Availability Cluster Multiprocessor (HACMP) environment with DB2 Data Links Manager. See the *HACMP for AIX Installation Guide*, the *HACMP for AIX Planning Guide*, and the Related links at the end of this topic for more information.
3. The total storage space is the minimum amount of disk space required for the Data Links Manager software, the DB2 software, and the Data Links File Manager database instance (by default, DLFM_DB) when setting up and installing DB2 Data Links Manager on the operating system.
4. You can use DB2 Data Links Manager Version 8 with DB2 Universal Database Version 6.1, Version 7.1, or Version 7.2. However, to use the latest database functionality and features with DB2 Data Links Manager Version 8, you should migrate to DB2 Universal Database Version 8.1.

The Data Links server and the DB2 server can reside on separate machines and can run under different operating systems.

DB2 Data Links Manager employs a DB2 database as a "Logging Manager." This DB2 database is created, installed, and maintained automatically. You need to consider the system requirements for this DB2 database as your use of DB2 Data Links Manager increases over time.

On AIX systems and Solaris Operating Environments: If you use the Network File System (NFS) protocol and want to use secured update-in-place operations to enable changes to linked files, you must use NFS 3.

Related concepts:

- "DB2 Data Links Manager" on page 4
- "DB2 Data Links Manager typical setups" on page 12
- "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:

- "Using Tivoli Storage Manager as an archive server (AIX)" on page 35
- "Using Tivoli Storage Manager as an archive server (Solaris Operating Environment)" on page 37
- "Using Tivoli Storage Manager as an archive server (Windows)" on page 38
- "Setting up high availability support with the Data Links server (AIX)" on page 161

Part 2. Administration Guide and Reference

Chapter 4. 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® 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® 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 34
- "Size considerations" on page 34
- "Using an XBSA-compliant storage manager" on page 41

Related tasks:

- "Using Tivoli Storage Manager as an archive server (AIX)" on page 35
- "Using Tivoli Storage Manager as an archive server (Solaris Operating Environment)" on page 37
- "Using Tivoli Storage Manager as an archive server (Windows)" on page 38

- "Using a local directory for the archive area" on page 40
- "Setting the number of copy processes" on page 65
- "Retrieving archive server backup file information" on page 93

Related reference:
- "dlfm retrieve command" on page 224

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 4 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 4. 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]	**Recommendation:** Use the default value of zero (0) to avoid the extra space required for multiple copies of the same file.
dl_time_drop	The interval of time, in days, files will be retained in the archive area after a **DROP DATABASE** command is issued.	1 [0-365]	A value of zero (0) means that the files are deleted immediately from the archive server when a **DROP DATABASE** command is issued.

Related reference:

- "dl_num_copies - Data Links number of copies configuration parameter" in the *Administration Guide: Performance*
- "dl_time_drop - Data Links time after drop configuration parameter" in the *Administration Guide: Performance*
- "UPDATE DATABASE MANAGER CONFIGURATION Command" in the *Command Reference*
- "num_db_backups - Number of database backups configuration parameter" 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 machine on which you installed the Tivoli Storage Manager 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 userprofile or usercshrc 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/dldump
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 ${HOME}/dldump
setenv PATH=${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 37
- "Using Tivoli Storage Manager as an archive server (Windows)" on page 38
- "Using a local directory for the archive area" on page 40

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 machine on which you installed the Tivoli Storage Manager 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 userprofile or usercshrc 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 35
- "Using Tivoli Storage Manager as an archive server (Windows)" on page 38
- "Using a local directory for the archive area" on page 40

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 machine on which you installed the Tivoli Storage Manager 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:

   DSMI_DIR
   Specify the complete path to the appropriate Tivoli Storage Manager client application directory on your Data Links server computer. The default installation path is C:\Program Files\Tivoli\TSM\baclient.

   DSMI_CONFIG
   Specify the complete path to the dsm.opt file (the Tivoli Storage Manager user options file). The default installation path is C:\Program Files\Tivoli\TSM\baclient\dsm.opt.

   DSMI_LOG
   Specify any existing path on your Data Links server computer. A sample path for DSMI_LOG is c:\tsm\dldump.
5. Ensure that the dsm.opt file (Tivoli Storage Manager user options file) is located in the directory that is pointed to by the DSMI_DIR environment variable.
6. Set the *PASSWORDACCESS* option to `generate` in the dsm.opt 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 Data Links File Manager initiates a connection to the TSM server. For more information, see your TSM product documentation.
8. 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.

9. Restart the DLFM by invoking the **dlfm restart** command.

Related tasks:

- "Using Tivoli Storage Manager as an archive server (AIX)" on page 35
- "Using Tivoli Storage Manager as an archive server (Solaris Operating Environment)" on page 37
- "Using a local directory for the archive area" on page 40

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/dlfm/dlfm_backup_dir
```

5. Restart the DLFM with the **dlfm start** command.

Related concepts:

- "Data Links server file backups" on page 33

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® systems: You must have root authority.

Windows®: You must use a system administrator account (for example, the dlmadmin 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® Data Links Manager Administrator account configuration.
   - On UNIX systems, you add the required environment variables to the Data Links Manager Administrator account's userprofile or usercshrc 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\libxbsa.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 35
- "Using Tivoli Storage Manager as an archive server (Solaris Operating Environment)" on page 37
- "Using Tivoli Storage Manager as an archive server (Windows)" on page 38
- "Using a local directory for the archive area" on page 40

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 43
- "Authorizations for replicating linked files" on page 44

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® 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 99
- "Approach 1: Unlink/update/relink files" on page 111
- "Approach 2: Update linked files/native file system defines access" on page 112
- "Approach 3: Update linked files/customize file access privileges" on page 113
- "Approach 4: Replace a linked file's contents" on page 124

Related tasks:

- "Replicating special data types in SQL replication" in the *IBM DB2 Information Integrator SQL Replication Guide and Reference*
- "Enabling the Data Links Manager Replication daemon" on page 60

Related reference:

- "dlfm deny command" on page 211
- "dlfm grant replication read command" on page 216
- "dlfm grant replication write request command" on page 217
- "dlfm grant command" on page 214
- "dlfm list registered replication access control command" on page 220
- "dlfm list registered users command" on page 220
- "dlfm list registered directories command" on page 218
- "dlfm revoke command" on page 225
- "dlfm revoke replication command" on page 226
- "dlfm set link security command" on page 228

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 58

Related tasks:

- "Creating additional Data Links File Systems on a Data Links server" on page 47
- "Modifying a Data Links File System" on page 52
- "Adding DLFMs to a specific DB2 database on the DB2 host" on page 56
- "Adding DB2 databases to a DLFM on a Data Links server" on page 54
- "Enabling the Data Links Manager Replication daemon" on page 60

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 5, "Working with the Data Links File Manager," on page 67.

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 42

Related tasks:

- "Identifying DLFS paths to contain linked files" on page 48
- "Creating a new UNIX file system (optional)" on page 49
- "Activating the Data Links Filesystem Filter" on page 51
- "Enabling file system sharing (AIX, Solaris Operating Environment)" on page 50
- "Enabling file system sharing (Windows)" on page 51

Related reference:

- "Basic DLFM configuration tasks" on page 45

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 share name (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 47
- "Activating the Data Links Filesystem Filter" on page 51
- "Enabling file system sharing (AIX, Solaris Operating Environment)" on page 50
- "Enabling file system sharing (Windows)" on page 51

Related reference:

- "Basic DLFM configuration tasks" on page 45

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 **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.
   - 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 47
- “Activating the Data Links Filesystem Filter” on page 51
- “Enabling file system sharing (AIX, Solaris Operating Environment)” on page 50
- “Enabling file system sharing (Windows)” on page 51

Related reference:

- “Basic DLFM configuration tasks” on page 45

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 47

- "Identifying DLFS paths to contain linked files" on page 48
- "Creating a new UNIX file system (optional)" on page 49
- "Activating the Data Links Filesystem Filter" on page 51
- "Enabling file system sharing (Windows)" on page 51

Related reference:
- "Basic DLFM configuration tasks" on page 45

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 47
- "Identifying DLFS paths to contain linked files" on page 48
- "Creating a new UNIX file system (optional)" on page 49
- "Activating the Data Links Filesystem Filter" on page 51
- "Enabling and registering file systems with DLFM (AIX, Solaris Operating Environment)" on page 70

Related reference:
- "Basic DLFM configuration tasks" on page 45

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 70
- "Registering a drive with DLFF (Windows operating systems)" on page 72
- "Creating additional Data Links File Systems on a Data Links server" on page 47
- "Identifying DLFS paths to contain linked files" on page 48
- "Creating a new UNIX file system (optional)" on page 49
- "Enabling file system sharing (AIX, Solaris Operating Environment)" on page 50
- "Enabling file system sharing (Windows)" on page 51

Related reference:

- "Basic DLFM configuration tasks" on page 45

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 53

Related reference:

- "Tivoli Storage Manager" in the *Data Recovery and High Availability Guide and Reference*
- "Basic DLFM configuration tasks" on page 45

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 dlmadmin 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/dlfmfsmd -j filesystem_name  //AIX
   /opt/IBM/db2/V8.1/instance/dlfmfsmd -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 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)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/dlfmfsmd dlfm_mountpoint //AIX
   /opt/IBM/db2/V8.1/instance/dlfmsmd 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 74
- "Loading, querying, and unloading a DLFF (Solaris Operating Environment)" on page 74
- "Increasing a DLFF-controlled file system size (AIX, Solaris Operating Environment)" on page 76

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:

- "Authorizing users to perform actions" on page 42

Related reference:

- "UPDATE DATABASE MANAGER CONFIGURATION Command" in the *Command Reference*
- "datalinks - Enable Data Links support configuration parameter" in the *Administration Guide: Performance*
- "ADD DATALINKS MANAGER Command" in the *Command Reference*
- "dlfm add_db command" on page 209
- "Basic DLFM configuration tasks" on page 45

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 42

Related tasks:

- "Creating additional Data Links File Systems on a Data Links server" on page 47

Related reference:

- "UPDATE DATABASE MANAGER CONFIGURATION Command" in the *Command Reference*
- "datalinks - Enable Data Links support configuration parameter" in the *Administration Guide: Performance*
- "ADD DATALINKS MANAGER Command" in the *Command Reference*
- "dlfm add_db command" on page 209
- "Basic DLFM configuration tasks" on page 45

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® 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 78

Related reference:

- "DROP DATALINKS MANAGER Command" in the *Command Reference*
- "Basic DLFM configuration tasks" on page 45
- "dlfm drop_dlm command" on page 213

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 SQL replication" in the *IBM DB2 Information Integrator SQL Replication Guide and Reference*

Related reference:

- "db2set - DB2 Profile Registry Command" in the *Command Reference*
- "dlfm grant replication read command" on page 216
- "dlfm grant replication write request command" on page 217
- "dlfm revoke replication command" on page 226

Server environment management

When you change the host name of a DB2® host computer or a Data Links server computer, you must modify certain settings in your Data Links Manager environment.

For example, you might need to change the host name of a computer because of a planned computer outage. Or, you might need to automate the task of changing host names if you use a failover mechanism, such as a High Availability Cluster Multiprocessor (HACMP) configuration.

Use the **db2dlm_upd_hostname** utility to easily and safely change the required Data Links Manager settings on any computer in your Data Links Manager environment. The specific computer where you use the **db2dlm_upd_hostname** utility depends on the function that the computer you want to change is performing in your Data Links Manager environment.

If the host name of a DB2 host computer changes, use the **db2dlm_upd_hostname** utility on the Data Links server computer or computers where the files that are referenced in the DB2 host databases are stored.

If the host name of a Data Links server computer changes, use the **db2dlm_upd_hostname** utility on the DB2 host or DB2 hosts with the databases that reference the files that are stored on that Data Links server computer.

Related tasks:

- "Changing environment settings for a DB2 host in a Data Links Manager environment" on page 61
- "Changing environment settings for a Data Links server computer" on page 62

Changing environment settings for a DB2 host in a Data Links Manager environment

When you change the host name of a Data Links server computer, use this procedure to modify the Data Links Manager environment settings on the corresponding DB2 host computer. The DB2 host computer references files that are stored on the Data Links server.

Repeat this procedure for each Data Links server computer that has changed in your Data Links Manager environment.

Prerequisites:

- You must have *sysadm* authority on the required DB2 host computer.
- You must know both the existing (old) and the new (changed) fully qualified names of the Data Links server computer. An example of a fully qualified computer name is region1.dev.bigco.com.
- You must be attached to the instance where the DB2 host database is located.

Procedure:

To change the host name of a Data Links Manager server on a DB2 host computer:

1. Using an account with *sysadm* authority, log on to the DB2 host computer where the host database that references the changed Data Links server computer is located.
2. Run the **db2dlm_upd_hostname** utility.
   - On UNIX systems, **db2dlm_upd_hostname** is located in the *INSTHOME*/sqllib/bin directory, where *INSTHOME* is the home directory of the instance owner.
   - On Windows systems, **db2dlm_upd_hostname** is located in the *x:*\sqllib\bin directory, where *x:* is the drive where you installed DB2 Data Links Manager.
3. Reset your database connections, if necessary, by using the **db2 connect reset** command.

This procedure updates to the datalink.cfg file, which is located in each of the database directories on the DB2 host.

Related concepts:

- "Server environment management" on page 61

Related tasks:

- "Changing environment settings for a Data Links server computer" on page 62

Related reference:

- "db2dlm_upd_hostname Command" in the *Command Reference*

Changing environment settings for a Data Links server computer

When you change the host name of a DB2 host computer, use this procedure to modify the Data Links Manager environment settings on the Data Links server where files that are referenced by the host databases are stored.

Repeat this procedure for each DB2 host computer that has changed in your Data Links Manager environment.

Prerequisites:

- You must have Data Links Manager Administrator authority.
- You must know both the existing (old) and the new (changed) fully qualified names of the DB2 host computer. An example of a fully qualified computer name is region2.myco.com.

- You must be attached to the database instance where the Data Links File Manager (DLFM) is located.

Procedure:

To change the host name of a DB2 host on a Data Links server computer:

1. As the Data Links Manager Administrator, log on to the Data Links server computer that contains a linked file that is referenced in a database on the DB2 host computer that has changed.
2. Run the **db2dlm_upd_hostname** utility.
   - On UNIX systems, **db2dlm_upd_hostname** is located in the *INSTHOME*/sqllib/bin directory, where *INSTHOME* is the home directory of the instance owner.
   - On Windows systems, **db2dlm_upd_hostname** is located in the *x:*\sqllib\bin directory, where *x:* is the drive where you installed DB2 Data Links Manager.

This procedure changes the name of the DB2 host computer that is stored in the DLFM database (called DLFM_DB by default) on the Data Links server computer.

Related concepts:

- "Server environment management" on page 61

Related tasks:

- "Changing environment settings for a DB2 host in a Data Links Manager environment" on page 61

Related reference:

- "db2dlm_upd_hostname Command" in the *Command Reference*

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 64
- "Ensuring sufficient DB2 log space for the DLFM_DB" on page 64

Related tasks:

- "Setting the number of copy processes" on page 65

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:

- "dl_expint - Data Links access token expiry interval configuration parameter" in the *Administration Guide: Performance*
- "dl_wt_iexpint - Data Links write token initial expiry interval configuration parameter" in the *Administration Guide: Performance*

Ensuring sufficient DB2 log space for the DLFM_DB

Like all DB2® databases, the Data Links File Manager (DLFM) 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 33

Related reference:

- "db2set - DB2 Profile Registry Command" in the *Command Reference*

Chapter 5. 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 209.
- 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 233.

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 FS attribute)
- 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® 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.

Starting the DLFM automatically

You can, at your option, automatically start the DLFM whenever you reboot a Data Links server machine.

On UNIX® systems, use the DLFM_AUTOSTART registry variable. When the DLFM_AUTOSTART registry variable is set to YES, the **dlfm start** command is automatically issued at startup time. The DLFM_AUTOSTART registry variable is set to NO by default at installation time.

To change the DLFM_AUTOSTART registry variable, log on to a specific Data Links server machine as the Data Links Manager Administrator, and use the **db2set** command to set the DLFM_AUTOSTART registry variable to YES. The configuration change takes affect when the machine is rebooted.

On Windows systems, use the **Services** control panel to set the **Startup type** to **Automatic** for the **Data Links File Manager** process. The configuration change takes affect when the machine is rebooted.

Related concepts:

- "DB2 Data Links Manager recovery scenarios" on page 182

Related tasks:

- "Restarting the DLFM after an abnormal termination" on page 68

Related reference:

- "dlfm restart command" on page 224
- "dlfm shutdown command" on page 229
- "dlfm start command" on page 230
- "dlfm stop command" on page 230

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 182
- "Basic operations: starting, stopping, and restarting the DLFM" on page 67

Related reference:

- "dlfm shutdown command" on page 229
- "dlfm start command" on page 230
- "dlfm stop command" on page 230

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_delgrpd_(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_upcalld_(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 228

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 234
- "dlff refreshtrace command (Windows operating system)" on page 235
- "dlff set loglevel command (Windows operating system)" on page 239

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/dlfmfsmd dlfm_mountpoint   //Solaris Operating Environments
   /usr/opt/db2_08_01/instance/dlfmfsmd dlfm_mountpoint //AIX systems
   ```

   where *dlfm_mountpoint* is the mount point from the previous step.

   As part of the mounting process, the `dlfmfsmd` 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/rdsk/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 73

Related reference:

- "dlfm add_prefix command" on page 210
- "dlfm start command" on page 230

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 \share name
   ```

   where *share name* is the name of an existing share name on the drive you just added. For example, if the share name 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 share name, and that drive does yet not have a share name associated with it, you must associate a share name 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 share name.

Related tasks:

- "Listing registered drives (Windows operating systems)" on page 73

Related reference:

- "dlff add command (Windows operating system)" on page 233
- "dlff list command (Windows operating system)" on page 235

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 70

Related reference:

- "dlfm list registered prefixes command" on page 219

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 72

Related reference:

- "dlff list command (Windows operating system)" on page 235
- "dlfm list registered prefixes command" on page 219

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 70
- "Listing registered file systems (AIX, Solaris Operating Environment)" on page 73

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 70
- "Listing registered file systems (AIX, Solaris Operating Environment)" on page 73

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 72
- "Listing registered drives (Windows operating systems)" on page 73

Related reference:

- "dlff add command (Windows operating system)" on page 233
- "dlff remove command (Windows operating system)" on page 236
- "dlff list command (Windows operating system)" on page 235

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/dlfmfsmd -j filesystem_name  //AIX systems
   /opt/IBM/db2/V8.1/instance/dlfmfsmd -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/dlfmfsmd dlfm_mountpoint  //AIX systems
/opt/IBM/db2/V8.1/instance/dlfmsmd 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 70
- "Listing registered file systems (AIX, Solaris Operating Environment)" on page 73

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 78

Related reference:

- "dlfm add_db command" on page 209

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 77

Related reference:

- "dlfm list registered databases command" on page 218

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 213

Changing the diagnostic level for error message log files

Several error message log files are maintained in the Data Links Manager environment, both on Data Links servers and at DB2 host databases 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. Therefore, 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 on DB2 host and Data Links server computers by using the *DIAGLEVEL* configuration parameter.

The *DIAGLEVEL* configuration parameter 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.

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 concepts:

- "DB2 trace (db2trc)" in the *Troubleshooting Guide*
- "DB2 traces" on page 190

Related reference:

- "diaglevel - Diagnostic error capture level configuration parameter" in the *Administration Guide: Performance*
- "diagpath - Diagnostic data directory path configuration parameter" in the *Administration Guide: Performance*
- "db2trc - Trace Command" in the *Command Reference*

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 `dlfs_cfg` file. The `dlfs_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 `dlfs_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 `dlfm`.

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 `dlfmxgrp`.

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 81

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 80

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 84
- "Migrating a DLFS to a different hard disk (Solaris Operating Environment)" on page 87
- "Migrating a DLFS to a different drive (Windows)" on page 90

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/newdlfslv 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/sqllib/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 70

Related reference:

- "db2_recon_aid - RECONCILE Multiple Tables Command" in the *Command Reference*
- "db2_recon_aid utility" on page 180

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 filesys | 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/sqllib/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 70

Related reference:

- "db2_recon_aid - RECONCILE Multiple Tables Command" in the *Command Reference*
- "db2_recon_aid utility" on page 180

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:\PROGRA~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:*\sqllib\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 180

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® 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 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 all of 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 90

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 dlfm_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:

- "dlfm retrieve command" on page 224

Chapter 6. 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 99
- "Read operation security" on page 103
- "Write operation security" on page 103
- "Security issues for updating linked files" on page 105

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® 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® 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).

Important: A DB2® 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. 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:

- "DB2 database and DB2 Data Links Manager file access controls" on page 19
- "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 5 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 5. Prohibited linked file operations

DATALINK column access permission attributes[1]	**Prohibited linked file operations**
• READ PERMISSION DB • WRITE PERMISSION BLOCKED or WRITE PERMISSION ADMIN	Read/delete/rename/write and set file attributes.[2] • The read operation is prohibited *only* if a valid access token is not specified. • The write operation is permitted *only* if a DATALINK column is defined with WRITE PERMISSION ADMIN and a valid write token is specified.
• READ PERMISSION FS • WRITE PERMISSION BLOCKED	Delete/rename files and reset the R/O flag attribute.
• READ PERMISSION FS • WRITE PERMISSION FS	Delete/rename files.

Notes:

1. The DATALINK column must also be defined with the FILE LINK CONTROL attribute.
2. For Windows® only: *Any* Administrator user account, not just the dlmadmin account, can view a linked file's permissions without using a read or write token. However, only the dlmadmin user account can read files.

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 216
- "dlfm grant replication write request command" on page 217
- "dlfm grant command" on page 214
- "dlfm list registered replication access control command" on page 220
- "dlfm list registered directories command" on page 218
- "dlfm set link security command" on page 228

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® 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 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

Possible column attribute	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 99

Related reference:

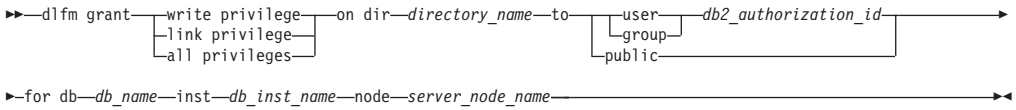
- "dlfm deny command" on page 211
- "dlfm grant command" on page 214
- "dlfm revoke command" on page 225
- "dlfm set link security command" on page 228

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:



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    app1     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 103
- "Write operation security" on page 103
- "Security issues for updating linked files" on page 105

Related reference:

- "dlfm deny command" on page 211
- "dlfm grant command" on page 214
- "dlfm list registered prefixes command" on page 219
- "dlfm list registered users command" on page 220

- "dlfm list registered directories command" on page 218
- "dlfm revoke command" on page 225
- "dlfm set link security command" on page 228

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® 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 99
- "Write operation security" on page 103
- "Security issues for updating linked files" on page 105

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® 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 103
- "Security issues for updating linked files" on page 105
- "Updating approaches overview" on page 109
- "Approach 1: Unlink/update/relink files" on page 111
- "Approach 2: Update linked files/native file system defines access" on page 112
- "Approach 3: Update linked files/customize file access privileges" on page 113
- "Approach 4: Replace a linked file's contents" on page 124

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® host database DATALINK column definitions impose.

Related concepts:

- "Advanced file management security rules and guidelines" on page 99
- "Write operation security" on page 103
- "Updating approaches overview" on page 109
- "Approach 1: Unlink/update/relink files" on page 111
- "Approach 2: Update linked files/native file system defines access" on page 112
- "Approach 3: Update linked files/customize file access privileges" on page 113
- "Approach 4: Replace a linked file's contents" on page 124

Related reference:

- "dlfm deny command" on page 211
- "dlfm grant command" on page 214
- "dlfm revoke command" on page 225
- "dlfm set link security command" on page 228

Part 3. Application Programmer's Reference

Chapter 7. 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® 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 111
- "Approach 2: Update linked files/native file system defines access" on page 112
- "Approach 3: Update linked files/customize file access privileges" on page 113
- "Approach 4: Replace a linked file's contents" on page 124

Related reference:

- "Updating approaches summary" on page 127

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, and so on.
- 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 112
- "Approach 3: Update linked files/customize file access privileges" on page 113
- "Approach 4: Replace a linked file's contents" on page 124

Related tasks:

- "Unlinking and relinking files" on page 111

Related reference:

- "dlfm grant command" on page 214
- "dlfm set link security command" on page 228
- "Updating approaches summary" on page 127

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 109
- "Approach 1: Unlink/update/relink files" on page 111

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 127

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 111
- "Approach 3: Update linked files/customize file access privileges" on page 113
- "Approach 4: Replace a linked file's contents" on page 124

Related reference:

- "Updating approaches summary" on page 127

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® 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 118
- "Usage considerations for write tokens" on page 122

Related tasks:

- "Approach 3 configuration tasks" on page 115
- "Approach 3 maintenance tasks" on page 116

Related reference:

- "dlfm grant command" on page 214
- "Updating approaches summary" on page 127

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 7. 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. • If you use the NOT REQUIRING TOKEN FOR UPDATE attribute, you only need the write token value when opening a linked file. • 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.
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 118

Related tasks:

- "Approach 3 maintenance tasks" on page 116

Related reference:

- "dlfm grant command" on page 214

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 111
- "Approach 2: Update linked files/native file system defines access" on page 112
- "Approach 3 application program flow" on page 118
- "Approach 4: Replace a linked file's contents" on page 124

Related tasks:

- "Backing out updates in progress" on page 121

- "Recovering missing write tokens" on page 123

Related reference:
- "dlfm deny command" on page 211
- "dlfm grant command" on page 214
- "dlfm list registered users command" on page 220
- "dlfm list upd_in_progress files for db command" on page 221
- "dlfm list upd_in_progress files for prefix command" on page 222
- "dlfm list registered directories command" on page 218
- "dlfm revoke command" on page 225
- "Updating approaches summary" on page 127

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 5 on page 119 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.

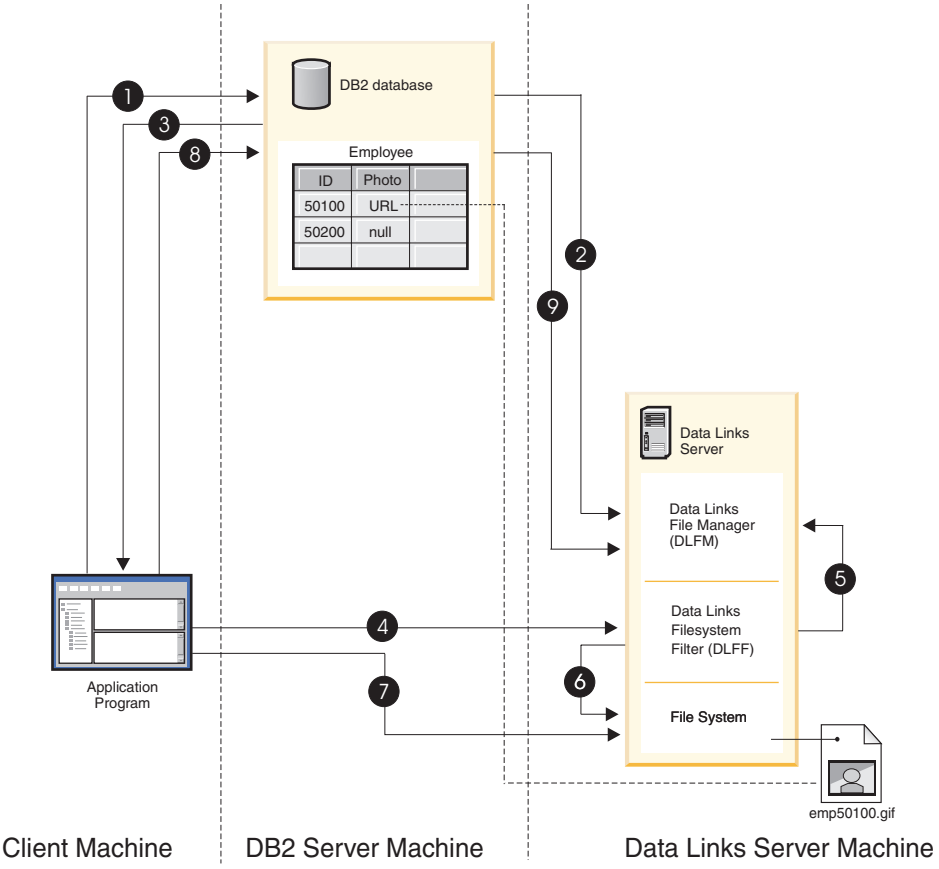


Figure 5. Approach 3 application program flow

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 113
- "Usage considerations for write tokens" on page 122

Related tasks:

- "Backing out updates in progress" on page 121
- "Recovering missing write tokens" on page 123

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 113
- "Usage considerations for write tokens" on page 122

Related tasks:

- "Recovering missing write tokens" on page 123

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 113
- "Approach 3 application program flow" on page 118

Related tasks:

- "Approach 3 configuration tasks" on page 115

- "Recovering missing write tokens" on page 123

Related reference:

- "dl_wt_iexpint - Data Links write token initial expiry interval configuration parameter" 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 **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_IBAuKyZLDII;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 113
- "Approach 3 application program flow" on page 118
- "Usage considerations for write tokens" on page 122

Related reference:

- "dlfm list upd_in_progress files for db command" on page 221
- "dlfm list upd_in_progress files for prefix command" on page 222

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 111
- "Approach 2: Update linked files/native file system defines access" on page 112
- "Approach 3: Update linked files/customize file access privileges" on page 113
- "Approach 4 Usage Scenario" on page 126

Related reference:
- "UPDATE statement" in the *SQL Reference, Volume 2*
- "DLREPLACECONTENT scalar function" in the *SQL Reference, Volume 1*
- "dlfm grant command" on page 214
- "dlfm set link security command" on page 228
- "Updating approaches summary" on page 127

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 111
- "Approach 2: Update linked files/native file system defines access" on page 112

- "Approach 3: Update linked files/customize file access privileges" on page 113
- "Approach 4: Replace a linked file's contents" on page 124
- "Approach 4 Usage Scenario" on page 126

Related reference:
- "UPDATE statement" in the *SQL Reference, Volume 2*
- "DLREPLACECONTENT scalar function" in the *SQL Reference, Volume 1*
- "Updating approaches summary" on page 127

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 111
- "Approach 2: Update linked files/native file system defines access" on page 112
- "Approach 3: Update linked files/customize file access privileges" on page 113
- "Approach 4: Replace a linked file's contents" on page 124

Related reference:

- "UPDATE statement" in the *SQL Reference, Volume 2*
- "DLREPLACECONTENT scalar function" in the *SQL Reference, Volume 1*
- "Updating approaches summary" on page 127

Updating approaches summary

Table 8 summarizes the DATALINK column definition requirements, restrictions, advantages, and disadvantages of all four updating approaches.

Table 8. Linked file updating approaches summary

Approach	Restrictions	DATALINK column requirements[1]	Benefits	Potential issues
1 - Unlink, update, then relink files	None	None	A simple approach to implement.	• 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. • 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, and so on. • 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 cannot perform updates-in-place.

Table 8. Linked file updating approaches summary (continued)

Approach	Restrictions	DATALINK column requirements[1]	Benefits	Potential issues
2 - Update linked files/native file system defines access		• WRITE PERMISSION attribute set to FS • RECOVERY attribute set to NO	• Simple approach to implement. • 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.	• 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. • Cannot back out changes made to linked files. • 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. • 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.

Table 8. Linked file updating approaches summary (continued)

Approach	Restrictions	DATALINK column requirements[1]	Benefits	Potential issues
3 - Update linked files/customize file access	If using Network File System (NFS), must be at version 3.0 or later	• WRITE PERMISSION attribute set to ADMIN • READ PERMISSION attribute set to DB • To back out file changes, column must also be defined with the RECOVERY attribute set to YES	• Authorization to modify linked files can be controlled at a very fine level. • 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. • 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. • DB2 coordinated backup and recovery options supported. • Can back out file changes if necessary, if DATALINK table column defined with the RECOVERY attribute set to YES.	• Multiple tasks necessary to implement this approach. • You must ensure that you have granted file writing privileges for the required users to the required Data Links server directories. • 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. • Cannot back out changes made to linked files if a table's DATALINK column is not defined with the RECOVERY attribute set to YES.
4 - Replace a linked file's contents	Replacement files must meet these requirements: • Must reside in the same directory as the linked files you want to update. • Files names must consist of the original file name plus a suffix string that can be a maximum of 20 characters.	None	• Simple approach to implement. • Linked file changes become visible only when committed in DB2, so if transaction fails or rolls back, original file contents remain unchanged.	• A replacement file is not under Data Links Manager or DB2 control until it replaces the original file. • Need additional disk space to accommodate replacement files. • You must ensure that you have granted file linking privileges for the required users to the required Data Links server directories.

Notes:

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 109
- "Approach 1: Unlink/update/relink files" on page 111

- "Approach 2: Update linked files/native file system defines access" on page 112
- "Approach 3: Update linked files/customize file access privileges" on page 113
- "Approach 4: Replace a linked file's contents" on page 124

Chapter 8. Replicating linked files with DB2 DataPropagator

This chapter describes how to configure and use DB2 Data Links Manager with DB2 DataPropagator to replicate (copy) linked files from one location to another.

Replication is the process of automatically maintaining one or more copies of data so that the copied data is synchronized with the original source data. When data is created, updated, or deleted at the source, the copied data is also changed. When you replicate linked files, the linked file *and* the database data, are replicated together as a single entity.

Replicating linked files enables you to copy files that reside outside of a database together with the metadata for the linked files. Thus you maintain the relationship between the copied metadata and the copied linked files.

An overview of how linked files get replicated

DB2® DataPropagator™ automatically maintains one or more copies of data and keeps the copied data (target data) synchronized with the original source data. When data at the source is created, updated, or deleted, the changes are also replicated to the target data.

You can use the DATALINK data type to represent data stored in external file systems. You can then use DB2 replication to replicate DATALINK column values with their associated linked files and other DB2 relational data. DATALINK column values contain:

- The names of the servers on which the files reside
- The fully qualified path names to the files

The Data Links Manager replication daemon (DLFM_ASNCOPYD), which works with the replication ASNDLCOPY exit routine, copies the external files referenced by the DATALINK column values from the source file system to the target file system.

This topic describes how DATALINK column values and their associated linked files are replicated with the ASNDLCOPY exit routine.

This topic includes the following sections:

The DATALINK data type replication process

Figure 6 on page 132 illustrates the replication process that is used to replicate DATALINK column values and other database data.

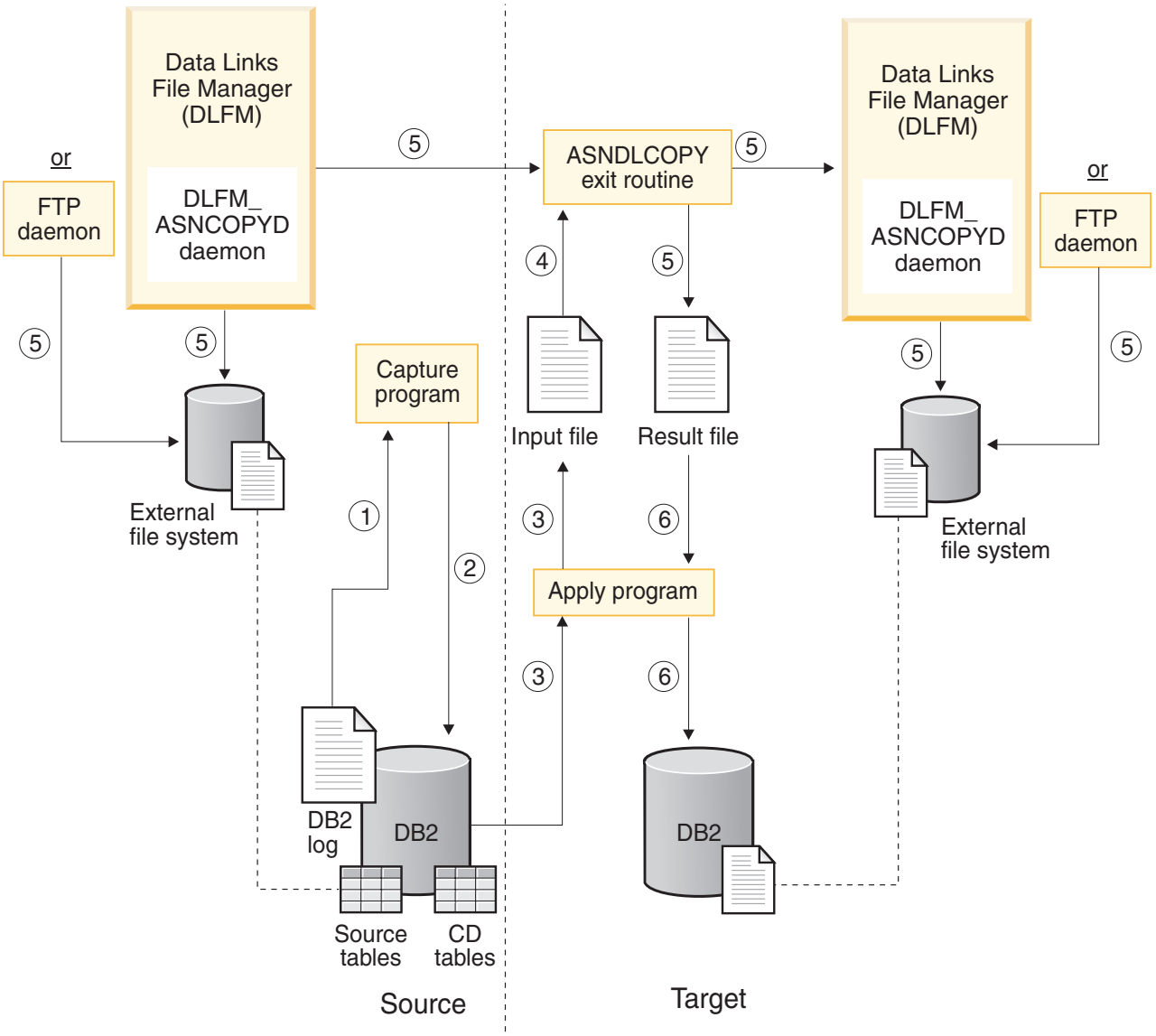


Figure 6. DATALINK replication process

The numbers in the diagram correspond to the following replication processing steps:

1. The Capture program reads the captured changes made to the source table data. These captured changes include any changes to DATALINK column values, which you can select as part of your registration for replication.
2. The Capture program writes the changed data to the change-data (CD) table as part of the differential refresh replication process.
3. When a subscription set is ready to be replicated during a differential refresh, the Apply program identifies the applicable rows in the CD table. (During a full refresh, the Apply program obtains the rows directly from the source table.) If the Apply program finds any DATALINK column values, it obtains the Uniform Resource Locator (URL) of each changed DATALINK value and places these URLs into the input file.
4. The Apply program invokes the ASNDLCOPY exit routine. The ASNDLCOPY exit routine reads the input file and maps each DATALINK source file location to its corresponding target file location.

5. The ASNDLCOPY exit routine connects either to the Data Links Manager replication daemon (DLFM_ASNCOPYD) or to the FTP daemon. The exit routine then replicates the referenced files from the source file system to the newly mapped target file system location and stores this target file system location in a result file.
6. After the ASNDLCOPY exit routine replicates all of the files and terminates, the Apply program writes the changes, including the updated URLs for the DATALINK column values, to the target table.

Note: The preceding diagram shows a typical configuration in which the Apply program runs on the target system. However, you can set up your replication environment to run the Apply program on the source system or other system as necessary.

The ASNDLCOPY exit routine

The ASNDLCOPY exit routine, which is part of the replication product, drives the replication of linked files identified by the DATALINK column values. The Apply program invokes the ASNDLCOPY exit routine if the Apply program identifies rows with DATALINK values in the CD table. The exit routine maps the URL on the source file system to the URL on the target file system. The exit routine then connects to the DLFM_ASNCOPYD or other file-copy daemon (for example, FTP) to replicate the referenced file.

A sample ASNDLCOPY exit routine is provided with the DB2 replication software. You can use the source code for this ASNDLCOPY exit routine and modify the sample program to meet the requirements of your system. The sample program is called ASNDLCOPY.smp.

On AIX® systems and Solaris™ Operating Environments, the sample program is located in the *INSTHOME*/sqllib/samples/repl directory, where *INSTHOME* is the home directory of the DB2 instance owner.

On Windows® NT and Windows 2000 systems, the sample program is located in the *x:*\sqllib\samples\repl directory, where *x:* is the drive where DB2 is installed.

See the PROLOG section of the sample program for more information on how to set up the configuration files and how to modify this exit routine.

The following sections describe the information that the Apply program passes to the ASNDLCOPY exit routine and the configuration files that the sample exit routine uses.

- "ASNDLCOPY parameters and input"
- "ASNDLCOPY configuration files" on page 135

ASNDLCOPY parameters and input

The Apply program passes the following information to the ASNDLCOPY exit routine:

- Subscription-set name
- Apply qualifier
- Source-table name
- Source-table owner
- Source-server name
- Target-table name
- Target-table owner

- Target-server name
- Input file
- Result file
- Trace option
- Source-server version
- Target-server version
- Apply path

The input file, result file, trace option, source-server version, target-server version, and Apply path contain specific information that is used by the ASNDLCOPY exit routine and by the Apply program when replicating linked files.

The input file contains a list of link references captured from the source tables. The Apply program transfers the information in this file to the ASNDLCOPY exit routine. This input file has the following format:

```
url-length source-link-reference new-link-indicator version-number
```

url-length
Indicates the length of the source link reference.

source-link-reference
Identifies the source link in URL format.

new-link-indicator
Represents a one-character link indicator. This link indicator specifies whether the DATALINK value is changed through a corresponding SQL operation in the source database and whether the changed file is replicated. A value of `Y` indicates that the changed file is replicated. A value of `N` indicates that the changed file is *not* replicated (for example, if a SQL UPDATE statement changes various column values, but not the DATALINK column values). Regardless of the value of the link indicator, the source file is mapped to its target file location.

version-number
Identifies the version number of the file to be replicated. The version number is optional and is included in the input file only if the corresponding DATALINK column is defined as RECOVERY YES. The ASNDLCOPY exit routine uses the version number to retrieve a particular backup version of the linked file through the DLFM_ASNCOPYD replication daemon.

Example input file:

```
37 HTTP://S1.SYS.COM/data/dir1/file1.gif Y 00000BFB4F503D3672D5
```

The result file contains transformed link references that are valid for the target system. The Apply program uses the references in this file when writing changes to the target table.

If the ASNDLCOPY exit routine detects that a replicated file already exists on the target file system, this exit routine replicates the source file contents to a temporary file. The temporary file, which is also called the *replacement file*, is given a name that is equal to the original source file name with an added suffix. The ASNDLCOPY exit routine retrieves this suffix, which is defined in the ASNDLPARM configuration file. The suffix and the target file reference are stored in the result file.

This result file has the following format:

```
url-length target-link-reference suffix-length replacement-file-suffix
```

url-length
Indicates the length of the target link reference.

target-link-reference
Identifies the target link in URL format.

suffix-length
Indicates the length of the suffix.

replacement-file-suffix
Identifies the suffix string that is added to the temporary replacement file.

Example result file:

```
37 HTTP://T1.SYS.COM/data/dir1/file1.gif 4 .new
```

Both the *suffix-length* and the *replacement-file-suffix* parameters are optional. The ASNDLCOPY exit routine writes these two parameters to the result file only if:

- The *replace_file* parameter in the ASNDLPARM configuration file is set to `YES`
- The replicated file already exists on the target file system

If both of these parameters are omitted, the Apply program uses the file specified in the *target-link-reference* parameter as the target file and does not move the file content to or from a temporary replacement file.

The trace option parameter can be either `yes` or `no`. If tracing is set to `yes`, the ASNDLCOPY exit routine creates a log file and a trace file. The trace file contains any trace information generated by the ASNDLCOPY exit routine.

The source-server version and the target-server version parameters represent the version numbers of the source and target DB2 servers, respectively.

The Apply path parameter specifies the directory where the Apply program stores its log files and its work files. You can specify an Apply path when you start the Apply program. (By default, the Apply path is the directory where you start the Apply program.)

ASNDLCOPY configuration files

The sample ASNDLCOPY exit routine uses three configuration files.

The three configuration files are:

- ASNDLSRVMAP
- ASNDLUSER
- ASNDLPARM

You can modify the sample ASNDLCOPY exit routine and its configuration files to enable replication of DATALINK column values at your site.

If you specify an Apply path parameter for the Apply program, you must place these configuration files in the Apply path directory. Otherwise, the configuration files must be in the working directory of the Apply program.

ASNDLSRVMAP

The ASNDLSRVMAP configuration file contains the server locations for link references. This file also contains optional directory paths to map the

source URL and directory paths to the target URL and directory paths. (The sample ASNDLCOPY exit routine initiates directory mapping for a given source link reference only if this ASNDLSRVMAP configuration file contains a directory mapping for that source link reference.)

Example ASNDLSRVMAP file:

```
HTTP://A.COM HTTP://B.COM /dir1 /dir2
HTTP://C.COM HTTP://D.COM /dir4 /dir5 /dir10 /dir11
```

Restriction: You can enter up to 50 source-to-target server mappings, each with a maximum of 10 source-to-target directory mappings, in the ASNDLSRVMAP file. Type each source-to-target server mapping on a single line. Use spaces to separate each server location and directory path, and press Enter only to create a new line for the next source-to-target server mapping.

ASNDLUSER

The ASNDLUSER configuration file contains login and address information. This information is used by the ASNDLCOPY exit routine to connect to the source and target file-copy daemons to replicate files. The ASNDLUSER configuration file has the following format:

server-address recv-port send-port userid passwd

server-address
Indicates the location of the server.

recv-port
Indicates the port number of either the DLFM_ASNCOPYD or the FTP file-copy daemon at the source file system. The ASNDLCOPY exit routine connects to the daemon through this port number when retrieving files from the source file system.

send-port
Indicates the port number of either the DLFM_ASNCOPYD or the FTP file-copy daemon at the target file system. The ASNDLCOPY exit routine connects to the daemon through this port number when sending and storing files to the target file system.

userid Specifies the login user ID. The given server uses the login user ID to authenticate access to the system.

passwd Specifies the password for the user ID.

Example ASNDLUSER file:

```
S1.ABC.COM 10000 -1 USERA XXXXYZ
T1.XYZ.COM -1 65535 USERB AAAXXY
```

When you enter information in the ASNDLUSER file for a *source* file server, you must enter the *recv-port* value. Typically, you would enter -1 (not applicable) for the *send-port* value. You need to specify a *send-port* value for sending and storing files only if this same file server is also a target file server in an update-anywhere replication configuration.

When you enter information in the ASNDLUSER file for a *target* file server, you must enter the *send-port* value. Typically, you would enter -1 (not applicable) for the *recv-port* value. You need to specify a *recv-port* value for retrieving files only if this same file server is also a source file server in an update-anywhere replication configuration.

The default port number value for the FTP daemon is 21. Specify this port number if you are using an FTP daemon and its default port number value.

When the Apply program invokes the ASNDLCOPY exit routine, the exit routine connects to the source and target file servers through the associated file-copy daemons (DLFM_ASNCOPYD or FTP). ASNDLCOPY uses the specified *recv-port* value to connect to the source file system and the specified *send-port* value to connect the target file system.

ASNDLPARM

The ASNDLPARM configuration file contains operational parameters used to control the function of the ASNDLCOPY exit routine. The ASNDLPARM configuration file has the following format:

```
parameter=value
```

parameter
Represents one of the supported ASNDLPARM parameter strings.

value Specifies one of the valid values for the specified parameter.

Example ASNDLPARM file:

```
replace_file=YES
replacement_file_suffix=.new
direct_copy=NO
```

ASNDLPARM is an optional configuration file. If the ASNDLPARM configuration file does not exist, the ASNDLCOPY exit routine operates using all default parameter settings. If you use a ASNDLPARM configuration file, you must place this file in the same directory as your other ASNDLCOPY configuration files.

Related concepts:

- "How the Data Links Manager replication daemon works" on page 137
- "Data replication example with DB2 Data Links Manager" on page 147

Related tasks:

- "Replicating special data types in SQL replication" in the *IBM DB2 Information Integrator SQL Replication Guide and Reference*
- "Operating the Apply program" in the *IBM DB2 Information Integrator SQL Replication Guide and Reference*
- "Enabling the Data Links Manager Replication daemon" on page 60
- "Setting up replication of linked files" on page 140

How the Data Links Manager replication daemon works

This topic describes the configuration of the Data Links Manager replication daemon (DLFM_ASNCOPYD) and how this daemon copies files that are referenced by DATALINK column values.

DLFM_ASNCOPYD is a daemon that enables the replication of DB2® Data Links Manager files. DLFM_ASNCOPYD, which is part of the Data Links File Manager (DLFM), uses standard FTP protocols and works with the ASNDLCOPY exit routine to retrieve, send, and store referenced files.

This topic includes the following sections:

- "DLFM_ASNCOPYD configuration"
- "DLFM_ASNCOPYD processing with replication" on page 139

DLFM_ASNCOPYD configuration

DB2 Data Links Manager launches the DLFM_ASNCOPYD daemon when Data Links starts if you enable this daemon. To enable DLFM_ASNCOPYD:

- Log on to the Data Links server using the Data Links Manager Administrator user ID.
- Set the DLFM_START_ASNCOPYD registry variable to `YES`.
- Set the DLFM_ASNCOPYD_PORT registry variable to the TCP-IP port number value. This value should match the port number that you specified in the ASNDLUSER configuration file.

  If you use DLFM_ASNCOPYD as your source file-copy daemon, the TCP-IP port number value is the *recv-port* value of the corresponding server entry in the ASNDLUSER configuration file. If you use DLFM_ASNCOPYD as your target file-copy daemon, the TCP-IP port number value is the *send-port* value.

  On AIX® systems and Solaris™ Operating Environments, this port number should be reserved in /etc/services.

  On Windows® NT and Windows 2000 systems, this port number should be reserved in \system32\drivers\etc\services.

You can set these registry variables during the DB2 Data Links Manager installation, or you can set them after the installation and then restart DLFM on a Data Links server.

After you enable the DLFM_ASNCOPYD daemon, it initiates the following actions:

- Listens to the port that is specified by the DLFM_ASNCOPYD_PORT registry variable
- Starts a child process to handle requests from the new connection after receiving a connection request from a dedicated port
- Verifies the login password
- Waits for incoming file-copy commands
- Verifies replication access permissions

If any DLFM_ASNCOPYD child process ends abnormally, DLFM_ASNCOPYD writes an error in the db2diag.log file, and the DLFM continues to run.

DLFM_ASNCOPYD supports the following FTP commands:

HELP Displays the supported commands.

HELP SITE
Displays all supported SITE commands and associated syntax.

MDTM
Displays last modification time, in Greenwich Mean Time (GMT), of a specified file.

MKD Creates the specified directory.

PASS Provides the login client with the login password.

PASV Sends a request to the FTP server to listen to the data port and wait for a connection rather than initiate a connection after receiving a transfer command.

PORT Specifies the number of the data port used when initiating a data connection.

RETR Sends a request to the FTP daemon to transfer a copy of a specified file.

SITE CHMOD
Changes the permissions of a specified file.

SITE TOUCH
Changes the last modification time of a file to the specified time (in GMT).

SITE UMASK
Sets the file-mode creation mask of the current session to the specified umask value.

SIZE Displays the size, in bytes, of the specified file.

STOR Accepts transferred data and stores this data in a file at the given location of the remote file server. This command overwrites the contents of the specified file if this file already exists. Otherwise, this command creates a new file.

TYPE Specifies how transferred data is represented, image (binary) or ASCII.

USER Specifies the login user id.

You must use the absolute path of any file or directory that you specify with an FTP command using the DLFM_ASNCOPYD daemon.

DLFM_ASNCOPYD processing with replication

During the replication process, the Apply program invokes the ASNDLCOPY exit routine to map the source Uniform Resource Locator (URL) to the target URL for each DATALINK value that the exit routine finds in the input file. If you specify the login and address information of the DLFM_ASNCOPYD daemon in the ASNDLUSER configuration file, the ASNDLCOPY exit routine connects to the DLFM_ASNCOPYD daemon.

DLFM_ASNCOPYD provides the interface for retrieving, sending, and storing linked files. DLFM_ASNCOPYD also provides functionality that is not available with the FTP daemon. This additional functionality includes:

- Retrieval of a particular version of a file that is referenced by a DATALINK column defined with RECOVERY YES
- Retrieval of files referenced by DATALINK columns defined as READ PERMISSION DB if the user has read permission access to the directory as granted by the **dlfm grant replication read** command
- The ability to preserve the last modification time of replicated files

When a source file is linked to a DATALINK column defined with RECOVERY YES, the DLFM tracks the version of the source file for coordinated file backup and recovery. The Capture program records the version number of the file each time it captures a change to a DATALINK column defined with RECOVERY YES. The Capture program then places that version number, along with the associated DATALINK column value information, into the change-data (CD) table.

DLFM_ASNCOPYD ensures the transfer of a consistent version of the external file from the source system to the target system. When the Apply program reads the

data changes from the CD table, it obtains the version number of the linked file. The Apply program then passes this version number and the URLs of the new DATALINK column values to the ASNDLCOPY exit routine. When the ASNDLCOPY exit routine connects to the DLFM_ASNCOPYD daemon, DLFM_ASNCOPYD retrieves a consistent version of the external file from the archive server. Therefore, both the database data and the replicated linked file are consistent.

Related concepts:

- "An overview of how linked files get replicated" on page 131
- "Data replication example with DB2 Data Links Manager" on page 147

Related tasks:

- "Replicating special data types in SQL replication" in the *IBM DB2 Information Integrator SQL Replication Guide and Reference*
- "Operating the Apply program" in the *IBM DB2 Information Integrator SQL Replication Guide and Reference*
- "Setting up replication of linked files" on page 140

Related reference:

- "dlfm grant replication read command" on page 216
- "dlfm grant replication write request command" on page 217

Setting up replication of linked files

This topic describes the steps that you need to take to set up your replication environment properly and to begin using the DLFM_ASNCOPYD replication daemon to replicate linked data.

Prerequisites:

- Verify that the DLFM_ASNCOPYD daemon is enabled. See the Related tasks at the end of this topic for more information about how to enable the Data Links Manager replication daemon.
- Check the following user authorization information in the target system:
  - Verify that the authorized users have the proper file system access permissions depending on the file link options of the DATALINK columns of the files being replicated:
    - If replicating files to a target DATALINK column that is defined as READ PERMISSION DB with a write permission of either WRITE PERMISSION BLOCKED or WRITE PERMISSION ADMIN, the file system permissions of each authorized user must be set so that only the authorized user can read or write to the files before the files are linked to the target database. After the replication process completes, these files are linked and owned by the DLFM user.
    - If replicating files to a target DATALINK column that is defined as READ PERMISSION FS, the file system permissions of each authorized user must be set so that other authorized users can access the files being replicated (created) by this authorized user.
  - Verify that the authorized user is *not* the DLFM user.
- Use the **dlfm grant replication read** command to enable authorized users to retrieve and obtain information about linked files.

- Use the **dlfm grant replication write request** command to enable authorized users to issue write request operations to the DLFM_ASNCOPYD daemon.
- If you are using an update-anywhere replication configuration, verify that these users have the proper access authorizations on both the source and target servers.

Procedure:

To set up the replication of linked data:

1. Modify the ASNDLCOPY exit routine to meet the requirements of your site. This exit routine maps the Uniform Resource Locators (URLs) of the source file system to the corresponding URLs on the target file system and then connects to the DLFM_ASNCOPYD or other file-copy daemon to replicate the referenced files. See the PROLOG section of the ASNDLCOPY.smp sample program for information on how to modify this exit routine.

   On AIX systems and Solaris™ Operating Environments, the sample program is located in the *INSTHOME*/sqllib/samples/repl directory, where *INSTHOME* is the home directory of the DB2 instance owner.

   On Windows NT and Windows 2000 systems, the sample program is located in the *x:*\sqllib\samples\repl directory, where *x:* is the drive where DB2 is installed.
2. Identify the source and target file servers in your environment, and obtain the login information and the associated source-to-target server and path mapping relationships.
3. Modify the ASNDLSRVMAP configuration file to specify the correct source-URL-to-target-URL mapping. For example:

   ```
   HTTP://S1.PRD.COM  HTTP://T1.PRD.COM  /data/dl_file_system/sd1  /data/dl_file_system/td1
   ```

   This example maps the source URL server of S1.PRD.COM to the target URL server of T1.PRD.COM. It also maps the source directory path of /data/dl_file_system/sd1 to the target directory path of /data/dl_file_system/td1.

   Each pair of mapping entries that you enter in the ASNDLSRVMAP configuration file corresponds to one source-to-target mapping relationship.

   Restriction: You can enter up to 50 source-to-target server mappings, each with a maximum of 10 source-to-target directory mappings, in the ASNDLSRVMAP file. Type each source-to-target server mapping on a single line. Use spaces to separate each server location and directory path, and press Enter only to create a new line for the next source-to-target server mapping.
4. Modify the ASNDLUSER configuration file to contain login and address information for *each* server that you identified in your environment.
   a. Set the *userid* and *passwd* variables to the correct login information.
   b. Set the *recv-port* and *send-port* variables to the port numbers of the DLFM_ASNCOPYD daemons on the source file server and the target file server, respectively. For example,

      ```
      S1.ABC.COM 10000 -1 USERA XPASS123
      ```

      The file in this example contains one line. This line specifies the login and address information for a source file server, where S1.ABC.COM is the URL location of the server, `10000` is the source file server port number of the DLFM_ASNCOPYD daemon, `USERA` is the user ID, and `XPASS123` is the password of the user ID.

The target file server port number value is -1 (not applicable). You need to specify a valid port number value for sending and storing files only if this same file server is also a target file server in an update-anywhere replication configuration.

5. If necessary, modify the ASNDLPARM configuration file to specify the operational parameters used to control the function of the ASNDLCOPY exit routine.

   Table 9 lists the parameters that you can change in the ASNDLPARM configuration file.

Table 9. Supported ASNDLPARM parameters and associated values

Parameter	Description	Valid values[a]
replace_file	Specifies whether a source file is replicated and assigned a different file name if the file already exists on the target file server. Set this parameter to NO if the columns that you are replicating to the target table are defined as NO LINK CONTROL. You must install DB2 Universal Database Version 8 on the target file server to use the full functionality of this parameter. This parameter always functions as if it were set to NO if the database version that you are using on the target file server is DB2 Universal Database Version 7 or earlier. The default value is YES.	YES, NO
replacement_file_suffix	Specifies the suffix string of the replacement file. If the replicated file already exists on the target file server, the ASNDLCOPY exit routine copies the content of the replicated file to a temporary file with a name that is equal to the source file name with the added suffix. This temporary file is called the replacement file. ASNDLCOPY uses this parameter only if the replace_file parameter is set to YES. The default value is .new.	A character string up to 20 bytes in length.[b]
use_file_on_disk	If set to YES, specifies that the ASNDLCOPY exit routine always attempts to locate a file on disk if the DLFM_ASNCOPYD daemon cannot find the appropriate version of the file on the archive server. The replication process continues to run as the exit routine locates the file. However, you might retrieve inconsistent versions of your files if the daemon cannot find the file on the archive server, even if the corresponding DATALINK column is defined with RECOVERY YES. The default value is NO.	YES, NO

Table 9. Supported ASNDLPARM parameters and associated values (continued)

Parameter	Description	Valid values[a]
min_filecopy	Specifies whether the ASNDLCOPY exit routine creates a subdirectory called .dlinfo to store the timestamp of the source file. This exit routine then determines whether the file has changed by comparing this timestamp value to the last modification time of the source file before copying the source file. Set this option to YES to minimize unnecessary file transfers in the following situations: • The DATALINK file reference is unlinked from a table and then linked to the same table or to another table, but the file content remains the same. • The Apply program invokes a full refresh of the target table. Files that have not changed are not transferred. The default value is YES.	YES, NO
create_tgtdir	Specifies whether the ASNDLCOPY exit routine attempts to re-create all parent directories before storing a target file if the parent directories do not exist on the target file system. The default value is YES.	YES, NO
zero_tgtfile	Specifies whether the ASNDLCOPY exit routine maps a source file to a zero-length URL value and issues a warning message to the log file if the exit routine cannot find a source file because of file availability errors. After issuing a warning message, the exit routine continues. You can fix the URL value at a later time without interrupting current processing. If you set this parameter to NO, the ASNDLCOPY exit routine stops and returns a non-zero return code, causing the Apply program to stop processing the current subscription set. The default value is YES.	YES, NO
preserve_modtime	Specifies whether the ASNDLCOPY exit routine preserves the last modification time of the files that you are replicating. ASNDLCOPY uses this parameter only if this exit routine is connected to the DLFM_ASNCOPYD daemon and if this daemon is configured as the target file-copy daemon. The default value is YES.	YES, NO

Table 9. Supported ASNDLPARM parameters and associated values (continued)

Parameter	Description	Valid values[a]
direct_copy	Specifies whether a source file is copied directly from the source server to the target server rather than replicated through the ASNDLCOPY exit routine. For this parameter to work, you must verify the following setup criteria: • The source file-copy daemon must be able to connect to the target file-copy daemon. Use the PING utility to verify this connectivity. • The target file-copy daemon must support the **PASV** command. The DLFM_ASNCOPYD daemon supports the **PASV** command, but an FTP daemon might not. If you are using FTP, log on to your FTP target file-copy daemon through an FTP client, and type `QUOTE PASV` to verify that this daemon supports the **PASV** command. The default value is `YES`.	YES, NO
umask	Specifies the file-mode creation mask of any file and directory created during replication. Use this parameter only if the target file-copy daemon supports the **SITE UMASK** command. The DLFM_ASNCOPYD daemon supports the **SITE UMASK** command, but an FTP daemon might not. If you are using FTP, log on to your FTP target file-copy daemon through an FTP client, and type `QUOTE SITE UMASK` to verify that this daemon supports the **SITE UMASK** command. The default value is `-1` (no change).	-1, an octal number between 000 and 777
chmod	Specifies the permission mode of any file that is created during replication. Use this parameter only if the target file-copy daemon supports the **SITE CHMOD** command. The DLFM_ASNCOPYD daemon supports the **SITE CHMOD** command, but an FTP daemon might not. If you are using FTP, log on to your FTP target file-copy daemon through an FTP client, and type `QUOTE SITE CHMOD` to verify that this daemon supports the **SITE CHMOD** command. The default value is `-1` (no change).	-1, an octal number between 000 and 777

Notes:

1. The value `YES` and the value `NO` are *not* case sensitive. Other character string values are case sensitive.
2. The string must contain characters that are valid in a file name on the operating system that you are using. Do not use a forward slash (/) on UNIX systems. Do not use a backslash (\) on Windows operating systems.

The following example shows the contents of a sample ASNDLPARM configuration file with four parameter settings:

```
replace_file=YES
replacement_file_suffix=.new
direct_copy=NO
preserve_modtime=YES
```

The ASNDLPARM file is optional. If this file does not exist, the default settings are used.

6. Create the control tables for the Capture program using the Create Capture Control Tables notebook of the Replication Center. See the Replication Center help for details.
7. Enable the database for replication using the Capture Control Servers folder of the Replication Center. Right-click the appropriate database, and select **Enable Database for Replication**. This step sets the archive logging so that the Capture program can read the DB2 log for changes to the registered tables. See the Replication Center help for details.
8. Register a replication source using the Register Tables window of the Replication Center. Select the DATALINK columns and the other table columns that you want to replicate. See the Replication Center help for details.
9. Create the control tables for the Apply program using the **Registered Definitions** folder of the Replication Center. Right-click the **Apply Control Servers** folder and select **Create Apply Control Tables**. See the Replication Center help for details.
10. Create new subscription sets and subscription-set members using the Create Subscription Set notebook of the Replication Center. The subscription set defines the relationship between the replication source and the target table. See the Replication Center help for details.

    Recommendation: Use a separate subscription set for DATALINK columns. However, in some situations, you might need to group multiple tables with different column types into one subscription set. If you do not require this subscription-set setup, try to use a separate subscription set because the Apply program waits for the ASNDLCOPY exit routine to complete its processing before the Apply program completes replication of the subscription set. If the DLFM_ASNCOPYD daemon fails to copy any external file, the entire subscription set fails. If the subscription set fails, the Apply program does not deactivate the subscription set but processes the subscription set again during the next Apply cycle.

Related concepts:

- "An overview of how linked files get replicated" on page 131
- "How the Data Links Manager replication daemon works" on page 137
- "Data replication example with DB2 Data Links Manager" on page 147

Related tasks:

- "Replicating special data types in SQL replication" in the *IBM DB2 Information Integrator SQL Replication Guide and Reference*
- "Configuring servers for SQL replication" in the *IBM DB2 Information Integrator SQL Replication Guide and Reference*
- "Enabling the Data Links Manager Replication daemon" on page 60
- "Operating replication to replicate linked files" on page 146

Related reference:

- "dlfm grant replication read command" on page 216
- "dlfm grant replication write request command" on page 217

Operating replication to replicate linked files

After you set up the DLFM_ASNCOPYD daemon in your replication environment and set up the replication subscription set, you can start the replication process to replicate linked data and other data that you selected for replication.

This topic explains how to start replicating linked files.

Prerequisites:

- Verify that the DLFM_ASNCOPYD replication daemon is enabled.
- Verify that your replication environment is properly set up to use the DLFM_ASNCOPYD replication daemon to replicate linked data.
- Verify that you granted proper Data Links File Manager (DLFM) authorizations to the users who need to replicate linked data.
- Because external files can be very large, ensure that you have sufficient network bandwidth.
- Ensure that you have sufficient disk space on your target system to accommodate the external files. You might need additional disk space depending on your replication requirements and on the size and number of the files that you need to replicate to the target system.

Restrictions:

- DB2 replication does not check update conflicts of external files referenced by DATALINK columns. If you use update-anywhere replication with DATALINK columns, specify **None** for the conflict-detection level to turn off the conflict detection for both the DATALINK columns and the other columns in the same subscription set.
- Target tables that are *base-aggregate* (containing aggregated data from a replication source table) or *change-aggregate* (containing aggregated data based on the contents of a CD table) tables do not support DATALINK columns.

Procedure:

You can now start the Capture and Apply programs. These programs invoke an initial full refresh of the target table and begin the change-capture replication of the linked data and the other data that you selected.

To start the Capture and Apply programs:

1. Start the Capture program using either the Start Capture window of the Replication Center or the **asncap** system command.

   The Capture program reads the DB2 log and retrieves the link type, link reference, and comment of each DATALINK value that it finds. After extracting this DATALINK information from the DB2 log, the Capture program writes this information to the CD table.
2. Start the Apply program using either the Start Apply window of the Replication Center or the **asnapply** system command.

   When a subscription set is ready to be replicated, the Apply program identifies the applicable rows in the CD table. The Apply program then writes the link reference of each DATALINK value that it finds in the CD table to an input file.

The Apply program then invokes the ASNDLCOPY exit routine to read the input file and to map each DATALINK source file location to its corresponding target file location. The ASNDLCOPY exit routine connects to the DLFM_ASNCOPYD (or FTP) daemon and replicates the referenced file from the source file system to the newly mapped target file system location. The Apply program then writes the newly mapped target DATALINK values and the other changed database data to the target database, which links to the replicated files.

Related concepts:

- "An overview of how linked files get replicated" on page 131
- "How the Data Links Manager replication daemon works" on page 137
- "Data replication example with DB2 Data Links Manager" on page 147

Related tasks:

- "Replicating special data types in SQL replication" in the *IBM DB2 Information Integrator SQL Replication Guide and Reference*
- "Operating the Apply program" in the *IBM DB2 Information Integrator SQL Replication Guide and Reference*
- "Operating the Capture program" in the *IBM DB2 Information Integrator SQL Replication Guide and Reference*
- "Registering tables and views as SQL replication sources" in the *IBM DB2 Information Integrator SQL Replication Guide and Reference*
- "Subscribing to sources for SQL replication" in the *IBM DB2 Information Integrator SQL Replication Guide and Reference*
- "Enabling the Data Links Manager Replication daemon" on page 60
- "Setting up replication of linked files" on page 140

Related reference:

- "asnapply: Starting Apply" in the *IBM DB2 Information Integrator SQL Replication Guide and Reference*
- "asncap: Starting Capture" in the *IBM DB2 Information Integrator SQL Replication Guide and Reference*

Data replication example with DB2 Data Links Manager

This topic presents a simple example of how to use replication with DB2® Data Links Manager to replicate DATALINK values and other table data. The replication configuration in this example is a data-distribution configuration in which changes from a single source database are replicated to one read-only target table.

The steps in this example show you how to initiate the following actions:

- Replicate data in a source table called EMPLOYEE to a newly created target table called TGTEMPCOPY
- Change a DATALINK column value and its associated linked file
- Replicate the change from the external source file system to the external target file system

Table 10 on page 148 lists the structure of the EMPLOYEE table on the source database.

Table 10. The EMPLOYEE table on the SRCDB database

Column	Column type
EMPNO	VARCHAR(3)
FIRSTNAME	VARCHAR(20)
LASTNAME	VARCHAR(20)
EMP_PICTURE	DATALINK

Table 11 lists the structure of TGTEMPCOPY table on the target database.

Table 11. The TGTEMPCOPY table on the TGTDB database

Column	Column type
EMPNO	VARCHAR(3)
FIRSTNAME	VARCHAR(20)
LASTNAME	VARCHAR(20)
EMP_PICTURE	DATALINK

The EMP_PICTURE column is defined with a DATALINK data type and is specified as RECOVERY YES.

The columns and column definitions are the same on both the EMPLOYEE source table and the TGTEMPCOPY target table. The DATALINK column values point to files, which contain photographs of the employees. These files reside on an external file system.

The external source file system in this example has the following configuration:

Host name
S1.PRD.COM

Path name
/data/dl_file_system/sd1

File names
*.gif

For example:

```
joe.gif
jane.gif
kathy.gif
```

The external target file system has a similar configuration:

Host name
T1.PRD.COM

Path name
/data/dl_file_system/td1

File names
*.gif

DATALINK replication steps

You must set up and enable the DLFM_ASNCOPYD daemon and then set up your replication environment to start replicating the linked data.

The following steps show you how to:

- Set up and enable the DLFM_ASNCOPYD daemon on the source and target file servers.
- Modify the ASNDLCOPY configuration files for Data Links replication.
- Set up the Capture and Apply programs.
- Replicate a changed DATALINK column value.

Step 1: Setting up the DLFM_ASNCOPYD daemon on the source file server

Set up and enable the DLFM_ASNCOPYD daemon on your source file server. Specify the login user ID and its authorization.

1. Log on to the Data Links source server 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 `YES`.
3. Use the **db2set** command to set the registry variable DLFM_ASNCOPYD_PORT to `10000` (the source replication daemon port number in this example). Make sure that this port number is reserved in the appropriate directory.

   On AIX® systems and Solaris™ Operating Environments, this directory is /etc/services.

   On Windows® NT and Windows 2000 systems, this directory is \system32\drivers\etc\services.
4. Restart the Data Links File Manager (DLFM) to initiate the Data Links Manager replication daemon.
5. Verify that the DLFM is running.
6. Choose a login user ID to retrieve files (`USERA` in this example).
7. Identify the directory where the source files are located (/data/dl_file_system/sd1 in this example).
8. Issue the following command:

   ```
   dlfm grant replication read privilege on dir /data/dl_file_system/sdl to system user USERA
   ```

   This command grants `USERA` the authority to read linked files in the specified directory.

Step 2: Setting up the DLFM_ASNCOPYD daemon on the target file server

Set up and enable the DLFM_ASNCOPYD daemon on your target file server. Specify the login user ID and its appropriate authorizations.

1. Log on to the Data Links target server 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 `YES`.
3. Use the **db2set** command to set the registry variable DLFM_ASNCOPYD_PORT to `65535` (the target replication daemon port number in this example). Make sure that this port number is reserved in the appropriate directory.

   On AIX systems and Solaris Operating Environments, this directory is /etc/services.

   On Windows NT® and Windows 2000 systems, this directory is \system32\drivers\etc\services.
4. Restart the DLFM to initiate the Data Links Manager replication daemon.
5. Verify that the DLFM is running.

6. Choose a login user ID to send and store files (USERB in this example).
7. Identify the directory where the target files are located (/data/dl_file_system/td1 in this example).
8. Issue the following command:

   ```
   dlfm grant replication read privilege on dir /data/dl_file_system/tdl to system user USERB
   ```

   This command grants USERB the read privilege needed to check the status of the files in the target file server.
9. Issue the following command:

   ```
   dlfm grant replication write request privilege to system user USERB
   ```

   This command grants USERB the authority to issue file write requests to the DLFM_ASNCOPYD daemon. USERB must have file system authority to create and to write to the /data/dl_file_system/td1 target directory.

Step 3: Modifying the ASNDLCOPY configuration files

Modify the contents of the ASNDLCOPY configuration files to specify the proper mapping, address, and operational information.

1. Modify the ASNDLSRVMAP configuration file to specify the correct source-URL-to-target-URL mapping for the example:

   ```
   HTTP://S1.PRD.COM  HTTP://T1.PRD.COM /data/dl_file_system/sd1  /data/dl_file_system/td1
   ```

   This example correctly maps the source-to-target host names and path names and sets the schemes (such as HTTP, FILE, and UNC) to match the schemes that are specified in the DATALINK values.
2. Modify the ASNDLUSER configuration file so that your authorized login ID can connect to the source and target file servers:

   ```
   S1.PRD.COM 10000 -1 USERA XPASS123
   T1.PRD.COM -1 65535 USERB XPASS999
   ```

   where:

   - S1.PRD.COM is the address of the source file server
   - 10000 is the DLFM_ASNCOPYD source replication daemon port number
   - USERA is the login user ID for the source file server
   - XPASS123 is the password for this user ID on the source file server
   - T1.PRD.COM is the target file server address
   - 65535 is the DLFM_ASNCOPYD target replication daemon port number
   - USERB is the login user ID for the target file server
   - XPASS999 is the password for this user ID on the target file server

   The *send-port* value in the first line of this ASNDLUSER configuration file is set to -1, indicating that this server is a source file server. The *recv-port* value in the second line of this configuration file is also set to -1, indicating that this server is a target file server. (In an update-anywhere configuration, you would specify actual port numbers for these two port number values to indicate that each server is configured as both a source file server and a target file server.)
3. Modify the ASNDLPARM configuration file to include the following lines:

   ```
   replace_file=YES
   replacement_file_suffix=.rpl
   ```

   These two lines ensure that a source file is replicated and assigned a different name, with a suffix of .rpl, if a file with the same name as the target file already exists on the target file server.

Step 4: Setting up the Capture and Apply programs for DATALINK replication

After modifying the ASNDLCOPY configuration files and planning the basic model for your DATALINK replication, you are ready to set up the replication Capture and Apply programs. You need to use the Replication Center for the following steps. Verify that the Replication Center is running. For example on the Windows operating systems, from the Windows **Start** menu: select **Programs —> IBM® DB2 —> General Administration Tools —> Replication Center**.

1. Create the control tables for the Capture program. Use the Create Capture Control Tables notebook.
2. Select the SRCDB database in the Select a Server window to specify the Capture control server.
3. From the Capture Control Servers folder, right-click the SRCDB database and select **Enable Database for Replication**. Click **OK** on the Enable Database for Replication window to use archive logging for the SRCDB database and to initiate a backup for the database.

   This step sets the archive logging so that the Capture program reads the DB2 log for log records that include changes to the registered tables. The log must be an archive log so that the log file is not reused by DB2 before the Capture program can read the log.
4. Use the Register Tables window to register the EMPLOYEE table as a replication source.
5. Create the control tables for the Apply program. Right-click the Apply Control Servers folder and select **Create Apply Control Tables** from the Registered Definitions folder.

   The Apply program reads its control tables for current subscription-set information and stores the replication status information in these tables.
6. Use the Create Subscription Set notebook to create a subscription set and a subscription-set member.

   The subscription set defines a relationship between the replication source database (SRCDB in this example) and a target database (TGTDB in this example). The subscription-set member defines a relationship between the replication source table (EMPLOYEE in this example) and one or more target tables (this example has only one table, which is called TGTEMPCOPY).

   Assume that you want to replicate all columns from the source table to the target table.

Step 5: Replicating the data

After you set up the control tables, register the source, and create your subscription set, you can start the Capture and Apply programs to invoke the initial full refresh of your target table (TGTEMPCOPY) and to begin change-capture replication of the DATALINK and other table data.

1. Start the Capture program. Use the Start Capture window and select the SRCDB database.

   The Capture program is now running but does not begin capturing changes for the registered table until the Apply program completes a full refresh of the table.
2. Start the Apply program. Use the Start Apply window.

   You should see a message in the DB2 Message window that the command ran successfully, indicating that the Apply program is now running.

   After one replication cycle, the EMPLOYEE source table and the TGTEMPCOPY target table contain the same rows. However, the link references

of the DATALINK column (EMP_PICTURE) in the target table point to the directory path of the external file system on the *target* system.

Table 12 lists the contents of the EMPLOYEE table on the SRCDB database.

Table 12. The EMPLOYEE table contents

EMPNO	FIRSTNAME	LASTNAME	EMP_PICTURE
100	Joe	Smith	HTTP://S1.PRD.COM/data/dl_file_system/sd1/joe.gif
200	Jane	Jones	HTTP://S1.PRD.COM/data/dl_file_system/sd1/jane.gif
300	Kathy	Chen	HTTP://S1.PRD.COM/data/dl_file_system/sd1/kathy.gif

Table 13 lists the contents of the TGTEMPCOPY target table on the TGTDB database after the initial full refresh.

Table 13. The TGTEMPCOPY table contents

EMPNO	FIRSTNAME	LASTNAME	EMP_PICTURE
100	Joe	Smith	HTTP://**T1.PRD.COM/data/dl_file_system/td1**/joe.gif
200	Jane	Jones	HTTP://**T1.PRD.COM/data/dl_file_system/td1**/jane.gif
300	Kathy	Chen	HTTP://**T1.PRD.COM/data/dl_file_system/td1**/kathy.gif

3. View the contents of the TGTEMPCOPY target table. Use the DB2 Command Center, or issue the following SQL statement from a DB2 command window:

   ```
   SELECT * FROM schema.TGTEMPCOPY;
   ```

   where *schema* is the name of the schema of the target table.

4. Change the picture for Joe Smith.

   a. Enter the following SQL statement to unlink the Data Links file in the SRCDB source database.

   ```
   EXEC SQL UPDATE EMPLOYEE
      SET EMP_PICTURE = NULL
    WHERE EMPNO = 100;

   EXEC SQL COMMIT;
   ```

   b. Copy the new picture of Joe Smith, replacing the old picture. Update the appropriate row of the SRCDB source table by entering the following SQL statement:

   ```
   EXEC SQL UPDATE EMPLOYEE
      SET EMP_PICTURE = DLVALUE('HTTP://S1.PRD.COM/data/dl_file_system/sd1/joe.gif')
    WHERE EMPNO = 100;

   EXEC SQL COMMIT;
   ```

   The next time that the Capture program captures changes, it will write new rows to the change-data (CD) table. These new CD rows correspond to your SQL UPDATE statements and contain the new URL value for the DATALINK column in the EMPLOYEE table. The CD table rows also contain the version number of the new linked file to be replicated because the DATALINK column value is defined as RECOVERY YES.

   When the subscription set is eligible for replication, the Apply program reads these new rows from the CD table and stores the URL value in the input file:

   ```
   51 HTTP://S1.SYS.COM/data/dl_file_system/sd1/joe.gif Y 00000BFB4F503D4442D9
   ```

   where *00000BFB4F503D4442D9* is the version number of the file.

The ASNDLCOPY exit routine then maps the external file based on the source-URL-to-target-URL mapping that you specified in the ASNDLSRVMAP configuration file.

Assume that this external file already exists on the target file system. The ASNDLCOPY exit routine then copies this file to a replacement file with a suffix of `.rpl`, which you specified in the ASNDLPARM configuration file. The ASNDLCOPY exit routine stores this temporary target file location in the result file:

```
51 HTTP://S1.SYS.COM/data/dl_file_system/sd1/joe.gif 4 .rpl
```

After the ASNDLCOPY exit routine finishes processing, the Apply program reads the result file and replicates the changes to the TGTEMPCOPY target table. DB2 then renames the temporary file, `joe.gif.rpl`, to the file name in the target URL, `joe.gif`, as the replication transaction is committed.

5. View the contents of both tables.

   Table 14 lists the contents of the EMPLOYEE table, and Table 15 lists the contents of the TGTEMPCOPY table after the change is replicated.

Table 14. The EMPLOYEE source table after replication

EMPNO	FIRSTNAME	LASTNAME	EMP_PICTURE
100	Joe	Smith	HTTP://S1.PRD.COM/data/dl_file_system/sd1/joe.gif
200	Jane	Jones	HTTP://S1.PRD.COM/data/dl_file_system/sd1/jane.gif
300	Kathy	Chen	HTTP://S1.PRD.COM/data/dl_file_system/sd1/kathy.gif

Table 15. The TGTEMPCOPY target table after replication

EMPNO	FIRSTNAME	LASTNAME	EMP_PICTURE
100	Joe	Smith	HTTP://T1.PRD.COM/data/dl_file_system/td1/**joe.gif**
200	Jane	Jones	HTTP://T1.PRD.COM/data/dl_file_system/td1/jane.gif
300	Kathy	Chen	HTTP://T1.PRD.COM/data/dl_file_system/td1/kathy.gif

The DATALINK value, HTTP://T1.PRD.COM/data/dl_file_system/td1/joe.gif, of the changed row in the TGTEMPCOPY target table now points to the new picture, which was copied to the target file system.

Related concepts:

- "An overview of how linked files get replicated" on page 131
- "How the Data Links Manager replication daemon works" on page 137

Related tasks:

- "Replicating special data types in SQL replication" in the *IBM DB2 Information Integrator SQL Replication Guide and Reference*
- "Operating the Apply program" in the *IBM DB2 Information Integrator SQL Replication Guide and Reference*
- "Operating the Capture program" in the *IBM DB2 Information Integrator SQL Replication Guide and Reference*
- "Registering tables and views as SQL replication sources" in the *IBM DB2 Information Integrator SQL Replication Guide and Reference*
- "Configuring servers for SQL replication" in the *IBM DB2 Information Integrator SQL Replication Guide and Reference*

- "Subscribing to sources for SQL replication" in the *IBM DB2 Information Integrator SQL Replication Guide and Reference*
- "Setting up replication of linked files" on page 140
- "Operating replication to replicate linked files" on page 146

Chapter 9. Data Links File Manager server availability issues

This chapter describes the issues that you need to consider when your Data Links File Manager (DLFM) is unavailable, when you recover a DB2 database that contains DATALINK column values, and when you use certain utilities or set up your environment for file server availability.

This chapter also describes high availability considerations for both DB2 utilities and for your Data Links server when you use an IBM High Availability Cluster Multiprocessor (HACMP) environment on an AIX system.

DB2 utilities and the Data Links File Manager

This topic describes the issues that you need to consider when you use certain DB2® utilities and the Data Links File Manager (DLFM) is unavailable.

You can use DB2 utilities for tables with DATALINK values and even run these DB2 utilities when your DLFM is down. When the DLFM is down, it is unable to communicate with DB2, and you cannot link or unlink files. You can, however, continue to run certain DB2 utilities, if necessary. But you need to be aware of how these utilities work with the DB2 Data Links Manager.

This topic describes the issues to consider when you use the following utilities:

- "BACKUP utility"
- "RESTORE and ROLLFORWARD utilities"
- "RECONCILE utility" on page 159

BACKUP utility

DB2 ensures that linked files at Data Links servers are backed up whenever a database or table space is backed up. (The BACKUP utility can run either online or offline, and the backup image can be of either a database or a table space.)

The Data Links server schedules linked files (specifically, files linked by DATALINK columns that are defined with RECOVERY YES) for asynchronous copying to an archive server or to disk. When you start the BACKUP utility, DB2 connects to all the Data Links servers that are specified in the Data Links configuration file. Then DB2 ensures that all scheduled linked files have been copied. If DB2 is unable to connect to one or more Data Links servers that are configured for the database, the BACKUP utility still continues to run and completes with a warning message.

After the previously unavailable Data Links server restarts, the BACKUP utility finishes processing asynchronously, and the Data Links server then becomes available to the database. All processing of outstanding backups must complete successfully before a Data Links server can be made available to the database.

RESTORE and ROLLFORWARD utilities

When you use the RESTORE and ROLLFORWARD utilities on tables with DATALINK columns (defined with RECOVERY YES), these utilities operate under the following conditions:

- If any Data Links server that is recorded in the backup image is not running, the RESTORE or ROLLFORWARD utility still completes successfully but generates a warning message.

  If a table with DATALINK columns references an unavailable Data Links server, the table is placed in a Datalink_Reconcile_Pending (DRP) state. The table state changes to DRP after the restore operation (or the rollforward operation, if used) completes. This restore processing must complete successfully before the Data Links servers become available again to the database. After the Data Links servers become available, the asynchronous process that completes the backup processing also completes the restore processing.
- If any Data Links server that is recorded in the backup image stops running during a restore operation, the restore operation fails. An asynchronous process will complete the restore processing after the Data Links server is restarted.
- If a previous database restore operation is incomplete on any Data Links server, a subsequent database or table space restore operation fails unless that Data Links server is restarted and the previous restore operation completes.
- All DATALINK column information that was recorded in the backup image must exist in the corresponding Data Links File Manager databases (DLFM_DBs) on the Data Links servers.

  If the DATALINK column information is missing from the DLFM_DB, DB2 places the appropriate database table in a Datalink_Reconcile_Not_Possible (DRNP) state. DB2 places the table in a DRNP state after the restore operation (or the rollforward operation, if used) completes.

  A table remains available to the users, even if all the DATALINK values do not exist in both the backup image and the DLFM_DB. However, the values in the DATALINK columns might not reference the linked files accurately. To prevent inaccurate references, issue the following command to place the table in a Check_Pending state:

  ```
  SET CONSTRAINTS for table-name TO DATALINK RECONCILE PENDING
  ```

  where *table-name* is the name of the table that you want to place into Check_Pending state.

DB2 records DRNP and DRP table states in the db2diag.log file when the RESTORE and ROLLFORWARD utilities run. You can also use the **db2dart** command to obtain this information.

Recommendation: You should archive the datalink.cfg file regularly to preserve the latest restore information. The datalink.cfg file in the database backup image reflects the status at backup time only. However, you need the latest available datalink.cfg file to satisfy all conditions that can occur during the restoration of tables with DATALINK column values. You should back up the datalink.cfg file after every **ADD DATALINKS MANAGER** or **DROP DATALINKS MANAGER** command invocation.

If the latest datalink.cfg file is not available on disk, replace the existing datalink.cfg file (restored from a backup image) with the latest datalink.cfg file that was archived before running a rollforward operation. Replace this file after the database is restored.

On AIX® systems and Solaris™ Operating Environments, the datalink.cfg file is located in the *INSTHOME/INSTDIR/NODExxxx/SQLyyyyy* directory,

where:

- *INSTHOME* is the home directory of the DB2 instance

- *INSTDIR* is the instance directory of the DB2 instance
- *NODExxxx* is the database directory
- *SQLyyyyy* is the data directory for the database

For example:

/home/user1x/user1x/NODE0000/SQL00001

Typically, you would have only one DB2 instance (user1x in the example) and that name is the same as the DB2 instance user name.

On Windows® NT and Windows 2000 systems, the datalink.cfg file is located in the *x:\INSTDIR\NODExxxx\SQLyyyyy* directory,

where:

- *x:* is the drive where DB2 is installed
- *INSTDIR* is the instance directory of the DB2 instance
- *NODExxxx* is the database directory
- *SQLyyyyy* is the data directory for the database

For example:

c:\DB2\NODE0000\SQL00001

Recommendation: If your database tables contain DATALINK columns that are defined as READ PERMISSION DB and WRITE PERMISSION ADMIN, use the **dlfm list upd_in_progress files for db** command *before* invoking the RESTORE utility. The **dlfm list upd_in_progress files for db** command lists the linked files (specifically, files linked to DATALINK columns that are defined with the WRITE PERMISSION ADMIN attribute) that are currently in an update-in-progress state. These files might have been recently changed but might not yet be synchronized with DB2. The RESTORE utility restores each file according to the file version number as specified by DB2 and renames the file to *file-name*.MOD, where *file-name* is the name of the restored linked file. By using the **dlfm list upd_in_progress files for db** command before invoking the RESTORE utility, you can see which linked files will be renamed when the utility is invoked; or you can notify your users to complete their updates to the files.

The following sections describe three different ways in which these utilities can work to restore databases and table spaces with DATALINK values.

Restoring databases from an offline backup without rolling forward

You can restore without rolling forward only at the database level, not at the table space level. To restore a database without rolling forward, you can restore an unrecoverable database (that is, a database that uses circular logging). Alternatively, you can specify the WITHOUT ROLLING FORWARD parameter on the **RESTORE DATABASE** command.

If you use the RESTORE utility and specify the WITHOUT DATALINK option, the utility places each table with DATALINK column values in a DRP state and does not initiate a reconciliation with the Data Links servers during the restore operation.

If tables contain DATALINK column values that reference data from a dropped Data Links server, the RESTORE utility places these tables in DRP states if these two conditions are met:

- You do not specify the WITHOUT DATALINK option.
- A Data Links server recorded in the backup image is no longer defined to the database (that is, the **DROP DATALINKS MANAGER** command was issued).

If you do not specify the WITHOUT DATALINK option and all the Data Links servers referencing DATALINK values are available, the following actions occur for each Data Links server recorded in the backup image:

- All linked files are marked as unlinked if these files were linked after the backup image for the database restore operation was created. These files must be marked as unlinked because the backup image did not record them as linked.
- All files that were linked before the backup image was created, but unlinked after the backup image was created, are marked as linked. These files must be marked as linked because they are recorded as linked in the backup image.

The preceding actions occur only if all the Data Links servers recorded in the backup image are running.

Restoring databases and table spaces, and rolling forward to the end of the logs

If you run the RESTORE utility and then initiate a rollforward operation of either a database or a table space to the end of the logs (meaning that all logs are provided), you do not need to initiate a reconciliation check. However, you *should* initiate a reconciliation check if at least one of the Data Links servers recorded in the backup image is not running during the restore operation. If you are not sure whether all the logs were available for the rollforward operation, or if you think that you might need to reconcile DATALINK column values, issue the following SQL statement for each applicable table:

```
SET INTEGRITY FOR table-name TO DATALINK RECONCILE PENDING
```

where *table-name* is the name of the table with DATALINK columns. This SQL statement places the table in DRP and Check_Pending states.

If you do not want the table to be in a Check_Pending state, issue the following SQL statement:

```
SET INTEGRITY FOR table-name DATALINK RECONCILE PENDING IMMEDIATE UNCHECKED
```

where *table-name* is the name of the table. This SQL statement takes the table out of the Check_Pending state but leaves the table in a DRP state. You must use the RECONCILE utility to take the table out of the DRP state.

A backup image can contain DATALINK data that references a DB2 Data Links Manager that is registered when the backup is made but is subsequently dropped from the database. When the corresponding table spaces are rolled forward, the ROLLFORWARD utility places each table that contains the DATALINK data in a DRP state.

Restoring databases and table spaces, and rolling forward to a point in time

You can use the RESTORE and ROLLFORWARD utilities for point-in-time recovery of tables with DATALINK column values.

When a table space is rolled forward to a point in time, any tables that are assigned to that table space and that contain DATALINK values are placed in DRP states. You can then use the RECONCILE utility to remove each table from a DRP state.

RECONCILE utility

You can use the RECONCILE utility to reconcile DATALINK values. This utility validates that the referenced files either exist on the Data Links server or that the links can be re-established.

Important: Before you invoke the RECONCILE utility, you should issue an SQL COMMIT or ROLLBACK statement to complete all transactions and to release all locks.

If a Data Links server file reference does not exist or cannot be re-established, the RECONCILE utility places a copy of the rows with errors into an exception table (if specified). The utility also places the reason for each error into the exception table. If a DB2 Data Links Manager that is referenced by a database table becomes unavailable during RECONCILE exception handling, the RECONCILE utility places the table in a DataLink_Reconcile_DLM_Pending (DRDP) state. The utility completes but issues a warning message.

If the Data Links File Manager is unavailable, you can still run the RECONCILE utility. The utility issues a warning indicating that the Data Links File Manager is unavailable, and the utility then continues processing. You can run the RECONCILE utility again when the Data Links File Manager is available.

You might need to run the RECONCILE utility under the following situations:

- The entire database is restored and rolled forward to a point in time. Because the entire database is rolled forward to a committed transaction, no tables will be in Check_Pending state (due to referential constraints or check constraints). All data in the database is brought to a consistent state. The DATALINK values, however, might not be synchronized with the metadata in the DB2 Data Links Manager, and reconciliation is required.
- A particular Data Links server running the DB2 Data Links Manager loses track of its metadata (for example, when the Data Links server is restarted or when the Data Links server metadata is restored to a back-level state).

  If the metadata of DATALINK columns is not defined on the DB2 Data Links Manager, the RECONCILE utility places the table in a DRNP state. The utility then completes with a warning message.
- A file system is not available (for example, because of a disk failure), and the file system contents are not synchronized with the DATALINK column references on the DB2 server. In this situation, files might be missing.
- A DB2 Data Links Manager is dropped from a database, and DATALINK FILE LINK CONTROL values referencing that Data Links Manager still exist.

Recommendation: If your database tables contain DATALINK columns that are defined as READ PERMISSION DB and WRITE PERMISSION ADMIN, use the **dlfm list upd_in_progress files for db** command *before* invoking the RECONCILE utility. The **dlfm list upd_in_progress files for db** command lists the linked files (specifically, files linked to DATALINK columns that are defined with the WRITE PERMISSION ADMIN attribute) that are currently in an update-in-progress state. These files might have been recently changed but might not yet be synchronized with DB2. When you invoke the RECONCILE utility, each file is restored according

to the file version number as specified by DB2 and is then renamed to *file-name*.MOD, where *file-name* is the name of the restored linked file. By using the **dlfm list upd_in_progress files for db** command before invoking the RECONCILE utility, you can see which linked files will be renamed when the utility is invoked; or you can notify your users to complete their updates to the files.

Related concepts:

- "Data Links server file backups" on page 33

Related reference:

- "BACKUP DATABASE Command" in the *Command Reference*
- "RESTORE DATABASE Command" in the *Command Reference*
- "ROLLFORWARD DATABASE Command" in the *Command Reference*
- "RECONCILE Command" in the *Command Reference*
- "DB2 Data Links Manager system setup and backup recommendations" on page 176
- "File system backup and restore recommendations" on page 178

Database recovery using DB2 utilities with the Data Links File Manager

This topic describes recovery processes and how DB2® utilities can restore and reconcile DATALINK column values with other DB2 relational data.

You can use the RESTORE, ROLLFORWARD, and RECONCILE utilities to recover your DB2 database, to apply outstanding transactions from the database log files, and to validate references to files for the DATALINK columns values. The options that you specify with the RESTORE utility and how you use the ROLLFORWARD and RECONCILE utilities depend on the type of database recovery that you need.

The following sections describe three possible recovery scenarios, the utilities that you need to invoke, and how the DB2 Data Links Manager works with these utilities during database recovery:

- "Scenario 1: Restoring the database without rolling forward"
- "Scenario 2: Restoring the database and rolling forward to the end of the logs" on page 161
- "Scenario 3: Restoring the database and rolling forward to a point in time" on page 161

Scenario 1: Restoring the database without rolling forward

Use the RESTORE utility, and specify the WITHOUT ROLLING FORWARD parameter to restore the DB2 database and automatically invoke a fast reconcile process. A fast reconcile process invokes the Data Links File Manager (DLFM) to synchronize the DLFM_DB with the database tables by using the timestamp of the restored backup image. If a file becomes linked after the backup process finished, the fast reconcile process unlinks the file. If a file was linked before the backup process started but becomes unlinked after the backup finished, the fast reconcile process relinks the file. The fast reconcile process restores changed or missing files from the archive area, if necessary.

If a fast reconcile process is unsuccessful (for example, because a backup record was not found), each affected table is placed in a Datalink_Reconcile_Pending (DRP) state. If a table is in a DRP state, you cannot issue an SQL UPDATE,

INSERT, or DELETE statement to change a DATALINK column value. Run the RECONCILE utility after you restore the database.

Scenario 2: Restoring the database and rolling forward to the end of the logs

Use the RESTORE utility, but do *not* specify the WITHOUT ROLLING FORWARD parameter. This utility restores the DB2 database only and does not initiate a fast reconcile process.

You need to run the ROLLFORWARD utility after you restore the DB2 database. The ROLLFORWARD utility brings the DB2 database to the current state, but does not affect the DLFM.

After you initiate a rollforward operation of the database to the end of the logs (meaning that all logs are provided), you probably do not need to run the RECONCILE utility. The DLFM_DB is probably synchronized with DB2. Therefore, each table with DATALINK column values should not be in a DRP state. However, if any table is in a DRP state, run the RECONCILE utility to reconcile the DATALINK column values.

Scenario 3: Restoring the database and rolling forward to a point in time

This scenario is similar to scenario 2, and you need to run these same utilities:

- The RESTORE utility but do *not* specify the WITHOUT ROLLING FORWARD parameter
- The ROLLFORWARD utility after you restore the DB2 database

After you initiate a rollforward operation of the database to a point in time, the entire database is rolled forward to a committed transaction (*not* to the end of the logs). Although all data in the database is brought to a consistent state, the DATALINK values might not be synchronized with the metadata in the DB2 Data Links Manager. You will need to run the RECONCILE utility after the rollforward operation completes to validate the references to files for the DATALINK values in the database tables that were placed in DRP states.

Related concepts:

- "DB2 utilities and the Data Links File Manager" on page 155

Related reference:

- "RESTORE DATABASE Command" in the *Command Reference*
- "ROLLFORWARD DATABASE Command" in the *Command Reference*
- "RECONCILE Command" in the *Command Reference*

Setting up high availability support with the Data Links server (AIX)

This topic explains how you can set up a High Availability Cluster Multiprocessor (HACMP) environment to support DB2 Data Links Manager on AIX systems and describes two HACMP configurations.

The HACMP environment is based on cluster multi-processing. Cluster multi-processing refers to a group of networked machines that share disk resources. In a cluster, multiple server processors interact to provide services and

resources to other clients and server components. The HACMP environment defines the processor relationships in which standby nodes on a cluster provide the services offered by the active node when that active node is disabled.

The relationship between the nodes is the basis for a failover. A failover of services occurs when the HACMP environment experiences a change that stops services on the active node and resumes those services on a standby node.

The two basic HACMP cluster configurations that you can use with your DB2 Data Links Manager environment are:

Hot standby
A configuration in which the DB2 host and the Data Links server reside on two different clusters. Each cluster is configured with two nodes: one active node that runs the DB2 host or the Data Links server during normal operations and one standby node that takes over functionality of the active node during a failover.

Mutual takeover
A configuration in which the DB2 host and the Data Links server reside on the same HACMP cluster. The DB2 host and the Data Links server back up each other, taking over each other's functionality during a failover.

The following sections describe sample hot standby and mutual takeover configurations setups. These samples include instructions on how to set up your DB2 host and your Data Links server for high availability using the hot standby and mutual takeover configurations.

Each sample describes only one possible high availability solution for your Data Links environment. You can choose to add more high availability configuration setups depending on the requirements of your site. For example, the following hot standby configuration sample uses both an active node and a standby node, each with a single network adapter. However, you could use two network adapters on each node: one active adapter and one standby adapter. If a failure occurs on the active network adapter, the application does not switch to the standby node, but the HACMP uses the standby adapter locally to make the application available.

For more information about high availability setups on AIX systems, see the *HACMP for AIX Concepts and Facilities*, the *HACMP for AIX Installation Guide*, the *HACMP for AIX Planning Guide*, and the Related links at the end of this topic.

Hot standby configuration

In this configuration, the DB2 host and the Data Links server typically reside on two different HACMP clusters. Each cluster contains one active node, which runs the DB2 host or Data Links server during normal operations, and one standby node, which takes over the functionality of the active node during a failure. The standby node in each cluster is dedicated to failover operations of the active node and typically does not run any other applications. Alternatively, you can configure the DB2 host and the Data Links server to the same HACMP cluster. Then use a third node within the same cluster as a standby node for each DB2 host and Data Links server.

Figure 7 on page 163 illustrates a typical hot standby configuration for a DB2 host or a Data Links server cluster.

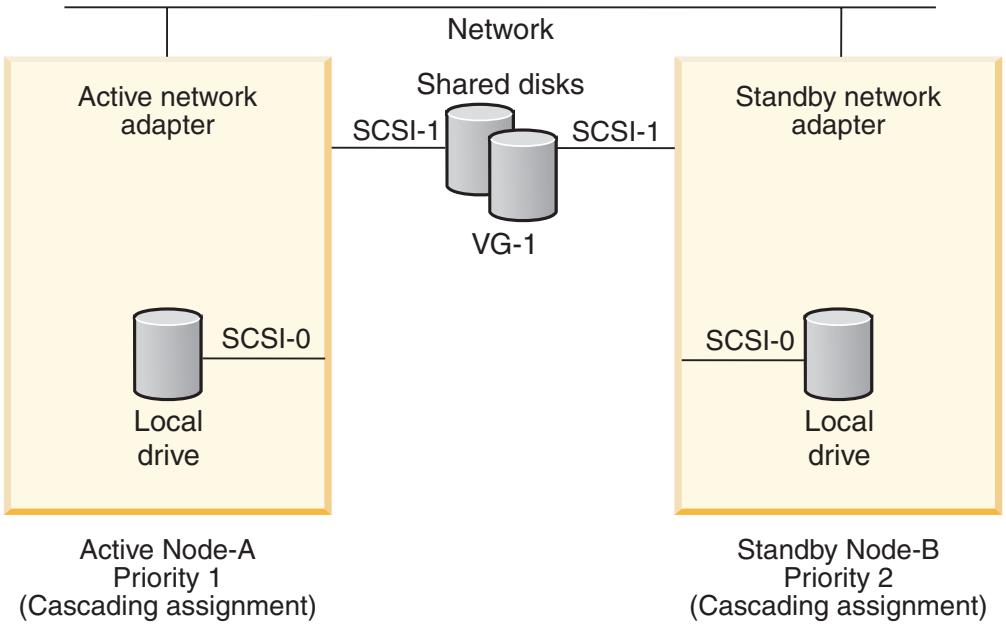


Figure 7. Hot standby configuration

The cluster contains two nodes: an active node and a standby node. Each node uses its own local disk on the SCSI-0 adapter.

VG-1 is the shared volume group of disks and file systems. The disks in VG-1 are connected on separate SCSI adapters. The active node has priority over the standby node for the VG-1 volume. The active node has primary control over the VG-1 volume and acts as the DB2 host. If the active node fails or is down for a scheduled outage, VG-1 uses the standby node. The HACMP Cluster Manager detects the failure and then starts the DB2 host or the Data Links server applications on the standby node.

Assume that the host name of the active node is Node-A and its service IP address is IP-A. Also, assume that the host name of the standby node is Node-B with a service IP address called IP-B. In case of a failover (that is, Node-A fails), the HACMP Cluster Manager causes Node-B to release its IP address (IP-B) and to configure that address with the IP address of Node-A.

Setting up the DB2 host for a hot standby configuration

You must properly install and configure the DB2 host for a hot standby configuration.

Prerequisite:

Before you install and configure the DB2 host, you should understand the general concepts and administration requirements of HACMP on AIX operating systems.

Procedure:

Assume that the DB2 host resides in a cluster that uses a shared volume group called VG-1.

1. Verify that the DB2 host contains the following file systems under the VG-1 volume group.
   - A file system that contains the DB2 instance home directory

- A file system that contains the DB2 host database directory

These file systems must be accessible to both the active and standby nodes under the same absolute path name.

2. Make sure that both the active node and the standby node of volume group VG-1 have the same name and the same major device number.
3. Install the DB2 host software with the same installation options and parameters on both nodes.

   When installing the DB2 host software on the active node, make sure that VG-1 is attached to this node. Assume that /home and /dbfs are the file systems created on the VG-1 and that the DB2 instance home directory (/home/DB2) and the database directory (/dbfs/DB2) are created during the installation process.

   Before installing the software on the standby node, create two temporary file systems on the local disk of the standby node. Mount these file systems as /home and /dbfs.

   After installing the software on the standby node, unmount the /home and /dbfs file systems, and then delete these temporary file systems from the local disk of the standby node.

   After a failover operation, the standby node mounts the /home and /dbfs file systems from the VG-1 volume group. The DB2 instance on the standby node uses the same instance home and database directories that were created on these file systems during the installation of the DB2 software on the active node.

   Important: You can have multiple instances and database directories on the active node. However, you must set up all the home and database directories for the instance using the preceding steps so that the standby node can share these directories after a failover.

You can modify the sample script called rc.db2server.dls in the ~/sqllib/samples/hacmp directory.

Important: Before you run the rc.db2server.dls script, run one of the following two scripts depending on the UNIX shell that you are using:

```
INSTHOME/sqllib/db2profile (for the bash, Bourne, or Korn shell)
INSTHOME/sqllib/db2cshrc (for the C shell)
```

where *INSTHOME* is the home directory of the DB2 instance. By running the *INSTHOME*/sqllib/db2profile or the *INSTHOME*/sqllib/db2cshrc script, you ensure that the rc.db2server.dls script will execute properly.

Use the rc.db2server.dls script to enable the HACMP Cluster Manager to start and stop the DB2 host.

Sample rc.db2server.dls script:

```
!/bin/ksh
#
# Licensed Materials - Property of IBM
#
# (C) COPYRIGHT International Business Machines Corp. 1990,2002
# All Rights Reserved
#
# US Government Users Restricted Rights - Use, duplication or
# disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
#
#############################################################################
```

```
#
#    Name:           rc.db2server.dls
#
#    Description:    Script to Start/Stop the Host DB2 Server HACMP
#                     Cluster manager.
#
#
#    Arguments:     $1 - db2user: is the user of the DB2 instance
#                   $2 - parm2: [start | stop] : Start or Stop option.
#                   $3 - parm3: [standby|active] : This is to indicate
#                     the node on which the script is being
#                     run, is an active or standby node for DB2 Server.
#
#    Returns:        0       success
#
###############################################################################
#
# Initialisation of variables etc.
# Change the Service_Host and Standby_Host with actual names,
Service_Host=Node-A       # Active Node for DB2 Server
Standby_Host=Node-B      # Standby node for DB2 Server

DB2user=$1
parm2=$2
parm3=$3

typeset -u parm2
HOST=`/bin/hostname -s `
PROGID=`echo $0 | sed 's%/usr/bin/%%g'`
lnndir=`lsuser -c -a home $DB2user | awk -F":" '!/#/ { print $2}'`
echo "\n`date`"

#
# Stop the DB2 instance.
#
if [[ "$parm2" = "STOP" ]] then
    echo "$PROGID - $HOST: Going to stop DB2 "
    date
    su - $DB2user -c $lnndir/sqllib/adm/db2stop force
    date
    su - $DB2user -c $lnndir/sqllib/bin/db2_kill
    sleep 15
    su - $DB2user -c killall

    #
    # Set the uname and hostname back to the Standby_Host.
    # Actually this must be done only when script is run on Standby node
    # to stop the DB2 server on Standby node.
    #
    if [[ "$parm3" = "standby" ]] then
        uname -S $Standby_Host
        hostname $Standby_Host
    fi

    # Exit
    exit 0
#
# Start the DB2 Instance.
#
elif [[ "$parm2" = "START" ]] then
    #
    # Set the uname and hostname as DB2 Server's active node. Actually this
    # setting of hostname needs to be done only when script is run on
    #Standby node during fail over.
    #
    uname -S $Service_Host
    hostname $Service_Host
```

```
    date
    echo "$PROGID - $HOST: Starting DB2 "
    su - $DB2user  -c $lnndir/sqllib/adm/db2start

    # Exit
    exit 0
else
    echo "$PROGID ERROR:: rc.db2server.dls $*"
    echo "$PROGID SYNTAX:: rc.db2server.dls [DB2_USER] [ start | stop ]
[standby | active]"

    # Exit
    exit 0
fi
```

Important: Check the ~/sqllib/samples/hacmp directory for the latest version of the rc.db2server.dls script. Customize this rc.db2server.dls script for your local environment *before* you enable the HACMP Cluster Manager to invoke this script. When this start script runs on the standby node during a failover, the script changes the host name from Node-B to Node-A. This change of the host name is required for Data Links to work properly after a failover.

Setting up the Data Links server for a hot standby configuration

You must now install and configure the Data Links server for a hot standby configuration.

Prerequisite:

Before you install and configure the Data Links server, you should understand the general concepts and administration requirements of HACMP on AIX operating systems.

Procedure:

Assume that the Data Links server resides in a cluster that uses a shared volume group called VG-2.

1. Verify that the following file systems are part of the VG-2 volume group.
   - The file system that contains the home directory for the local DB2 instance of the Data Links File Manager (DLFM). By default, this instance user name is dlfm.
   - The file system that contains the DLFM's own database. By default, this database name is DLFM_DB. If you create the DLFM_DB in a file system that is not the default, this file system must be part of the VG-2 volume group.
   - The file system that contains the dlfm_backup directory, if you are using the local disk backup option. If the DB2 Data Links Manager uses a network backup program, such as Tivoli Storage Manager or an XBSA-compliant storage application, you must set up the network backup program client on both the active node and the standby node. Register both nodes with the network backup program server.
   - All file systems defined as a Data Links File System (DLFS).

   These file systems must be accessible to both nodes.
2. Make sure that both the active node and the standby node of volume group VG-2 have the same name and the same major device number.
3. Install the DLFM and the Data Links Filesystem Filter (DLFF) software with the same configuration parameters on both nodes.

a. Install the DB2 Data Links Manager software on the active node when the VG-2 volume group is attached to this node. After the installation process finishes on the active node, complete the DLFM administration work of registering the prefixes and host databases. The active node is now ready for service.
b. Install the DB2 Data Links Manager software on the standby node, creating the temporary local file systems with the same path names as those in VG-2. After the installation process completes, shut down the DLFM and unload the DLFF kernel extension.
c. Unmount and delete the temporary file systems.

During a failover, the HACMP Cluster Manager loads the DLFS kernel extensions and mounts the DLFSs from the shared VG-2 volume group. The HACMP Cluster Manager then starts the Data Links File Manager on the standby node.

You can modify the sample script called rc.db2dls in the ~/sqllib/samples/hacmp directory.

Important: Before you run the rc.db2dls script, run one of the following two scripts depending on the UNIX shell that you are using:

INSTHOME/sqllib/db2profile (for the bash, Bourne, or Korn shell)
INSTHOME/sqllib/db2cshrc (for the C shell)

where *INSTHOME* is the home directory of the DB2 instance. By running the *INSTHOME*/sqllib/db2profile or the *INSTHOME*/sqllib/db2cshrc script, you ensure that the rc.db2dls script will execute properly.

Use the rc.db2dls script to enable the HACMP Cluster Manager to start and stop the Data Links server.

Sample rc.db2dls script:

```
#!/bin/ksh
#
# Licensed Materials - Property of IBM
#
# (C) COPYRIGHT International Business Machines Corp. 1990,2002
# All Rights Reserved
#
# US Government Users Restricted Rights - Use, duplication or
# disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
#
#########################################################################
#
#    Name:           rc.db2dls
#
#    Description:    Sample script to Start/Stop the Data Links File
#                    Manager Server.
#
#    Arguments:      $1 - instance: dlfm instance user (default dlfm)
#                    $2 - status: Either start or stop
#
#    Returns:        0       success
#
#########################################################################


#
#  Initialisation of variables etc.
#
```

```
DB2user=$1
parm2=$2
typeset -u parm2
HOST=`/bin/hostname -s `
PROGID=`echo $0 | sed 's%/usr/bin/%%g'`
lnndir=`lsuser -c -a home $DB2user | awk -F":" '!/#/ { print $2}'`
echo "\n`date`"

#
# STOP the Data Links Manager (Needs to be done before un-loading DLFS)
#
if [[ "$parm2" = "STOP" ]] then
    echo "$PROGID - $HOST: Going to stop DLFM "
    Date
    su - $DB2user -c dlfm shutdown
    sleep 5

    #
    #  Un-mount your Data Links file systems (unmount all DLFS file systems)
    #
    umount /dlfsmountpoint(s)

    #
    # Unload the DLFS kernel extension (change path of dlfs_cfg based on DB2
    # version number)
    #
    /usr/sbin/strload -u -f  /usr/opt/db2_08_01/cfg/dlfs_cfg

    # Exit
    exit 0
fi

#
# START the Data Links Manager and Load the DLFS
#
if [[ "$parm2" = "START" ]] then
    echo "$PROGID - $HOST: Starting DLFM "
    #
    # Execute dlfmfsmd for each dlfs mount point. It will create/update
    # /etc/rc.dlfs file.
    #
    /dlfm-home/sqllib/int/instance/dlfmfsmd  /dlfsmountpoint(s)

    #
    # Load the DLFS kernel extension. Un-mount and Mount all the DLFS file
    # systems by executing the rc.dlfs file created by
    # /dlfm-home/sqllib/int/instance/dlfmfsmd
    #
    /etc/rc.dlfs

    #
    # Shutdown and Restart the DLFM server. It is good to shutdown the DLFM
    # before starting it again as shutdown does cleanup of previous instance
    # of DLFM if any shared resources (ipcs) are still in use.
    #
    su - $DB2user -c dlfm shutdown
    su - $DB2user -c dlfm start

    # Exit
    exit 0
else
    echo "$PROGID ERROR:: rc.db2dls $*"
    echo "$PROGID SYNTAX:: rc.db2dls [DB2_USER] [ start | stop ]"
    exit 1
fi
```

Important: Check the ~/sqllib/samples/hacmp directory for the latest version of the rc.db2dls script. Customize the rc.db2dls script for your local environment *before* you enable the HACMP Cluster Manager to invoke this script.

Mutual takeover configuration

In this configuration, the DB2 host node and the Data Links server node reside in the same HACMP cluster. One node takes over the functionality of the other node during a failover.

Figure 8 illustrates a typical mutual takeover configuration for a DB2 host and a Data Links server.

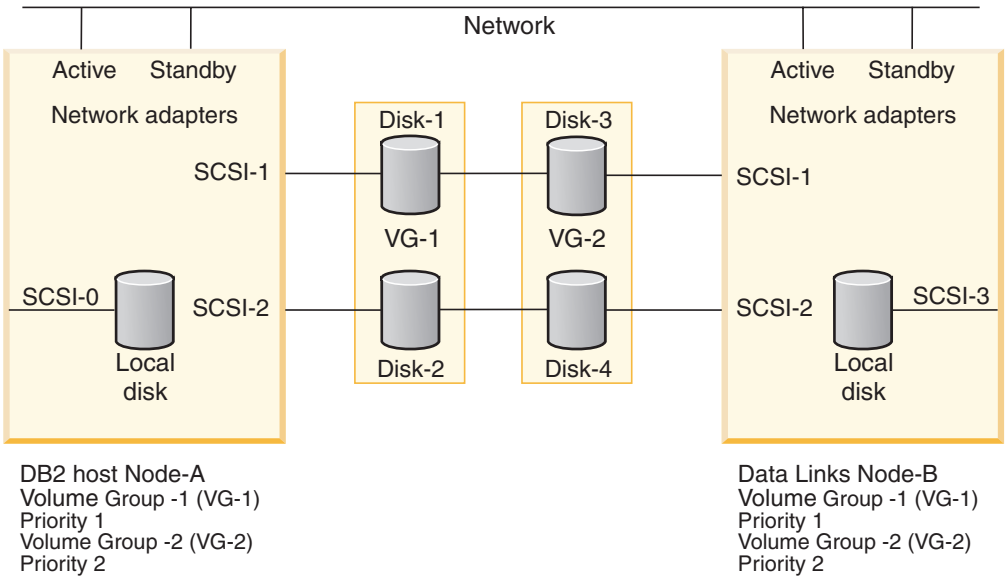


Figure 8. Mutual takeover configuration

Prerequisite:

Before you install and configure your DB2 host and Data Links server for this configuration, you should understand the general concepts and administration requirements of HACMP on AIX operating systems.

The HACMP cluster for mutual takeover has the following configuration:

- Node-A is the active node and Node-B is the standby node for the DB2 host. The configuration is *reversed* for the Data Links server: Node-*B* is the active node and Node-*A* is the standby node.
- VG-1 is the shared volume group for the DB2 host. VG-1 includes all file systems of the home directories for all DB2 instances and database directories. VG-1 requires these instances and directories during a failover, when the active node (Node-A) fails and the standby node (Node-B) resumes control. Node-A has priority over Node-B for this shared volume group. When Node-A comes back into a cluster, it re-establishes a connection with the VG-1 volume group and takes over the functionality of the DB2 host.
- VG-2 is the shared volume group for the Data Links server. VG-2 contains the file systems that are required during a failover, when the Data Links server active node (Node-B) fails and the standby node (Node-A) resumes control. Node-B has priority over Node-A for this shared volume group. When Node-B

comes back into a cluster, it re-establishes a connection with the VG-2 volume group and takes over the functionality of the Data Links server. VG-2 must contain the following file systems:

  - The file system that contains the home directory for the local DB2 instance of the Data Links File Manager. By default, this instance user name is dlfm.
  - The file system that contains the metadata database of the Data Links File Manager. (By default, this database name is DLFM_DB. If you create the DLFM_DB in a file system that is not the default, the file system must also be part of the VG-2 volume group.)
  - The file system that contains the dlfm_backup directory, if you are using the local disk backup option.
  - All file systems defined as a DLFS.

- Each node has two network adapters: one active adapter and one standby adapter.

  The active adapter on Node-A is configured with the DB2 host service IP address to which DB2 client applications connect. The standby adapter carries the boot IP address (IP-B1) during normal processing. If Node-B fails, the HACMP Cluster Manager transfers the IP and hardware addresses of the active adapter on Node-B to the standby adapter on Node-A.

  The active adapter on Node-B is configured with the Data Links File Manager service IP address to which the DB2 host server connects. The standby adapter carries the boot IP address (IP-B2) during normal processing. If Node-A fails, the HACMP Cluster Manager transfers the IP and hardware addresses of the active adapter on Node-A to the standby adapter on Node-B.

  You should configure each active adapter on Node-A and Node-B with two network addresses: the boot IP address and the service IP address. The boot IP address prevents a network address conflict during the startup of a failed node. The HACMP Cluster Manager uses the boot IP address when the failed node reboots. The HACMP Cluster Manager then revokes the service IP address from the standby node and assigns that service IP address in place of the boot IP address.

You can now install the DB2 host, the Data Links File Manager, and the Data Links Filesystem Filter software on both nodes. Use the same installation steps that you would use when setting up a hot standby configuration.

Modify the rc.db2server.dls script in the ~/sqllib/samples/hacmp directory, and use this script to enable the HACMP Cluster Manager to start and stop the DB2 host. Then you can modify the rc.db2dls sample script, which is also in this directory. Use this script to enable the HACMP Cluster Manager to start and stop the Data Links server.

Important: Check the ~/sqllib/samples/hacmp directory for the latest versions of these two scripts. Also, customize the rc.db2server.dls and the rc.db2dls scripts for your local environment *before* you enable the HACMP Cluster Manager to invoke them.

When the DB2 host server Node-A fails, the HACMP Cluster Manager sets the host name on Node-B to the host name on Node-A. This change of the host name is required by the DB2 Data Links Manager. The DNS or the local /etc/hosts file does not change the association between Node-B and its service IP address. Therefore, all the network requests to Node-B continue to go to Node-B, even

though the HACMP Cluster Manager changed the name of the host to Node-A. The Data Links File Manager service on Node-B is not affected by this host name change.

Related concepts:

- "Failure and recovery overview" on page 175

Related tasks:

- "Registering the Data Links Server with the DB2 database (AIX)" in the *Quick Beginnings for Data Links Manager*
- "Enabling file system sharing (AIX, Solaris Operating Environment)" on page 50
- "Loading, querying, and unloading a DLFF (AIX)" on page 74

Related reference:

- "File system backup and restore recommendations" on page 178

Part 4. Troubleshooting

Chapter 10. 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 175

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 175
- "db2_recon_aid utility" on page 180

DB2 Data Links Manager system setup and backup recommendations

The following system setup and backup procedures are recommended for easier recovery.

1. Place the Data Links File Manager (DLFM) database (usually called DLFM_DB), any file systems under control of the Data Links Filesystem Filter (DLFF), the server archive area, 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 regular basis. In the event of disk failure, this backup will provide added recoverability.

   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.

   For an example of a high availability file system backup method for an AIX Journaled File System (JFS), see "An alternative approach to backing up a Journaled File System on AIX systems" on page 177.
3. Back up the archive area or archive server on a regular basis, if possible.

The archive area itself is used as a repository for DB2 Data Links Manager-initiated backups of DLFM data, and possibly linked files. Keeping these backups might result in some data redundancy, but having them will provide added recoverability in the event of disk failure.

4. 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 database (DLFM_DB) on the Data Links server at regular intervals. Coordinate the timing of the database backups so that they occur together, and ensure that their data is transactionally consistent.

An alternative approach to backing up a Journaled File System on AIX systems

This section describes a high availability backup method for an AIX Journaled File System environment. Using this approach enables you to perform a file system backup without stopping your DB2 Data Links Manager.

This backup method uses the following programs:

- `online.sh`
- `quiesce.c`

These programs are provided with your DB2 Data Links Manager installation software and are located under the `/sqllib/samples/dlfm` directory. The `online.sh` script, which calls the `quiesce.c` program, performs the following actions:

- Temporarily deactivates all the tables in databases that are registered with the DB2 Data Links Manager. This stops any new Data Links Manager activity.
- Unmounts and remounts the file system as a read-only file system.
- Performs a file system backup.
- Unmounts and remounts the file system as a read-write file system.
- Resets and reactivates the database tables.

Prerequisites:

To use the `online.sh` script, you must have a catalog entry on the DB2 Data Links Manager node for each database that is registered with the DB2 Data Links Manager. You must also have the complete entry for the Data Links File System (DLFS) on the `/etc/filesystems` file.

Procedure:

To execute the backup, perform the following steps:

1. Copy the `quiesce.c` CLI source file and the `online.sh` shell script to directory of your choice on the Data Links server where the DLFS is located.

   You will modify and work with these copies in the rest of this procedure.
2. Compile `quiesce.c` using the following command:

   ```
   xlC -o quiesce -ldb2 -L$HOME/sqllib/lib -I$HOME/sqllib/include quiesce.c
   ```
3. Modify the `online.sh` script to suit your environment:
   a. Select a backup command for the do_backup function of the `online.sh` script. At the top of the do_backup function, there are several commented lines of backup options. Remove the comment characters from the lines you want to use, and insert any necessary backup commands and parameters.

b. At the top of the script, replace the specified default DLFM_INST environment variable with your Data Links File Manager instance name.
c. On the next line, replace the default PATH_OF_EXEC environment variable with the path where your `quiesce.c` executable resides.

4. From the Data Links server where the DLFS is located, run the script as follows:

```
online.sh <filesystem_name>
```

Related concepts:

- "Failure and recovery overview" on page 175

Related reference:

- "File system backup and restore recommendations" on page 178

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 175
- "Bringing the file system directory hierarchy to the current point in time" on page 178

Related reference:

- "Data Links server machine failure" on page 175

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® 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 Manager 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/RENAME> Object type = <DIR/FILE> Path = <fully qualified
source name, destination 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
- *RENAME* indicates the file was renamed

where Object type can be:

- *DIR* the directory
- *FILE* the file

and where Path is the fully qualified path of the file or directory. If the action was RENAME, the destination name is displayed after the path information.

On Windows® systems, the directory changes are logged in the *x:*\sqllib\dlfm\fsysadm.log file, where *x:* represents the drive where you installed DB2® 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/RENAME>
  Object type = <DIR/FILE> Path = <fully qualified source name, destination 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 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
- *RENAME* indicates the file was renamed

where Object type can be:
- *DIR* the directory
- *FILE* the file

and where Path is the fully qualified path of the file or directory. If the action was RENAME, the destination name is displayed after the path information.

(*) For the definitions of these hexadecimal values, refer to the Access Control Entry structures in the Microsoft® SDK documentation for Windows NT® and Windows 2000.

Related concepts:
- "Failure and recovery overview" on page 175

db2_recon_aid utility

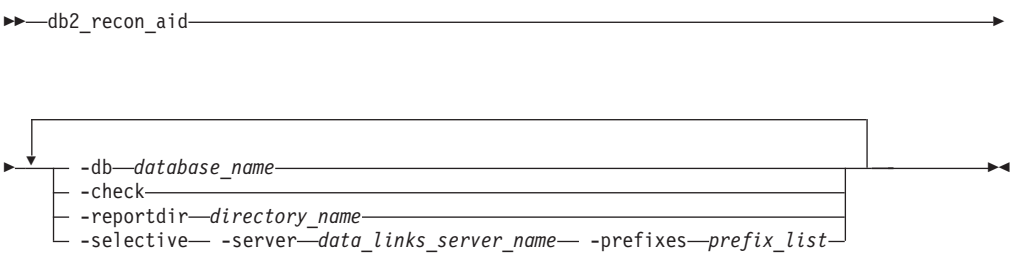
The `db2_recon_aid` utility provides a mechanism for checking and running `RECONCILE` on tables of a database that are potentially inconsistent with the `DATALINK` file data on the file server.

Like the RECONCILE utility, the db2_recon_aid utility must be run on a DB2 server containing tables with DATALINK columns to be reconciled.

On AIX systems or Solaris Operating Environments, the `db2_recon_aid` utility is located in the *INSTHOME*/sqllib/adm directory, where *INSTHOME* is the home directory of the instance owner.

On Windows systems, the `db2_recon_aid` utility is located in *x:*\sqllib\bin directory, where *x:* is the drive where you installed DB2 Data Links Manager.

To run the `db2_recon_aid` utility, use the following syntax:



database_name
A required value, which specifies the name of the database containing the tables with DATALINK columns that need to be reconciled.

-check A parameter that instructs the utility to list the tables that might need

reconciliation. If you use this parameter, no reconcile operations will be performed. This parameter is required when the **-reportdir** parameter is not specified.

-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 175

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® 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

DB2 database is lost or was accidentally dropped, but DB2 backup and log files are available.
 	
Important: When a database gets dropped, it happens within the period of time as specified by the DB2 database configuration parameter *dl_time_drop*.
 	
In this example, the DB2 database has **not** been dropped from the Data Links server.
	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.
	`db2 "restore database CROWN"`
	`db2 "rollforward database CROWN to end of logs and stop"`
	`db2 "connect to CROWN"`
	
	2. Place all tables with DATALINK columns into Datalink_Reconcile_Pending (DRP) state with the following commands:
	`db2 set integrity for <table> to datalink reconcile pending`
	`db2 set integrity for <table> datalink reconcile pending immediate unchecked`
	`db2 reconcile <table> dlreport <filename>`

Scenario	Recovery Steps

DB2 database was explicitly dropped, but DB2 backup and log files are available.
 	
Important: When a database gets dropped, it happens within the period of time as specified by the DB2 database configuration parameter *dl_time_drop*.
	Requirement: You *must* take a backup of your DLFM_DB database before updating it as described in the following procedure.
	
	Recommendation: Work with IBM® Service in updating your DLFM_DB database. This database is an essential part of your Data Links Manager configuration.
	1. Ensure that the drop database operation is complete, and that all files associated with that database have been unlinked.
	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.
	`db2 "connect to dlfm_db"`
	`db2 "select dbid, dbname, dbinst, hostname from dfm_dbid"`
	`db2 "update dfm_dbid set action=5 where dbid=x'35B3D7BE0006BF7B'"`
	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.
	`db2 "restore database CROWN"`
	`db2 "rollforward database CROWN to end of logs and stop"`
	`db2 "connect to CROWN"`
	4. For each table placed in DRNP state in step 3, enter the following commands:
	`db2 set integrity for <table> to datalink reconcile pending`
	`db2 set integrity for <table> datalink reconcile pending immediate unchecked`
	`db2 reconcile <table> dlreport <filename>`

The DLFM_DB database is lost, but the backup and all log files for the DLFM_DB database are available.	1. On the Data Links server enter the following commands:
	`db2 "restore database dlfm_db"`
	`db2 "rollforward database dlfm_db to end of logs and stop"`
	2. On the DB2 host, enter the following command to run the **db2_recon_aid** utility. This utility automatically runs RECONCILE for each table with URL file references to the affected Data Links server:
	`db2_recon_aid -db CROWN -reportdir <dirpath>`
	`-selective -server <dlm_hostname> -prefixes <dlfs_prefix>`
	• **dlm_hostname** is the registered IP hostname of the affected Data Links Manager
	• **dlfs_prefix** is the registered prefix corresponding to the affected Data Links File System (DLFS)

The DLFM_DB database is lost, a backup of the DLFM_DB database is available, but not all of the log files are available.	1. On the Data Links server enter the following commands:
	`db2 "restore database dlfm_db"`
	`db2 "rollforward database dlfm_db to end of logs and stop"`
	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.
	`db2 "restore database CROWN"`
	`db2 "rollforward database CROWN to end of logs and stop"`
	`db2 "connect to CROWN"`
	3. Place all tables with data link values into DRP state by entering the following commands:
	`db2 set integrity for <table> to datalink reconcile pending`
	`db2 set integrity for <table> datalink reconcile pending immediate unchecked`
	`db2 reconcile <table> dlreport <filename>`

Scenario	Recovery Steps

The Data Links File System (DLFS) is lost.	1. Restore the DLFS from your storage manager.
	2. On the DB2 host, enter the following command to run the **db2_recon_aid** utility. Notice that by using the **-selective** options, you can perform reconciliation for just the DLFS that was lost.
	`db2_recon_aid -db CROWN -reportdir <dirpath>`
	`-selective -server <dlm_hostname> -prefixes <dlfs_prefix>`
	• **dlm_hostname** is the registered IP hostname of the affected Data Links Manager
	• **dlfs_prefix** is the registered prefix corresponding to the affected Data Links File System (DLFS)

The DLFM backup directory is lost.	Restore the DLFM backup directory from your storage manager.

The DLFS file system and the DLFM backup directory are lost.	1. Restore the DLFM backup directory from your storage manager.
	2. Restore the DLFS from your storage manager.
	3. On the DB2 host, enter the following command to run the **db2_recon_aid** utility. Notice that by using the **-selective** options, you can perform reconciliation for just the DLFS that was lost.
	`db2_recon_aid -db CROWN -reportdir <dirpath>`
	`-selective -server <dlm_hostname> -prefixes <dlfs_prefix>`
	• **dlm_hostname** is the registered IP hostname of the affected Data Links Manager
	• **dlfs_prefix** is the registered prefix corresponding to the affected Data Links File System (DLFS)

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.	1. On the Data Links server enter the following commands:
	`db2 "restore database dlfm_db"`
	`db2 "rollforward database dlfm_db to end of logs and stop"`
	2. Restore the DLFM backup directory from your storage manager.
	3. Restore the DLFS from your storage manager.
	4. On the DB2 host, enter the following command to run the **db2_recon_aid** utility. Notice that by using the **-selective** options, you can perform reconciliation for just the DLFS that was lost.
	`db2_recon_aid -db CROWN -reportdir <dirpath>`
	`-selective -server <dlm_hostname> -prefixes <dlfs_prefix>`
	• **dlm_hostname** is the registered IP hostname of the affected Data Links Manager
	• **dlfs_prefix** is the registered prefix corresponding to the affected Data Links File System (DLFS)

Scenario	Recovery Steps

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.	1. On the Data Links server enter the following commands:
	`db2 "restore database dlfm_db"`
	`db2 "rollforward database dlfm_db to end of logs and stop"`
	2. Restore the DLFM backup directory from your storage manager.
	3. Restore the DLFS from your storage manager.
	4. On the DB2 host, enter the following command to run the **db2_recon_aid** utility. Notice that by using the **-selective** options, you can perform reconciliation for just the DLFS that was lost.
	`db2_recon_aid -db CROWN -reportdir <dirpath>`
	`-selective -server <dlm_hostname> -prefixes <dlfs_prefix>`
	• **dlm_hostname** is the registered IP hostname of the affected Data Links Manager
	• **dlfs_prefix** is the registered prefix corresponding to the affected Data Links File System (DLFS)

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.	1. On the Data Links server enter the following commands:
	`db2 "restore database dlfm_db"`
	`db2 "rollforward database dlfm_db to end of logs and stop"`
	2. Restore the DLFM backup directory from your storage manager.
	3. Restore the DLFS from your storage manager.
	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.
	`db2 "restore database CROWN"`
	`db2 "rollforward database CROWN to end of logs and stop"`
	`db2 "connect to CROWN"`
	5. For each table placed in DRNP state in step 4, enter the following commands to place them in DRP state:
	`db2 set integrity for <table> to datalink reconcile pending`
	`db2 set integrity for <table> datalink reconcile pending immediate unchecked`
	`db2 reconcile <table> dlreport <filename>`
	Note: 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.

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 180

Chapter 11. Troubleshooting problems

This chapter describes methods for solving DB2 Data Links Manager problems, provides a list of problems that you might encounter, and suggests solutions to those problems.

An overview of the troubleshooting process

Problem determination is best approached by a process of elimination and refinement of the available data to arrive at a conclusion (the problem location and a solution).

To recognize that a certain problem condition exists, you must understand the environment where the problem condition has occurred. In a DB2® Data Links Manager environment, a problem can be caused by an error in an application, a problem on the DB2 host, or a problem with one of the components on the Data Links server.

The following steps describe a process to follow when you are troubleshooting a problem in a DB2 Data Links Manager environment. If necessary, use this process to troubleshoot both the DB2 host and the Data Links server environments.

1. Create a detailed, written description of the problem. The more detail that you can provide about a problem, the better your understanding of the problem.

   Your problem description should include at least these items:

   - All error codes, error conditions, and any applicable reason codes.

     Tip: Explanations for many of the error conditions that are associated with problems can be found throughout the DB2 library.

     – For error messages that begin with the characters `DB2` or `SQL`, see the *DB2 Universal Database™ Message Reference*, Volume 1 (GC09-4840) and Volume 2 (GC09-4841).

     – For error messages that begin with the characters `DLFM`, see DLFM messages.

     – For a list of DB2 internal return codes, see the "Internal Error Codes - Version 8" technote (number 1066178), which is available from the DB2 Universal Database Support Web site at www.ibm.com/software/data/db2/udb/support.html.

   - The actions that preceded the problem.
   - An accurate description of the problem itself.

2. Determine whether the problem can be reproduced or whether it was a one-time occurrence.

   If the problem can be reproduced, determine and write down the actions that are required to reproduce the problem.

3. Identify the source (or cause) of the problem.

   Use the following series of questions as a starting point to complete this task:

   - Is the system configuration working as designed?

     For example, perhaps someone did not understand the behavior of the system, or the system is working as intended.

- Is your system configuration properly supported by your current hardware and software?

  For example, it is possible that your configuration is not intended to run with the hardware or software that is currently in use.
- Is the problem a result of a human error?

  For example, perhaps there was a data entry or typing error, or the wrong command was entered.
- Is the problem occurring on a DB2 host server or on a specific database?
- Is the problem occurring on a Data Links server?

  Is the problem occurring on the DLFM (Data Links File Manager) server, on the Data Links File System (DLFS), or on the native file system?

4. Provide a solution to address the problem.
   - An application or system environment change might be required if the problem is a result of any of these causes:
     - Human error
     - System is working as designed
     - Unsupported environment or configuration
   - If the problem is the result of a defect in the DB2 or DB2 Data Links Manager software, notify IBM® Service. IBM Service will either provide a software update or a workaround.

Related reference:

- "Troubleshooting commands (UNIX)" in the *Troubleshooting Guide*
- "Diagnostic tools for Windows NT, Windows 2000 and Windows XP operating systems" in the *Troubleshooting Guide*
- "Required diagnostic information for problem analysis" on page 188
- "DLFM messages" in the *Message Reference Volume 1*

Required diagnostic information for problem analysis

The troubleshooting process requires that you gather a basic set of diagnostic information about a problem. You will use this information to both describe and diagnose the problem.

You might also need to provide this diagnostic information to IBM Service if you require assistance.

The following set of information is required to analyze a problem in the DB2 Data Links Manager environment for troubleshooting purposes:

- The date and approximate time that the problem occurred.
- An accurate problem description.
- A description of the actions that preceded the problem.
- All SQL error codes and corresponding reason codes (RCs), if any were returned.

  For example, `SQL0357N, RC = "03"` is a possible error and reason code returned when initiating the Reconcile utility.
- All system error codes, if any were returned.
- The specific machine or machines where the problem was encountered, and the purpose of each machine in your DB2 Data Links Manager system.

In other words, is the machine or machines where the problem occurred being used as a DB2 host, as a Data Links server, or as both a DB2 host and as a Data Links server?

- The operating and file systems of the specific machine or machines where the problem was encountered.
- The database manager configuration settings for the DB2 host and Data Links server machines where the problem was encountered. Invoke the following command on each machine to collect the file information: `db2 get dbm cfg`.
- The database configuration settings for the DB2 host and Data Links server machines where the problem was encountered. Invoke the following command on each machine to collect the file information: `db2 get db cfg for database name`.

  database name is either the name of a specific database on the DB2 host where the problem occurred, or the name of the DLFM (Data Links File Manager) database on the Data Links server (named DLFM_DB by default).
- The DB2 software code level on the DB2 host and Data Links server machines where the problem was encountered. Collect this information by invoking the **db2level** command on each machine.
- The DB2 diagnostic log file (db2diag.log) from both the DB2 host and Data Links server machines where the problem was encountered. The db2diag.log file contains significant diagnostic information, and this file must be collected on *all* machines involved in processing when the problem was encountered.

  The location of the db2diag.log file is controlled by the DB2 server configuration parameter DIAGPATH. DIAGPATH is a database configuration parameter that points to the directory location for placing diagnostic data. Therefore, the diagnostic data directory on your system might be different from the default diagnostic data directory.

  – **On AIX and the Solaris Operating Environment**: db2diag.log 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**: db2diag.log is located, by default, in the *x:*\sqllib*instance* directory. *x:* is the drive where you installed DB2 or the DB2 Data Links Manager, and *instance* is the name of the instance that is owned by the DB2 database owner or the Data Links Manager Administrator (dlfm, by default).

  Recommendation: If the problem is reproducible, change your machine configuration settings to capture the maximum amount of information in the DB2 diagnostic logchange your machine configuration settings to capture the maximum amount of information in the DB2 diagnostic log, then recapture the diagnostic information in db2diag.log.

  See the Interpreting the administration logs topic for more details about the information that is written to the db2diag.log file.
- Collect any dump files mentioned in db2diag.log.

  Dump files are named *x.nnn*, where *x* identifies the process that produced the dump file, and *nnn* identifies the database partition number (000 on single partition databases).
- Collect any trap files in the DIAGPATH directory on both the DB2 host and Data Links server machines where the problem was encountered.

  On UNIX-based systems, the first letter in trap file names is ″t″, followed by a process identifier (PID). The file extension is the partition number (or 000 on single partition databases).

On Windows systems, each trap file is named P*xxxxx.yyy* where *xxxxx* is the PID and *yyy* is the database partition number (or 000 on single partition databases). If the trap file is generated because of an exception, it will have the extension TRP.

Tip: Explanations for many of the error conditions associated with problems can be found throughout the DB2 library.

- For error messages that begin with the characters `DB2` or `SQL`, see the *DB2 Universal Database Message Reference*, Volume 1 (GC09-4840) and Volume 2 (GC09-4841).
- For error messages that begin with the characters `DLFM`, see DLFM messages.

Related concepts:

- "First failure data capture" in the *Troubleshooting Guide*
- "Administration notification logs – location" in the *Troubleshooting Guide*
- "Interpreting the administration logs" in the *Troubleshooting Guide*
- "Dump Files" in the *Troubleshooting Guide*
- "Trap Files" in the *Troubleshooting Guide*
- "DB2 traces" on page 190

Related tasks:

- "Changing the diagnostic level for error message log files" on page 79

Related reference:

- "diagpath - Diagnostic data directory path configuration parameter" in the *Administration Guide: Performance*
- "Data Links File Manager dump utility (dlfm_dump)" on page 191
- "DLFM messages" in the *Message Reference Volume 1*

DB2 traces

When you report problems to IBM® Service, you might be asked to perform a DB2® trace to provide the service personnel with details of your DB2 Data Links Manager environment. DB2 traces can be especially useful for analyzing a recurring and reproducible problem.

The Data Links server contains its own DB2 database, which is used as a logging manager to track all linked files. Therefore, you can take a DB2 trace on both the DB2 host and Data Links server machines where a problem was encountered, if necessary. In certain situations, you might need to run a *concurrent* DB2 trace on several machines. For example, you might encounter a communication problem between a DB2 host and a Data Links server machine.

During the process of performing a DB2 trace, all actions and any relevant parameter values get logged. Traces should be taken when there is minimum activity on the machine to prevent the capture of unnecessary information.

The process of performing a trace has a global effect on the behavior of a DB2 instance. The degree of performance degradation is dependent on the type of problem and on how many resources are being used to gather the trace information.

Related concepts:

- "DB2 trace (db2trc)" in the *Troubleshooting Guide*

Related reference:

- "db2trc - Trace Command" in the *Command Reference*

Data Links File Manager dump utility (dlfm_dump)

When you report problems to IBM Service, you might be asked to run the Data Links File Manager (DLFM) dump utility, dlfm_dump, to provide the service personnel with details of your DB2 Data Links Manager environment.

The dlfm_dump utility enables you to obtain a "snapshot" of significant data that is stored in the Data Links File Manager's own database (called `DLFM_DB` by default at installation time). With this data, you can examine various DLFM system configuration details, plus other DLFM-related data on a Data Links server. The output is stored in a file in the directory where the dlfm_dump utility is invoked. The data that gets placed into the output file includes:

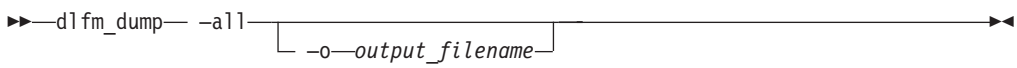
- The version of the current `DLFM_DB`.
- The keys currently in use for access-token generation.
- Security control information.
- Registered databases and prefixes.
- Data Links File System (DLFS) directory tree structures.
- DATALINK columns that reference this Data Links server.
- DB2 backups involving linked files on this Data Links server.
- Lists of all linked and unlinked files on this Data Links server.

Important:

- If the DLFM is managing a large number of files, both those that are currently linked as well as those that were previously linked, the output file could be very large. Ensure that you have sufficient space in the file system where the output file is to be written. You can use the amount of space that the `DLFM_DB` occupies, or that a backup of the `DLFM_DB` occupies, to estimate the space that is required for the dlfm_dump output file.
- Because the output of the dlfm_dump utility contains sensitive security information, ensure that you place the output file in a secure directory.

You use the following syntax to invoke the dlfm_dump utility.

Requirement: You must execute this command from the Data Links server using the Data Links Manager Administrator user ID.



–all A required parameter, which specifies that all significant DLFM definitions and data are to be placed into an output file.

–o *output_filename*
An optional parameter, which specifies the name of the output file where the data will be saved. The default output file name is `dlfm_dump.log`.

The dlfm_dump utility output does not include temporary data that the DLFM_DB might be maintaining, such as data that is stored during transaction processing.

Note: The dlfm_dump utility output is minimally formatted, and is primarily intended to assist in diagnosing problems, not for general reporting purposes. To produce more readable output of Data Links server data, use any or all of the following commands:

- **dlfm retrieve**
- **dlfm list registered databases**
- **dlfm list registered directories**
- **dlfm list registered prefixes**
- **dlfm list registered replication access control**
- **dlfm list registered users**
- **dlfm list upd_in_progress files for db**
- **dlfm list upd_in_progress files for prefix**

Related reference:

- "dlfm list registered databases command" on page 218
- "dlfm list registered prefixes command" on page 219
- "dlfm list registered replication access control command" on page 220
- "dlfm list registered users command" on page 220
- "dlfm list upd_in_progress files for db command" on page 221
- "dlfm list upd_in_progress files for prefix command" on page 222
- "dlfm list registered directories command" on page 218
- "dlfm retrieve command" on page 224

Problem symptoms and potential solutions

DB2 Data Links Manager problems can usually be classified into the following categories:

- Data Links File Manager (DLFM) problems on a Data Links server. For example, the DLFM will not start, or it cannot communicate with the DB2 host.
- DB2 host server problems.
- Data Links File System (DLFS) problems on a Data Links server. For example, client workstations are unable to access linked files, or the DLFF (Data Links Filesystem Filter) cannot be loaded.

The following sections describe the more common problems, typical causes and symptoms, and potential solutions.

Data Links File Manager problems

This section describes common problems that can occur with the Data Links File Manager (DLFM) on a Data Links server in a DB2 Data Links Manager environment.

DLFM does not start

If required resources that the Data Links File Manager (DLFM) needs to run are not available, then the DLFM will not start. Depending on the specific reasons that the DLFM cannot start, a DLFM101E error message might get written to the db2diag.log file.

Typical causes: This problem most commonly occurs when attempting to restart the DLFM after invoking the **dlfm stop** command. However, it can also occur if the DLFM cannot access its own DB2 database (called DLFM_DB by default), communication services cannot start, or if the Data Links Filesystem Filter (DLFF) device driver is not loaded.

Typical symptoms: After running the **dlfm start** command on the Data Links server, one or more of the following conditions exists:

- The **dlfm see** command (on UNIX systems) or the Task Manager (on Windows) shows only a subset of the required DLFM processes running on the Data Links server, or shows no DLFM processes running at all.
- Your application program receives a SQL0357N return code, 03 reason code, when attempting to perform an SQL SELECT, INSERT, or UPDATE on a DATALINK column value.
- The db2diag.log file on the DB2 host server contains a message that indicates that DLFM is unreachable.
- The db2diag.log on the DB2 host server contains a message that indicates that restart recovery is pending or in progress for DLFM.

Potential solutions: Perform the following steps to solve the problem:

1. Log on to the Data Links server as the Data Links Manager Administrator.
2. Issue the **dlfm shutdown** command. The shutdown process can take from 30 to 40 seconds to fully complete. After the shutdown process completes, issue the **dlfm start** command.
   - If the DLFM starts successfully, no further action is required.
   - If the DLFM does not start, proceed to the next step.
3. Ensure that the port number specified in the DLFM_PORT registry variable is not being used by another process. Use the **db2set -all** command to view all DB2 registry variables.
   - If the port designated in the DLFM_PORT registry variable is not being used by another process, proceed to the next step.
   - If another process is using the DLFM port, either change the DLFM_PORT variable to another, unused port, or stop the other process that is using the DLFM port. Then, restart the DLFM using the **dlfm restart** command.
     – If the DLFM starts successfully, no further action is required.
     – If the DLFM does not start, proceed to the next step.
4. Validate that the DLFM's own database instance (*dlfm* by default at installation time) can be started by issuing the **db2start** command.
   - If the command succeeds, proceed to the next step.
   - If the command fails, then it is likely that the DLFM cannot start because it cannot start the dlfm instance for some reason.

     Collect the required diagnostic information, including any error messages that result from issuing the **db2start** command, and try to diagnose why the DLFM database instance cannot be started. You can attempt to fix the instance failure, or you can call IBM Service for assistance.
5. Verify that the DLFM's own database (named DLFM_DB) is usable by connecting to it using the **db2 connect** command. For example: `db2 connect to dlfm_db`.

   Disconnect from the DLFM_DB after you finish testing by using the **db2 connect reset** command.
   - If the connection attempt is successful, proceed to the next step.

- If the connection attempt is not successful, then it is likely that the DLFM cannot start because it cannot connect to the DLFM_DB for some reason.

  Collect the required diagnostic information, including the resulting "connection failed" message, and try to diagnose why the DLFM database connection is failing. You can attempt to fix the connection failure, or you can call IBM Service for assistance.

6. Query the DLFF device driver to ensure that it is loaded (UNIX) or registered (Windows).
   - For detailed instructions on how to query, load, and unload the DLFF device driver on AIX, see the Loading, querying, and unloading a DLFF topic.
   - For detailed instructions on how to query, load, and unload the DLFF device driver in the Solaris Operating Environment, see the Loading, querying, and unloading a DLFF topic.
   - For detailed instructions on how to query, register, and de-register the DLFF device driver on Windows, see the Registering, querying, and de-registering a DLFF topic.
7. Ensure that the device driver is loaded.
   - If the query shows that the device driver is loaded, proceed to step 8.
   - If the query shows that the device driver is not loaded, attempt to reload it. If the reload attempt fails, perform the following steps:
     a. Unload the DLFF driver.

        You can ignore any error messages that might display. The purpose of this step is to "clean up" any leftover system configurations that might be causing problems.

     b. Reload the DLFF driver.

        If this step fails, the next step will show that the driver is not loaded.

     c. Query the DLFF driver to ensure that it is loaded.
        - If the query shows that the device driver is loaded, attempt to start the DLFM again. If the DLFM starts successfully, no further action is required.
        - If the query shows that the device driver is *not* loaded, proceed to step 8.
8. Examine the db2diag.log file on the DB2 host server and the Data Links server for additional messages. The log might contain messages with specific instructions or actions to take to try to resolve the problem.

   The location of the db2diag.log file is controlled by the DB2 server configuration parameter DIAGPATH.
   - For error messages that begin with the characters `DB2` or `SQL`, see the *DB2 Universal Database Message Reference*, Volume 1 (GC09-4840) and Volume 2 (GC09-4841).
   - For error messages that begin with the characters `DLFM`, see DLFM messages.
9. If the problem persists, report it to IBM Service.

Related tasks:

- "Loading, querying, and unloading a DLFF (AIX)" on page 74
- "Loading, querying, and unloading a DLFF (Solaris Operating Environment)" on page 74
- "Registering, querying and de-registering a DLFF (Windows operating system)" on page 75

Related reference:

- "Required diagnostic information for problem analysis" on page 188
- "DLFM messages" in the *Message Reference Volume 1*

DLFM does not start automatically after machine reboot

Potential solutions (UNIX): Perform the following steps to solve the problem.

1. Log on to the Data Links server where the problem is occurring as the Data Links Manager Administrator.
2. Ensure that /etc/inittab has an entry to start the dlfsmount script file.

   If an entry for dlfsmount exists, proceed to the next step.

   If an entry for dlfsmount does not exist, you must add one to the /etc/inittab file manually. Log on to the Data Links server as ″root″ before updating the /etc/inittab file.

   - On AIX, dlfsmount is installed in /usr/opt/db2_08_01/instance/dlfsmount.
   - On the Solaris Operating Environment, dlfsmount is installed in /opt/IBM/db2/V8.1/instance/dlfsmount.

   After you have added an entry to start dlfsmount, reboot the machine. If DLFM starts, then no further action is required. If DLFM does not start, proceed to the next step.
3. Verify that the DLFM_AUTOSTART registry variable is set to YES.
   a. Log on to the Data Links server as the Data Links Manager Administrator.
   b. Issue the command **db2set DLFM_AUTOSTART** to view the current registry variable setting.
      - If the setting is YES, proceed to step 4.
      - If the setting is NO, change it to YES by issuing the following command: **db2set DLFM_AUTOSTART=YES**.
   c. Reboot the machine.
      - If the DLFM starts successfully, no further action is required.
      - If the DLFM does not start, proceed to step 4.
4. Ensure that all Data Links File Systems (DLFSs) are being mounted correctly at start up time.

   AIX The mount option in /etc/filesystems for the DLFSs must be set to `false`. You must mount all DLFSs by issuing the **mount -v dlfs** command, which is done automatically by the dlfsmount script when the machine is rebooted.

   Solaris Operating Environment
   The mount option in /etc/vfstab for the DLFSs must be set to `false`. You must mount all DLFSs by issuing the **mount -F dlfs** command, which is done automatically by the dlfsmount script when the machine is rebooted.

   - If the DLFS mount entry is correct, proceed to step 5.
   - If the DLFS mount entry is not correct, correct it in the appropriate file, then reboot the machine. If the DLFM starts successfully, no further action is required. If the DLFM does not start, proceed to step 5.
5. If the problem persists, report it to IBM Service.

Potential solutions (Windows): Perform the following steps to solve the problem.

1. Log on to the Data Links server where the problem is occurring as the Data Links Manager Administrator.
2. Open the Windows **Services** control panel.
3. Ensure that the **Data Links File Manager** entry is present.
   - If the **Data Links File Manager** entry is present, proceed to the next step.
   - If the **Data Links File Manager** entry is not present, you must re-install the DB2 Data Links Manager software. See the Before you install DB2 Data Links Manager topic for detailed instructions. If the problem persists after you re-install the software, report it to IBM Service.
4. Ensure that the **Startup type** of the **Data Links File Manager** entry is set to **Automatic**.
   - If the **Startup type** is set to **Automatic**, proceed to the next step.
   - If the **Startup type** is set to **Manual** or **Disabled**, change it to **Automatic**, then reboot the machine. If the DLFM starts successfully, no further action is required. If the DLFM does not start, proceed to the next step.
5. Verify that the **Status** of the **Data Links File Manager** entry is set to **Started**.
   - If the **Status** is set to **Started**, proceed to the next step.
   - If the **Status** is blank, check the System Log and Application Log in the Windows Event Viewer for messages from the Data Links Filesystem Filter (DLFF) or the Data Links File Manager (DLFM). Follow the instructions provided with those messages, then reboot the machine. If the DLFM starts successfully, no further action is required. If the DLFM does not start, proceed to the next step.
6. Verify that at least one drive is enabled as a Data Links File System (DLFS), as described in the Listing registered drives topic.
   - If you can verify that a drive is already enabled as a DLFS, proceed to step 7.
   - If you cannot verify that a DLFS drive exists, create and configure one now.

     A DLFS drive must be configured as shareable, have an assigned share name, and be under the control of the Data Links Filesystem Filter (DLFF). For instructions on how to perform these tasks, see:

     a. The Enabling file system sharing topic.
     b. The Activating the Data Links Filesystem Filter topic.
     c. The dlff add command topic.

     After you create a DLFS drive, try to start the DLFM again. If the DLFM starts successfully, no further action is required. If the DLFM still does not start, proceed to step 7.
7. If the problem persists, report it to IBM Service.

Related concepts:

- "Before you install DB2 Data Links Manager (Windows)" in the *Quick Beginnings for Data Links Manager*

Related tasks:

- "Activating the Data Links Filesystem Filter" on page 51
- "Enabling file system sharing (Windows)" on page 51
- "Listing registered drives (Windows operating systems)" on page 73

Related reference:

- "dlff add command (Windows operating system)" on page 233

DB2 host server problems

This section describes common problems that can occur with the DB2 host server in a DB2 Data Links Manager environment.

DB2 host server cannot communicate with DLFM

Typical causes: A DB2 host server might be unable to communicate with the Data Links File Manager (DLFM) on a Data Links server for a variety of reasons.

When a DB2 database that references one or more DB2 Data Links Managers is started and a DATALINK column value is accessed, the DB2 host server attempts to connect to one or more of the Data Links servers that are registered with it. If the DLFM on a particular Data Links server is not running, an error is written to the db2diag.log file on the DB2 host server. Database users will not see any problems until they attempt to access a DATALINK column value. The attempt will result in an SQL0357N error.

Communication problems can also occur if a DB2 database is not registered with a Data Links server, if the database was incorrectly registered with a Data Links server, or if the DB2 host server DATALINKS database configuration parameter was set to NO. Communication problems due to nonexistent or incorrect registration between a database and a Data Links server are usually indicated with the SQL0368N error.

Finally, the version of the DB2 software installed on a Data Links server and a DB2 host server must be compatible at both the release *and* fixpack level. If the DB2 software is not compatible, the DLFM on the Data Links server will refuse a connection from the DB2 host server. For more information about valid combinations of DB2 and DB2 Data Links Manager software, see the Related concepts section at the end of this topic.

Typical symptoms: After invoking the **dlfm start** command on the Data Links server, the **dlfm see** command (on UNIX systems) or the Task Manager (on Windows systems) shows that DLFM processes are running. However, one or both of the following conditions exist:

- Your application program receives a SQL0357N return code, 03 reason code, when attempting to perform an SQL SELECT, INSERT, or UPDATE on a DATALINK column value.
- The db2diag.log file on the DB2 server contains messages that indicate that the DLFM is unreachable.

Potential solutions: Perform the following steps to solve the problem:

1. Log on to the Data Links server where the problem is occurring as the Data Links Manager Administrator.
2. Verify that the correct database, instance, and hostname are registered with the DLFM by issuing the command **dlfm list registered databases**.
   - If the information is correct, proceed to step 3 on page 198.
   - If the information is incorrect, add the correct information by using the **dlfm add_db** command.

     You can remove the incorrect registration information at a later time by using the **dlfm drop_dlm** command.

     After you add the correct information, attempt the communication process again. If you are successful, then no further action is required. If you are unsuccessful, proceed to step 3 on page 198.

3. Verify that the Data Links server is correctly registered with the DB2 database:
   a. Connect to the DB2 database where the problem is occurring.
   b. Issue the command **db2 list datalinks managers for database** *dbname*, where `dbname` is the name of the database.
   c. Check the hostname and port number that is registered.
      - If the information is correct, proceed to step 4.
      - If the information is not correct, you must remove the incorrect Data Links server information, then register the correct Data Links server information with the database.

        Use the **db2 drop data links manager** command to remove (drop) the incorrect Data Links server information. Then, use the **db2 add data links manager** command to register (add) the correct Data Links server information.

        After you complete all of these tasks, attempt the communication process again. If you are successful, then no further action is required. If you are unsuccessful, proceed to step 4.
4. Ensure that the DATALINKS database configuration parameter is set to YES on the DB2 host server where the problem is occurring.

   Log on to the DB2 host server, and issue the command **db2 get dbm cfg**.
   - If the configuration parameter is set to YES, proceed to step 5.
   - If the configuration parameter is set to NO, correct it by issuing the following command: **db2 update dbm cfg using DATALINKS YES**. Then, attempt the communications process again. If you are successful, then no further action is required. If you are unsuccessful, proceed to step 5.
5. Verify that the versions of the DB2 software that are installed on the Data Links server and the DB2 host server are compatible at both the release *and* fixpack level.
   a. Log on to each server machine where the problem is occurring and issue the command **db2level**.
   b. Compare the results with the documented list of compatible DB2 and DB2 Data Links Manager releases and levels (see the Related concepts section at the end of this topic).
      - If the results indicate compatibility, proceed to step 6.
      - If the results indicate incompatibility, you must upgrade the product on at least one of the server machines.

        After you upgrade the product, attempt the communication process again. If you are successful, then no further action is required. If you are unsuccessful, proceed to step 6.
6. Ensure that the DB2COMM registry variable on the DB2 host server includes the value `TCPIP`.

   Log on to the DB2 host server, and issue the command **db2set DB2COMM**.
   - If the `TCPIP` value is included in the registry variable, proceed to step 7.
   - If the `TCPIP` value is not included in the registry variable, use the command **db2set DB2COMM** again to specify the `TCPIP` value. Then, attempt the communications process again. If you are successful, then no further action is required. If you are unsuccessful, proceed to step 7.
7. Verify that TCP/IP connectivity is properly established between the DB2 host server machine and the Data Links server machine.

From the DB2 host server, issue the commands **ipconfig** *dlfm_hostname* and **ping** *dlfm_hostname*, where <*dlfm_hostname*> is the name of the Data Links server machine.

From the Data Links server, issue the commands **ipconfig** *db2_hostname* and **ping** *db2_hostname*, where *db2_hostname* is the name of the DB2 host server machine.

- If the command output indicates that each machine can be reached, proceed to step 8.
- If the command output indicates that one or both machines cannot be reached, verify the following settings for each machine:
  - Your IP Domain Name Server (DNS) contains the correct entry for the machine.
  - If present, the host names are properly listed in the server machine hosts file (/etc/hosts on UNIX systems, c:\winnt\system32\drivers\etc\hosts on Windows).

  After you complete the verification and make any necessary corrections, attempt the communication process again. If you are successful, then no further action is required. If you are unsuccessful, proceed to step 8.

8. If the problem persists, report it to IBM Service.

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 reference:

- "ADD DATALINKS MANAGER Command" in the *Command Reference*
- "DROP DATALINKS MANAGER Command" in the *Command Reference*
- "Communications variables" in the *Administration Guide: Performance*
- "dlfm add_db command" on page 209
- "dlfm drop_db command" on page 213

Data Links File System problems

This section describes common problems that can occur in a Data Links File System (DLFS) on a Data Links server.

Files can be written to but not read in a Data Links File System

Typical symptoms: One or more of the following conditions exist.

- The db2diag.log file on the DB2 host server does not record errors.
- The db2diag.log file on the Data Links server contains messages such as ″Dest not valid for upcall″ and ″Expired or Invalid token.″
- Application errors on reading a file persist even though the DL_EXPINT database configuration parameter on the DB2 host server is set to 600 seconds. The read errors occur within the 600 seconds.

Potential solution: Ensure that the system times on the Data Links server and the DB2 host server are synchronized. For example, a one-hour difference in times between the two servers will cause the token to expire immediately, regardless of the DL_EXPINT database configuration setting.

If the problem persists, report it to IBM Service.

Related concepts:

• "Synchronizing system clocks" on page 64

Data Links File System mount error (Solaris operating environment)

Typical symptoms: You attempt to mount a Data Links File System (DLFS) with the following command: **/opt/IBM/db2/V8.1/instance/dlfmfsmd** */dlfs*, where */dlfs* is the mount point of the system to be mounted.

The following error message is returned:

```
dlfs mount Error: Invalid argument umount :warning:/dlfs not in mnttab
Explanation: An attempt to mount the specified file system has failed.
User Response: Verify that the file system is defined.
Correct any errors from the mount command and try again.
DB1035E Failed to mount file system /dlfs
```

Potential solution: Ensure that the Data Links server is booted in 32-bit kernel mode, not 64-bit kernel mode. Use the command **isainfo -v** to display the mode in which the server was booted.

If the Data Links server is booted in 64-bit kernel mode, the **isainfo -v** command output displays both the architectures `32-bit sparc` and `64-bit sparcv9`.

If the problem persists, report it to IBM Service.

Cannot mount the Data Links File System (UNIX)

The following conditions must exist for a DLFS mount to succeed:

- The dlfsdrv device driver must be loaded.
- The file system must be designated as a DLFS in the appropriate operating system file (/etc/filesystems for AIX, or /etc/vfstab for the Solaris Operating Environment).

A mount will fail if both of these conditions do not exist.

Typical symptoms: One of the following conditions exists.

- The mount command does not show the file system mounted.
- The mount command shows the file system mounted, but not as a DLFS.
- The **mount -v dlfs** command (AIX) or the **mount -F dlfs** command (Solaris Operating Environment) fails with one of these messages:

  ```
  dlfs mount Error: Function not implemented
  dlfs mount helper: Mount Unsuccessful
  Unmount the base file system
  ```

  ```
  dlfs mount helper: Error in getting basefstype
  dlfs mount helper: No base file system specified
  ```

Potential solutions: Perform the following steps to solve the problem.

1. Log on to the Data Links server where the problem is occurring as ″root.″
2. Ensure that the dlfsdrv device driver is loaded.
   a. On AIX, enter the following command: **strload -qf /usr/opt/db2_08_01/cfg/dlfs_cfg**.
      - If the device driver is loaded, the command returns the following output: `/usr/opt/db2_08_01/bin/dlfsdrv:yes`. Proceed to the next step.
      - If the device driver is not loaded, the command returns the following output: `/usr/opt/db2_08_01/bin/dlfsdrv:no`.

        Load the DLFF device driver using the following command: **strload -f /usr/opt/db2_08_01/cfg/dlfs_cfg**, then attempt to mount the DLFS again. If the mount attempt is successful, no further action is required. If the mount attempt is unsuccessful, proceed to the next step.
   b. On the Solaris Operating Environment, enter the following command: **/usr/sbin/modinfo | grep dlfs**.
      - If the dlfsdrv device driver is loaded, the command returns the following output:

        ```
        94 60f68000  1ab7e 155   1  dlfsdrv (dlfsdrv)
        94 60f68000  1ab7e  18   1  dlfsdrv (dlfs)
        ```

        Proceed to step 3.
      - If the dlfsdrv device driver is not loaded, the command does not return any output.

        Load the dlfsdrv device driver using the following command: **add_drv -m ’* 0777 dlfm dlfmgrp’ dlfsdrv**. In this example, it is assumed that the Data Links Manager Administrator ID dlfm is created on the machine in the primary group dlfmgrp.

        Attempt to mount the DLFS again. If the mount attempt is successful, no further action is required. If the mount attempt is unsuccessful, proceed to step 3.
3. Verify that the correct settings for the file system are stored in the appropriate operating system file (/etc/filesystems for AIX, or /etc/vfstab for the Solaris Operating Environment).
   a. On AIX, verify that the following settings exist in the /etc/filesystems file:

      ```
      vfs =dlfs
      nodename =- //ensure that there are no trailing spaces after the dash
      mount =false
      options =rw,Basefs=jfs
      ```

      - If these options are set correctly, proceed to step 4 on page 202.
      - If the file system is defined as a Journaled File System (JFS), or does not have these options set correctly, convert (or re-convert) the file system to a DLFS with the correct options by issuing the following command.

        `/usr/opt/db2_08_01/instance/dlfmfsmd` *`file_system_name`*

        file_system_name is the name of the JFS that is to be converted to a DLFS. Make sure that you are logged on as ″root″ when you run this command.

        After you run the command, attempt to mount the DLFS again. If the mount attempt is successful, no further action is required. If the mount attempt is unsuccessful, proceed to step 4 on page 202.
   b. On the Solaris Operating environment, verify that the following settings exist in the /etc/vfstab file:

      ```
      /dev/dsk/c1t2d0s1  /dev/rdsk/c1t2d0s1   /dlfstest dlfs 2  yes -
      ```

      - If these options are set correctly, proceed to step 4 on page 202.

- If the file system is defined as a UNIX File System (UFS), or does not have these options set correctly, convert (or re-convert) the file system to a DLFS with the correct options by issuing the following command.

  ```
  /opt/IBM/db2_08_01/instance/dlfmfsmd file_system_name
  ```

  file_system_name is the name of the UFS that is to be converted to a DLFS. Make sure that you are logged on as "root" when you run this command.

  After you run the command, attempt to mount the DLFS again. If the mount attempt is successful, no further action is required. If the mount attempt is unsuccessful, proceed to step 4.

4. Ensure that the helper programs for a DLFS are identified in the operating system.

   AIX Check the /etc/vfs file for an entry that contains dlfs_fshelper. For example, the following entry contains dlfs_fshelper:

   ```
   dlfs 12 /usr/opt/db2_08_01/bin/dlfs_mnthlp /usr/opt/db2_08_01/bin/dlfs_fshelper
   ```

   - If the helper programs are correctly identified, proceed to step 5.
   - If the helper programs are not correctly identified, modify the /etc/vfs file as necessary.

     After you modify the /etc/vfs file, attempt to mount the DLFS again. If the mount attempt is successful, no further action is required. If the mount attempt is unsuccessful, proceed to step 5.

   Solaris Operating Environment

   Ensure that the /usr/lib/fs/ directory contains a dlfs subdirectory.

   The /usr/lib/fs/dlfs directory must contain the dlfs_fshelper and mount executables.

   - If the /usr/lib/fs/ directory contains a dlfs subdirectory with the correct executables, proceed to step 5.
   - If the /usr/lib/fs/ directory does not contain a dlfs subdirectory, or the dlfs subdirectory does not contain the correct executables, then the DB2 Data Links Manager installation process did not complete correctly. You must re-install the DB2 Data Links Manager product.

     Before re-installing DB2 Data Links Manager, verify that you have adequate space available in the /usr , /opt, and other system file systems. The original installation might have failed due to lack of space. See the list of related topics below for more information about DB2 Data Links Manager space requirements.

     If the /usr/lib/fs/dlfs directory is still missing or empty after you re-install DB2 Data Links Manager, report the problem to IBM Service.

5. If the problem persists, report it to IBM Service.

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*
- "Minimum hardware and software requirements" on page 27

Related tasks:

- "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*

DB2 clients cannot access files in a Data Links File System

Clients and client applications need the following items to access files in a Data Links File System (DLFS):

- Connectivity to the DLFS through a network.
  - On UNIX systems, this connectivity is typically established through an NFS mount. Before an NFS mount can occur, the file system must be configured as shareable by using the **exportfs** command.
  - On Windows systems, this connectivity is typically accomplished through a shared drive. The drive can be configured as shareable through a series of Windows Explorer actions.
- Read, write, or read and write access permissions on the DLFS.

  If files are linked in DATALINK columns that were defined with the READ PERMISSION DB or WRITE PERMISSION ADMIN attributes, clients need valid access tokens to read or write to files.

  If files are linked in DATALINK columns that were defined with the READ PERMISSION FS or WRITE PERMISSION FS attributes, the file system permissions on directories and files must be set accordingly.

Potential solutions: Perform the following steps to solve the problem.

1. Ensure that the DLFS is properly connected on the client.

   UNIX Ensure that the DLFS is properly mounted. A *stale mount* situation might be occurring when a client attempts to establish connectivity through an NFS mount. A stale mount can occur when a file system is unmounted and remounted on the Data Links server.

   If you suspect a stale mount situation is causing the problem, perform the following steps:

   a. Unmount the file system on the client.
   b. Validate that the file system is mounted on the Data Links server.
   c. Export the file system on the Data Links server using the **exportfs** command.
   d. Remount the file system on the client.
   e. Have the client attempt to access the DLFS. If the attempt is successful, no further action is required. If the attempt is unsuccessful, proceed to step 2 on page 204.

   Windows

   Ensure that the DLFS is properly mapped (connected) on the client. A network problem might have interfered with your connection to the DLFS shared network drive. If you suspect that a network problem occurred, perform the following steps:

   a. Disconnect the network drive on the client.
   b. Validate that the DLFS drive is marked shareable on the Data Links server.
   c. Validate that the drive is under control of the DLFF by using the **dlff list** command.

d. Connect (map) the network drive on the client.
e. Have the client attempt to access the DLFS. If the attempt is successful, no further action is required. If the attempt is unsuccessful, proceed to step 2.

2. Check the access permissions of the DLFS on the Data Links server and on the client.

   The DLFS can be made available to clients with read-only or read-write access. If the client needs write access, the DLFS must be exported (UNIX) or shared (Windows) with access controls set accordingly for specific clients.

   - Reset the access controls if necessary, then have the client attempt to access the DLFS. If the attempt is successful, no further action is required. If the attempt is unsuccessful, proceed to step 3.
   - If you do not need to reset access controls, proceed to step 3.

3. Determine whether read or write access tokens generated by a DB2 database are required to access files. If an access token is needed to perform read, write, or both read and write actions, ensure that the client or client application is using the appropriate access token.

   If an application program is having a problem accessing a file, you might need to run a DB2 event monitor to trace the SQL statement activity of the application.

   a. Check the READ PERMISSION and WRITE PERMISSION attributes settings of the DATALINK columns where the files are referenced.

      You can query tables directly for column information using the DB2 Command Line Processor. See the System catalog views topic and the SYSCAT.COLUMNS catalog view topic for more information about querying tables.

      If a column uses the READ PERMISSION DB attribute, the WRITE PERMISSION ADMIN attribute, or both attributes, then a read or write access token generated by the DB2 database is required to perform the corresponding action on the files referenced in the DATALINK column.

      If the DATALINK column attributes do not specify that access tokens are required, proceed to step 4. If the DATALINK column attributes specify that access tokens are required, continue with this step.

   b. If the DATALINK column attributes require access tokens for referenced files, ensure that both of the following conditions are met:

      - The client or client application is using the correct Data Links Manager scalar function to retrieve the URL *with* the required access token. For example, when a client application uses the DLURLPATHONLY or DLURLCOMPLETEONLY function to extract the path name and file name from the DATALINK value, DB2 does not return an access token.
      - The client or client application was granted the authority to request a write token to update the file from the Data Links server where the file is stored. This authority is granted by using the **dlfm grant** command.

      After you ensure that both of these conditions are met, have the client attempt to access the DLFS. If the attempt is successful, no further action is required. If the attempt is unsuccessful, proceed to step 4.

4. Check the DL_EXPINT and DL_WT_IEXPINT database configuration parameters on the DB2 host server. These parameters determine the length of time, in seconds, for which a generated read or write access token is valid.

If a client or client application does not use the token within that period of time, DB2 Data Links Manager rejects the token, and the client or client application will not be allowed to read or write to the file.

Also, ensure that the system times on the Data Links server and the DB2 host server are synchronized. For example, a one hour difference in times between the two servers will cause the tokens to expire immediately, regardless of the database configuration settings. For more information about synchronizing system times, see the Related concepts section, below.

5. If the problem persists, report it to IBM Service.

Related concepts:

- "Event monitors" in the *System Monitor Guide and Reference*
- "Synchronizing system clocks" on page 64
- "Advanced file management security features" on page 98
- "Approach 3: Update linked files/customize file access privileges" on page 113

Related tasks:

- "Creating an event monitor" in the *System Monitor Guide and Reference*

Related reference:

- "dl_expint - Data Links access token expiry interval configuration parameter" in the *Administration Guide: Performance*
- "SYSCAT.COLUMNS catalog view" in the *SQL Reference, Volume 1*
- "dl_wt_iexpint - Data Links write token initial expiry interval configuration parameter" in the *Administration Guide: Performance*
- "System catalog views" in the *SQL Reference, Volume 1*
- "dlff list command (Windows operating system)" on page 235
- "dlfm grant command" on page 214

DB2 host database or Data Links File Manager hang situations

An example of a hang situation is when you attempt to connect to a database, but nothing gets returned.

Troubleshooting hang situations on a DB2 host server or on a Data Links server can be a complicated process that usually requires the intervention of IBM Service.

IBM Service will direct you to use a combination of DB2 debugging tools and operating system commands to provide detailed information about the hang situation. You will be instructed on how to use the specific tools and commands as required for your situation.

For an overview of the debugging tools and commands, see the topics listed below.

Related concepts:

- "Diagnostic tools for UNIX-based systems" in the *Troubleshooting Guide*
- "DB2 traces" on page 190

Related reference:

- "Troubleshooting commands (UNIX)" in the *Troubleshooting Guide*

- "Diagnostic tools for Windows NT, Windows 2000 and Windows XP operating systems" in the *Troubleshooting Guide*
- "Required diagnostic information for problem analysis" on page 188

DB2 host database or Data Links File Manager failure situations

A *failure* (or a *crash*) is a severe error or condition that causes a program, application, or process to abnormally terminate. An example of a failure is a connect request from a client to a server that does not return any result.

In a DB2 Data Links Manager environment, a failure can occur that causes the DB2 database manager on a DB2 host server to abnormally terminate, the Data Links File Manager (DLFM) on a Data Links server to abnormally terminate, or both components on both server machines to abnormally terminate.

- If a failure occurs on a Data Links server, try to diagnose and correct the failure yourself. Collect the basic set of diagnostic information as described in the Required diagnostic information for problem analysis topic. Then, refer to theData Links server machine failure topic and the DB2 Data Links Manager recovery scenarios topic for additional troubleshooting assistance.

  If you still cannot diagnose and correct the problem, report it to IBM Service.
- Failures that occur on a DB2 host usually require the assistance of IBM Service to troubleshoot and correct. Collect the basic set of diagnostic information as described in the Required diagnostic information for problem analysis topic. You must also collect the diagnostics files located in the DIAGPATH directory on the DB2 host.

If a problem is severe, IBM Service might direct you to perform a DB2 trace, or to collect additional information, such as operating system logs.

Related concepts:

- "DB2 Data Links Manager recovery scenarios" on page 182
- "DB2 traces" on page 190

Related reference:

- "Data Links server machine failure" on page 175
- "Required diagnostic information for problem analysis" on page 188

Considerations for dropping a DB2 Data Links Manager from a database

When you want to drop (remove) a DB2® Data Links Manager from a database, you must remove information from both the DB2 host server *and* from the corresponding Data Links server that is being dropped.

The process of dropping a DB2 Data Links Manager from a DB2 database is not difficult, but it is a major change to your database environment. You *must* follow specific procedures that to ensure that the operation goes smoothly.

For detailed instructions and procedures on dropping a DB2 Data Links Manager from a database, see the topics listed below.

Related concepts:

- "Dropping Data Links Manager from a specific DB2 database on the DB2 host" on page 58

Related tasks:

- "Removing information about a DB2 database from Data Links Manager" on page 78

Creating and dropping the DB2 database on the Data Links server

This section describes how to create the DLFM's own database, named DLFM_DB by default, and how to drop the DLFM_DB database.

The DLFM_DB database is used to track files that are stored on a Data Links server and linked to a remote DB2 host server.

You need to create the DLFM_DB database only if the DB2 Data Links Manager installation program failed and did not create the database. Generally, you should *not* interact with this database directly.

Attention:

- Do not drop the DLFM_DB database if it has been used in an active DB2 Data Links Manager environment that you intend to continue using. Loss of the DLFM_DB database can result in an unrecoverable DB2 Data Links Manager environment.
- If you need to drop the DLFM_DB database and it contains *any* data, contact IBM Service before dropping it. IBM Service will help you to determine whether you can drop the DLFM_DB and continue to use your DB2 Data Links Manager environment, or if there are actions that you must take before dropping the DLFM_DB to preserve your environment.

Prerequisites:

You must have DB2 Data Links Manager Administrator authority.

Procedure:

To create the DB2 database on the Data Links server:

1. Log on to the system as the DB2 Data Links Manager Administrator.
2. Enter the **dlfm setup** command. This command starts the DB2 database manager, creates the DLFM_DB database and tables, and then stops the DB2 database manager.

To drop the DB2 database on the Data Links server:

1. Log on to the system as the DB2 Data Links Manager Administrator.
2. Enter the **dlfm drop_db** command to drop the DLFM_DB database.

Related reference:

- "dlfm drop_db command" on page 213
- "dlfm setup command" on page 229

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.

Displays the syntax of all Data Links File Manager commands that apply on the current platform and environment.

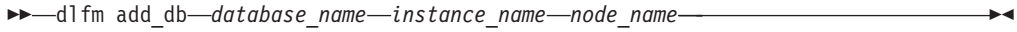
To display command descriptions as well as command syntax, use the **dlfm ?** or the **dlfm help** commands.

Related reference:

- "dlfm ? command" on page 231
- "dlfm help command" on page 218

dlfm add_db command

Requirement: You must execute this command from the Data Links server using the Data Links Manager Administrator user ID.



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 70
- "Registering a drive with DLFF (Windows operating systems)" on page 72

Related reference:

- "dlfm list registered prefixes command" on page 219

dlfm add_prefix command

Requirement: You must execute this command from the Data Links server using the Data Links Administrator user ID.

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 share name (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 70
- "Registering a drive with DLFF (Windows operating systems)" on page 72
- "Listing registered file systems (AIX, Solaris Operating Environment)" on page 73
- "Listing registered drives (Windows operating systems)" on page 73

dlfm bind command

Requirement: You must execute this command from the Data Links server using the Data Links Manager Administrator user ID.

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**.

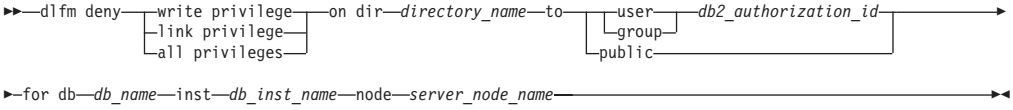
Related concepts:

- "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.



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 16 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 16. DATALINK column attributes and applicable DLM security privileges

Possible column attribute	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 99

Related reference:

- "dlfm grant command" on page 214
- "dlfm revoke command" on page 225
- "dlfm set link security command" on page 228

dlfm drop_db command

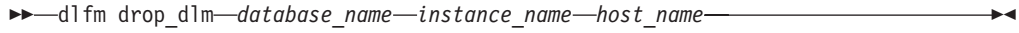
Requirement: You must execute this command from the Data Links server using the Data Links Manager Administrator user ID.

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.



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.

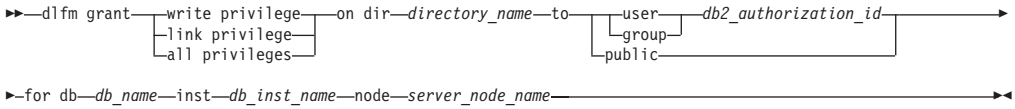
Example:
- `dlfm drop_dlm sampledatabase myinstance myhostname.com`

Related tasks:
- "Removing information about a DB2 database from Data Links Manager" on page 78

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.



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 17 on page 215 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 17. DATALINK column attributes and applicable DLM security privileges

Possible column attribute	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 99

Related tasks:

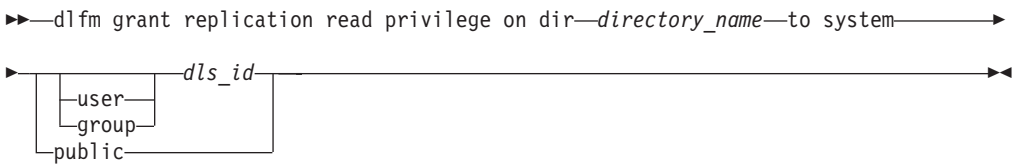
- "Approach 3 configuration tasks" on page 115

Related reference:

- "dlfm deny command" on page 211
- "dlfm revoke command" on page 225
- "dlfm set link security command" on page 228

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 SQL replication" in the *IBM DB2 Information Integrator SQL Replication Guide and Reference*

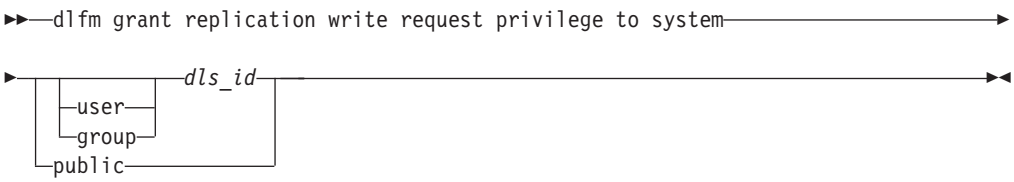
Related reference:

- "dlfm grant replication write request command" on page 217

- "dlfm list registered replication access control command" on page 220
- "dlfm revoke replication command" on page 226

dlfm grant replication write request command

Requirement: You must execute this command from the Data Links server using the Data Links Manager Administrator user ID.



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 DataPropagator 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 SQL replication" in the *IBM DB2 Information Integrator SQL Replication Guide and Reference*

Related reference:
- "dlfm grant replication read command" on page 216
- "dlfm list registered replication access control command" on page 220
- "dlfm revoke replication command" on page 226

dlfm help command

Requirement: You must execute this command from the Data Links server using the Data Links Manager Administrator user ID.

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 209
- "dlfm ? command" on page 231

dlfm list registered databases command

Requirement: You must execute this command from the Data Links server using the Data Links Manager Administrator user ID.

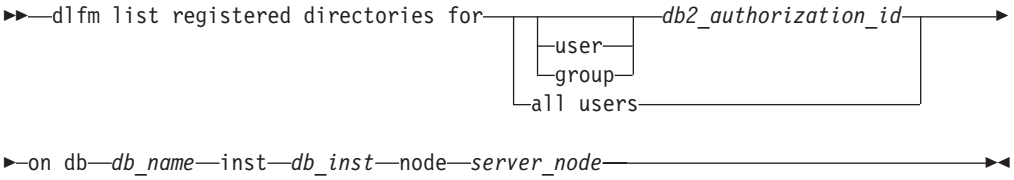
Lists all the databases registered with the Data Links File Manager.

Related tasks:

- "Listing databases registered with the Data Links server" on page 78

dlfm list registered directories command

Requirement: You must execute this command from the Data Links server using the Data Links Manager Administrator user ID.



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 99

Related tasks:

- "Approach 3 maintenance tasks" on page 116

Related reference:

- "dlfm grant command" on page 214
- "dlfm set link security command" on page 228

dlfm list registered prefixes command

Requirement: You must execute this command from the Data Links server using the Data Links Manager Administrator user ID.

Lists all prefixes within a Data Links File System (DLFS) that are registered with the Data Links File Manager.

Related tasks:

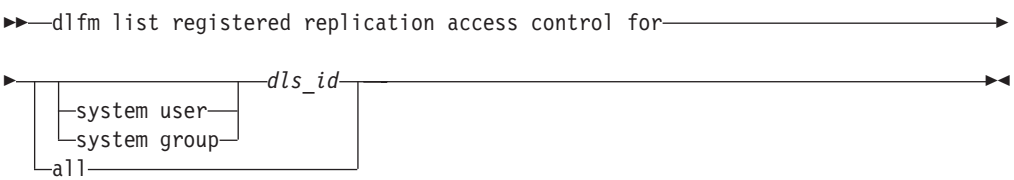
- "Listing registered file systems (AIX, Solaris Operating Environment)" on page 73
- "Listing registered drives (Windows operating systems)" on page 73

Related reference:

- "dlfm add_prefix command" on page 210

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.



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:

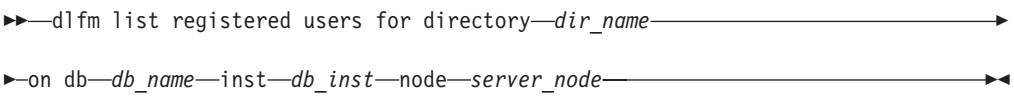
- "Replicating special data types in SQL replication" in the *IBM DB2 Information Integrator SQL Replication Guide and Reference*

Related reference:

- "dlfm grant replication read command" on page 216
- "dlfm grant replication write request command" on page 217

dlfm list registered users command

Requirement: You must execute this command from the Data Links server using the Data Links Manager Administrator user ID.



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 99

Related tasks:

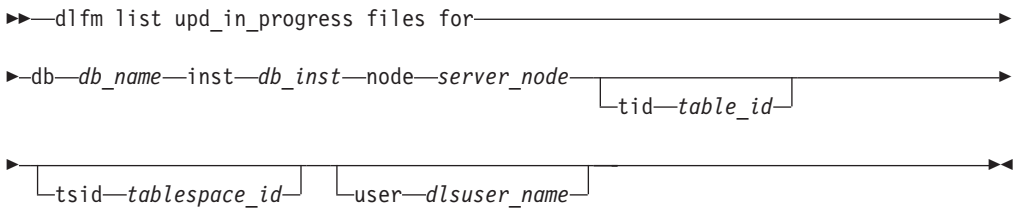
- "Approach 3 maintenance tasks" on page 116

Related reference:

- "dlfm grant command" on page 214
- "dlfm set link security command" on page 228

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.



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`

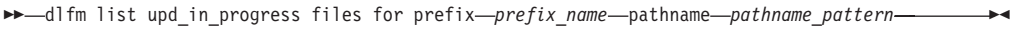
Related concepts:
- "Advanced file management security rules and guidelines" on page 99
- "Approach 3: Update linked files/customize file access privileges" on page 113

Related tasks:
- "Approach 3 maintenance tasks" on page 116
- "Recovering missing write tokens" on page 123

Related reference:
- "dlfm grant command" on page 214
- "dlfm refresh key command" on page 223

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.



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 share name (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 99
- "Approach 3: Update linked files/customize file access privileges" on page 113

Related tasks:

- "Approach 3 maintenance tasks" on page 116
- "Recovering missing write tokens" on page 123

Related reference:

- "dlfm grant command" on page 214

dlfm refresh key command

Requirement: You must execute this command from the Data Links server using the Data Links Administrator user ID.

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:

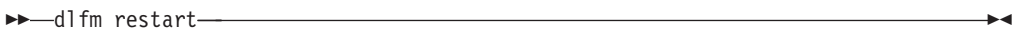
- "Approach 3: Update linked files/customize file access privileges" on page 113
- "Usage considerations for write tokens" on page 122

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.



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 67

Related tasks:

- "Restarting the DLFM after an abnormal termination" on page 68

Related reference:

- "dlfm start command" on page 230
- "dlfm stop command" on page 230

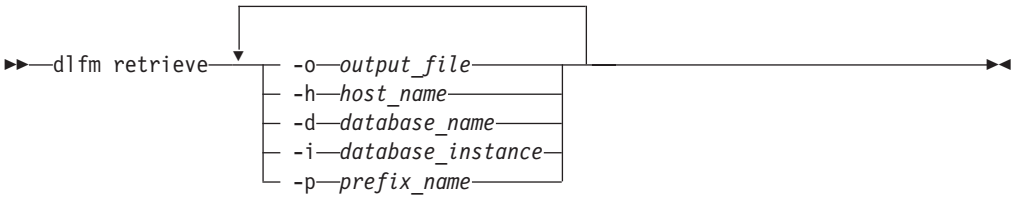
dlfm retrieve command

Requirement: You must execute this command from the Data Links server using the Data Links Manager Administrator user ID.

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
```

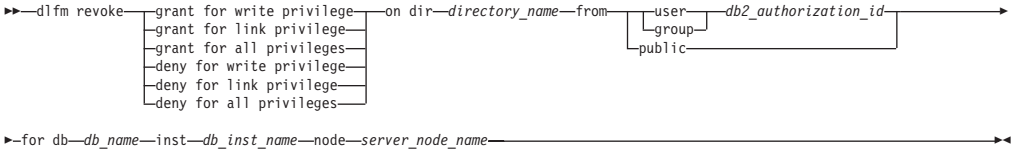
Related tasks:

- "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:

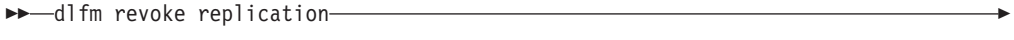
- "Advanced file management security rules and guidelines" on page 99

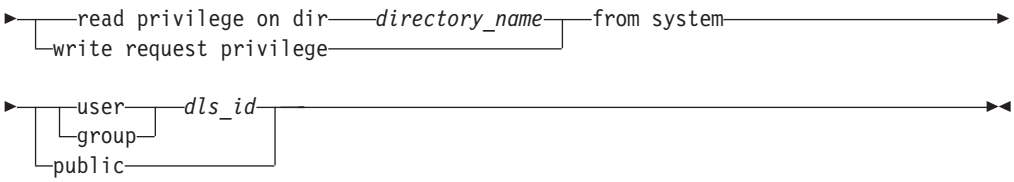
Related reference:

- "dlfm deny command" on page 211
- "dlfm grant command" on page 214
- "dlfm set link security command" on page 228

dlfm revoke replication command

Requirement: You must execute this command from the Data Links server using the Data Links Server Administrator user ID.





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 SQL replication" in the *IBM DB2 Information Integrator SQL Replication Guide and Reference*

Related reference:

- "dlfm grant replication read command" on page 216
- "dlfm grant replication write request command" on page 217
- "dlfm list registered replication access control command" on page 220

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.

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 69

dlfm set link security command

Requirement: You must execute this command from the Data Links server using the Data Links Manager Administrator ID.

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 99
- "Approach 3: Update linked files/customize file access privileges" on page 113

Related reference:

- "dlfm deny command" on page 211
- "dlfm drop_dlm command" on page 213
- "dlfm grant command" on page 214
- "dlfm revoke command" on page 225

dlfm setup command

Requirement: You must execute this command from the Data Links server using the Data Links Manager Administrator ID.

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 211
- "dlfm create_db command" on page 211
- "dlfm stopdbm command" on page 231
- "dlfm startdbm command" on page 230

dlfm shutdown command

Requirement: You must execute this command from the Data Links server using the Data Links Manager Administrator user ID.

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 67

Related tasks:

- "Restarting the DLFM after an abnormal termination" on page 68

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 67

Related reference:

- "dlfm see command (AIX and Solaris Operating Environment)" on page 228

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 67

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 67

Related reference:

- "dlfm see command (AIX and Solaris Operating Environment)" on page 228
- "dlfm shutdown command" on page 229

dlfm stopdbm command

Requirement: You must execute this command from the Data Links server using the Data Links Manager Administrator user ID.

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 67

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.

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 209
- "dlfm help command" on page 218

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.

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 72

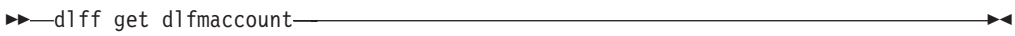
Related reference:

- "dlfm add_prefix command" on page 210
- "dlff list command (Windows operating system)" on page 235

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.



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 238

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.

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 82

Related reference:
- "dlff set loglevel command (Windows operating system)" on page 239

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.

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 73

Related reference:
- "dlff add command (Windows operating system)" on page 233
- "dlfm list registered prefixes command" on page 219

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.

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 82

Related reference:
- "dlff get loglevel command (Windows operating system)" on page 234
- "dlff set loglevel command (Windows operating system)" on page 239

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 72
- "Registering, querying and de-registering a DLFF (Windows operating system)" on page 75

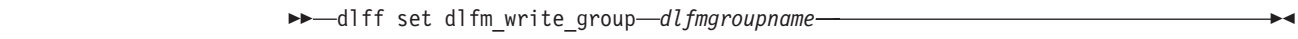
Related reference:

- "dlff list command (Windows operating system)" on page 235

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.



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 dlfmxgrp`

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*

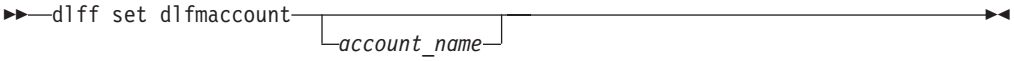
Related reference:

- "dlff refreshtrace command (Windows operating system)" on page 235
- "dlff set dlfmaccount command (Windows operating system)" on page 238

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.



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 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 235
- "dlff set dlfm_write_group command (Windows operating system)" on page 237

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 82

Related reference:

- "dlff get loglevel command (Windows operating system)" on page 234
- "dlff refreshtrace command (Windows operating system)" on page 235

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)

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

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

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" on page 244
- "Opening a file with an expired token" on page 244
- "Opening a file with an invalid token" on page 245

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 18 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 18 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 18. 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
cat filename	cat: 0652-050 Cannot open filename.
cp filename newfilename	cp: filename: The file access permissions do not allow the specified action.
diff filename filenamefs	diff: filename: The file access permissions do not allow the specified action.
grep "str" filename	grep: 0652-033 Cannot open filename.
head filename	filename: The file access permissions do not allow the specified action.
tail filename	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 19 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 19 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 19. 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
cat "04E2_DEGnck__JD8hFHOy6JU;filename"	cat: 0652-050 Cannot open 04E2_DEGnck__JD8hFHOy6JU;filename.
cp "04E2_DEGnck__JD8hFHOy6JU;filename" newfilename	cp: 04E2_DEGnck__JD8hFHOy6JU;filename: The file access permissions do not allow the specified action.

Table 19. 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
diff "04E2_DEGnck__JD8hFHOy6JU;filename" filenamefs	diff: 04E2_DEGnck__JD8hFHOy6JU;filename: The file access permissions do not allow the specified action.
grep "str" "04E2_DEGnck__JD8hFHOy6JU;filename"	grep: 0652-033 Cannot open 04E2_DEGnck__JD8hFHOy6JU;filename.
head "04E2_DEGnck__JD8hFHOy6JU;filename"	04E2_DEGnck__JD8hFHOy6JU;filename: The file access permissions do not allow the specified action.
more "04E2_DEGnck__JD8hFHOy6JU;filename"	04E2_DEGnck__JD8hFHOy6JU;filename: The file access permissions do not allow the specified action.
tail "04E2_DEGnck__JD8hFHOy6JU;filename"	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 20 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 20 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 20. 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_AAAAAAAAAAAAAHOy6JU;filename"	cat: 0652-050 Cannot open 04E2_AAAAAAAAAAAAAHOy6JU;filename.
cp "04E2_AAAAAAAAAAAAAHOy6JU;filename" newfilename	cp: 04E2_AAAAAAAAAAAAAHOy6JU;filename: A file or directory in the path name does not exist.
diff "04E2_AAAAAAAAAAAAAHOy6JU;filename" filenamefs	diff: 04E2_AAAAAAAAAAAAAHOy6JU;filename: A file or directory in the path name does not exist.
grep "str" "04E2_AAAAAAAAAAAAAHOy6JU;filename"	grep: 0652-033 Cannot open 04E2_AAAAAAAAAAAAAHOy6JU;filename.
head "04E2_AAAAAAAAAAAAAHOy6JU;filename"	04E2_AAAAAAAAAAAAAHOy6JU;filename: A file or directory in the path name does not exist.
more "04E2_AAAAAAAAAAAAAHOy6JU;filename"	04E2_AAAAAAAAAAAAAHOy6JU;filename: A file or directory in the path name does not exist.

Table 20. Partial list of AIX commands that you might use to open a file with an invalid token and the associated error message. (continued)

Command name	Error message
tail "04E2_AAAAAAAAAAAAAHOy6JU;filename"	04E2_AAAAAAAAAAAAAHOy6JU;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 255

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 249

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" on page 247
- "ls filename command" on page 247
- "mkdir command" on page 247
- "mount command" on page 248
- "mvdir dir11 newdir11" on page 248
- "mv filename newfilename command" on page 248

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.

mvdir 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 250
- "Opening a file with an invalid token" on page 250

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 21 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 21 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 21. Partial list of commands that you might use in the Solairs Operating Environment to open a file without a token and the associated error message.

Command name	Error message
cat filename	cat: cannot open filename
cp filename newfilename	cp: cannot access filename
diff filename filenamefs	diff: filename: Permission denied
grep "str" filename	grep: can't open filename
head filename	filename: Permission denied
more filename	filename: Permission denied
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 22 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 22 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 22. 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 23 on page 251 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 23 on page 251 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 23. 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 255

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 253

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″"
- "mkdir command"
- "move command"
- "ren ″d:\test\filename″ ″newfilename″"
- "rmdir or delete command from Windows Explorer"

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:\test\filename″ ″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 254
- "Opening a file with an invalid token" on page 254

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 24 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 24 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 24. 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
more <″d:\test\filename″	Access is denied.
type ″filename″	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 25 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 25 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 25. 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
more ″d:\test\04E2_D3iwQk__H5G1CXMzq_g;filename″	The system cannot find the file specified.
type ″04E2_D3iwQk__H5G1CXMzq_g;filename″	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 26 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 26 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 26. 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
more ″d:\test\24E2_D3iwQk__H5G1CXMzq_g;filename″	The system cannot find the file specified.
type ″24E2_D3iwQk__H5G1CXMzq_g;filename″	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 27. 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 28. Possible causes and suggested actions for errors that result from opening a file for write 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 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 **dlfm list upd_in_progress files** 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 241
- "DLFS errors from commands (Solaris Operating Environment)" on page 246

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 Manager (DLFM) server. The core component on a Data Links server machine that provides a variety of functions in support of the other components in a Data Links Manager environment. An active DLFM server supports: all Data Links Manager requests coming from registered DB2 databases, requests from the DLFF component to validate file references, and requests from DB2 Replication to access linked files to be replicated. The DLFM server also manages the contents of its private DLFM_DB 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 host. In a DB2 Data Links Manager configuration, a DB2 database that contains a DATALINK column, on a DB2 server.

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.

DLFM server. See Data Links File Manager server.

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.

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 share name (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.

Licensees 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
AISPO
AIX
AIXwindows
AnyNet
APPN
AS/400
BookManager
C Set++
C/370
CICS
Database 2
DataHub
DataJoiner
DataPropagator
DataRefresher
DB2
DB2 Connect
DB2 Extenders
DB2 OLAP Server
DB2 Information Integrator
DB2 Query Patroller
DB2 Universal Database
Distributed Relational Database Architecture
DRDA
eServer
Extended Services
FFST
First Failure Support Technology
IBM
IMS
IMS/ESA
iSeries
LAN Distance
MVS
MVS/ESA
MVS/XA
Net.Data
NetView
OS/390
OS/400
PowerPC
pSeries
QBIC
QMF
RACF
RISC System/6000
RS/6000
S/370
SP
SQL/400
SQL/DS
System/370
System/390
SystemView
Tivoli
VisualAge
VM/ESA
VSE/ESA
VTAM
WebExplorer
WebSphere
WIN-OS/2
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

E

F

H

I

J

L

M

N

O

P

R

S

T

U

V

W

X

Z

Contacting IBM

Before you contact DB2 customer support, check the *Troubleshooting Guide* for help with your specific technical problem. If you still need help, follow the procedures for gathering necessary information before you contact DB2 customer support.

For information or to order any of the DB2 Universal Database products, contact an IBM representative at a local branch office or contact any authorized IBM software remarketer.

If you live in the U.S.A., then you can call one of the following numbers:

- 1-800-237-5511 for customer support
- 1-888-426-4343 to learn about available service options

Product information

DB2 Universal Database product information is available by telephone or by the World Wide Web at
http://www.ibm.com/software/data/db2/udb/winos2unix/support

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 http://www.ibm.com/planetwide

In some countries, IBM-authorized dealers should contact their dealer support structure for information.

Printed in USA

SC27-1221-02

Spine information:

IBM® DB2® Universal Database™

Data Links Manager Administration Guide and Reference

Version 8.2