DATALINKS:

MANAGING EXTERNAL DATA WITH DB2 UNIVERSAL DATABASE

Prepared for IBM Corporation by
Judith R. Davis
February, 1999

TABLE OF CONTENTS

Executive Summary 1

Overview of DataLinks 2
The Business Problem 2
Extending the Data Management Environment 3
What is DataLinks? 4

Business Applications for DataLinks 5
Introduction 5
Web-Based Content Management 6
CAD/CAM Application Software 7
Other Applications 10

How DataLinks Works 11
DataLinks Components 11
The DATALINK Data Type 12
The Data Links File Manager 14
The Data Links Filesystem Filter 16

The Benefits of DataLinks and Alternatives 17
Benefits 17
Alternatives 17
Availability 18

Conclusions 18

EXECUTIVE SUMMARY

IBM is focused on extensible data management

IBM has focused research efforts on creating an extensible data-management infrastructure for more than a decade. The results of many of these efforts have already appeared in IBM's DB2 family of relational database management systems (RDBMSs). DB2 Universal Database (UDB) is the company's flagship for the implementation of object-relational extensibility.

DataLinks enables DB2 to manage external data

But an extensible RDBMS is only one necessary component of an extensible data-management environment. IBM has also developed a new technology, called DataLinks, which extends the management umbrella of the RDBMS to data stored in external operating system files. As part of the current release of DB2 UDB, DataLinks gives DB2 comprehensive control over external data in the following areas:

- Referential integrity
- Access control
- Coordinated backup and recovery
- Transaction consistency

DataLinks provides the vital integration between the RDBMS and the file system through extensions to DB2 UDB, including a new DB2 DATALINK data type, and a layer of coordinating software called DB2 Data Links Manager that runs on the file servers where the RDBMS-managed external data are stored.

Benefits: taking advantage of both the RDBMS and the file system

In summary, DataLinks meets a very challenging application requirement that has existed for many years. DataLinks enables organizations to continue storing data (particularly large files of unstructured or semi-structured data such as documents, images, video clips, and engineering drawings) in the file system to take advantage of file-system capabilities, while at the same time coordinating the management of these files and their contents with associated data stored in an RDBMS. In addition to the advantages inherent in RDBMS management of external data, DataLinks has other significant benefits:

- Allows external files to be stored in close proximity to the application to reduce network traffic and maximize application performance.

- Maintains speed of file access by continuing to use the file system for this, not the DBMS. This is also critical to application performance.

- Requires minimal or no changes to existing applications.

- Works with any file system on Unix or Windows NT, and takes advantage of support for hierarchical storage managers within the file system.

DataLinks has broad appeal

DataLinks is innovative software that has broad appeal across a variety of application segments. Any application with significant content-management requirements can benefit from DataLinks, from Web-based electronic-commerce and intranet applications to more traditional computer-aided design and manufacturing (CAD/CAM) applications.

OVERVIEW OF DATALINKS

THE BUSINESS PROBLEM

Users want integrated content management…

Over the past decade, the relational database management system (RDBMS) has become indispensable for its ability to manage traditional business data with integrity, security, and reliability, and for its flexible data-access capabilities. But until recently, the RDBMS was not an effective manager of unstructured or semi-structured data: documents, images, email messages, presentations, engineering drawings, spreadsheets, video clips, and other business formats. Most of these types of data remain outside the DBMS in a file system (and there are good reasons to keep them there as described in the next section). However, these files are often related in some way to traditional data stored in the RDBMS. Or the files contain structured data that, if stored in an RDBMS along with a reference to the file, could be used for searching and analysis purposes. In these cases, it would be valuable to provide an overall “content-management” system that integrates management of the file and its associated data, synchronizing updates, backup and recovery, and other functions across both the RDBMS and the file system.

…without duplicating RDBMS functionality in custom software

A very simple example is the ability to link a file containing a photograph of a product (as it appears in a catalog) to inventory information on the product stored in a database. If the manufacturer stops selling the product, both the photo and the DBMS information should be deleted or archived. Yet it is difficult to ensure this level of coordination because the photo is managed in the file system and the inventory information is managed by the RDBMS. To solve this data-integrity problem, organizations have been forced to develop and maintain custom software to manage the links between files and data. Unfortunately, this means recreating much of the functionality that the RDBMS already provides for data stored inside a database.

Internet and CAD/CAM applications lead the way

The bottom line is that there are many applications that store data in flat files but need robust management capabilities without having to move all of the data into the DBMS. A primary example is Internet and intranet applications. As the Internet/Web grows dramatically in popularity as an application platform, it has become the driving force for many content-management applications. Organizations want to provide users with access to a wide variety of information in many different formats. As a result, content management is now an important component of electronic commerce (“e-commerce,” or “e-business” to use IBM’s term). Another example is computer-aided design and manufacturing (CAD/CAM) applications.

These applications clearly illustrate the long-term challenge of creating a comprehensive content-management system that can integrate external data and files with traditional business data. Users don’t necessarily want to move everything into the RDBMS; they want an alternative that combines the advantages of existing industrial-strength data-management technology with the advantages of file-management technology. This is the precise set of requirements IBM is targeting with delivery of its new DataLinks technology as part of DB2 Universal Database.

EXTENDING THE DATA MANAGEMENT ENVIRONMENT

Extending relational to manage a broad array of business content

The latest development in RDBMS technology is enabling the database server to manage not just traditional business data but a much broader array of business content. To do this, the RDBMS must be extended to understand new types of data, complex business relationships, and additional application semantics. IBM's DB2 Universal Database (DB2 UDB) is a good example of an extended-relational, or object-relational, DBMS (ORDBMS). DB2 UDB has evolved into a database server that can better represent real-world objects and business content through the implementation of an extensible type system; user-defined functions; an extensible, cost-based optimizer; and object-oriented modeling techniques plus many other new features and functions.

An ORDBMS is a key component of a comprehensive and fully extensible data-management architecture. Other important capabilities include transparent access to heterogeneous data, client support for objects, and an extended file management system that allows the RDBMS to manage external files and data. IBM has already delivered most of these components and will continue to enhance its support for extensibility in its DB2 family of products as described in a paper titled *Creating An Extensible, Object-Relational Data Management Environment: IBM's DB2 Universal Database*. (This and other papers covering IBM's data-management strategy in more detail, such as *IBM's DB2 Spatial Extender: Managing Geo-Spatial Information within the DBMS*, are available on IBM's web site; see page 18.)

Management of external data is a key ORDBMS requirement

As part of its core capabilities, an ORDBMS should offer the flexibility to store large user-defined objects either within the database (as large objects, or LOBs) or outside the database in external file systems. For external data, the ORDBMS should provide the same level of integrity, access control, transaction consistency, and backup and recovery as it does for internal data.

In reality, most of the world's data still lives in file systems and not in a formal DBMS environment. This is particularly true for unstructured and semi-structured data such as images, documents, engineering drawings, email messages, video clips, spreadsheets, presentations, etc. The sheer volume of data in file systems is not only greater than that stored in a DBMS; it also continues to grow at a much faster rate. And this difference in growth rate is increasing with the advent of Internet/Web applications that focus on creating large images, documents, and videos for online display. Although these types of data could be moved into the ORDBMS and stored as LOBs, continuing to store data in external files is useful for many reasons, and may, in fact, be necessary.

For data that must be "streamed" to the client within a specified time period in order to be meaningful, such as video frames or audio, it is best to use a specialized server optimized for delivery of such data. RDBMSs do not know how to deliver isochronous data efficiently.

Another advantage to storing data as external files is application performance. There are two important aspects to this. The first is the ability to maintain the file system's speed of access to its files. The DBMS, with its inherent overhead—parser, optimizer, etc.—will never be able to match the performance of the file system in this area. The second is the ability to locate the data close to the application. This optimizes performance by minimizing network traffic. This is critical in the case of very large files.

A third benefit is continued use of existing applications that directly access operating system files without having to modify any application code, the application programming interfaces (APIs), or the file system. In addition, other file-system advances, such as support for hierarchical storage management, can be exploited.

On the other hand, file systems do not provide the same comprehensive data-management facilities, such as transaction control, enforcement of integrity constraints, and backup and recovery as the ORDBMS. The real-life examples discussed in the section on "Business Applications of DataLinks" clearly illustrate the need for the ORDBMS to manage external files.

IBM's DataLinks technology marries the ORDBMS and the file system

A major challenge, however, is ensuring the integrity of external data and controlling access to the data even though they are physically stored outside the database. To do this, IBM has developed a technology called "DataLinks" that has been incorporated into DB2 UDB version 5.2. DataLinks extends robust ORDBMS functionality to the management of external files and data and enables SQL to provide a single interface for transparent query access to data stored in both the DBMS and external files. The goals are to add integrated search capability to existing applications that already use a file API for data storage and access, and to give SQL-based applications transparent access to external data. DataLinks has the potential to dramatically increase the scope of data that falls under the management of the ORDBMS.

Linking file data with database data through DataLinks is an important component of IBM's comprehensive architecture for creating a fully extensible data-management environment. DataLinks demonstrates IBM's commitment to delivering DBMS products that support its architecture and strategy.

WHAT IS DATALINKS?

Referential integrity, access control, and recoverability for external data

DataLinks is software that enables DB2 UDB to manage data stored in external operating system files as if the data were stored directly in the database. By extending the reach of the DBMS to operating system files, DataLinks gives users flexibility to store data inside or outside the database as appropriate. DataLinks provides several levels of control over external data:

- *Referential integrity*—DataLinks ensures that users cannot delete or rename any external file as long as it is referenced in the database.
- *Access control*—DB2's permissions can also be used to grant or deny a user the ability to read a referenced external file. Read access control is optional.
- *Coordinated backup and recovery*—The DBMS is responsible for backup and recovery of external data in synchronization with the associated database. This type of control over external data is also optional.
- *Transaction consistency*—Changes that affect both the database and external files are executed within a transactional context to preserve the logical integrity and consistency of the data.

IBM's well-respected Almaden Research lab has worked on the development of the DataLinks technology for many years. DataLinks is a good example of IBM's strategy of creating a pipeline of research projects that ultimately become part of the company's ORDBMS family of products. This pipeline started with the 1995 delivery of DB2 Universal Database version 2, which incorporated IBM's Starburst technology for extensibility (see the paper referenced on page 3 for details).

DataLinks will be included in ISO and ANSI SQL3

IBM is also committed to standards and brings its technology to the appropriate standards body before delivering the technology in IBM products. This is IBM's strategy for driving the database industry while getting valuable feedback on technology directions. The International Organization for Standardization (ISO) and the American National Standards Institute (ANSI) have both accepted IBM's proposal to include the DataLinks specification in the SQL3 standard. ANSI and ISO have created a new component of SQL3 called "Management of External Data," which has committee-draft status and has begun its approval process. This will enable vendors other than IBM to implement DataLinks functionality. While the standard specifies the SQL syntax and semantics for DataLinks, it does not define for the vendor how to actually implement DataLinks functionality. This is the more difficult challenge, especially the development of the APIs between the DBMS and the file system. IBM has devoted significant resources to this effort and has applied for patents covering its implementation of DataLinks.

This paper focuses on management of external data from within the ORDBMS through DataLinks technology. We first describe, from the customer's perspective, the business problems addressed by DataLinks and then discuss how the technology actually works.

BUSINESS APPLICATIONS FOR DATALINKS

INTRODUCTION

Customers understand the benefits of DataLinks

As with any new RDBMS technology, it is important to get the user's perspective on functionality and business benefits. While CAD/CAM applications created the initial customer push for DataLinks, integrated content-management is required in a broad array of market segments. We interviewed some DataLinks customers, all of whom are very clear on the advantages of DataLinks. In this paper we describe one company that plans to use DataLinks to coordinate the management of Web content, and another that is using DataLinks in a CAD/CAM environment to manage product geometry in synchronization with product information stored in DB2 UDB.

All of these organizations have cited the same overall benefits of DataLinks. All like the fact that DataLinks doesn't require them to physically move files stored in the file system into DB2 UDB in order to gain the benefits of DB2's robust management capabilities. This means that existing applications that create or access the files do not have to be modified to run against DB2. Second, DataLinks eliminates the need to develop and maintain homegrown, custom software to manage referential data integrity between the DBMS and the file system. This results in significant savings and enhanced productivity for these organizations. Third, turning over responsibility for managing referential integrity to DB2 minimizes the opportunity for integrity errors between external files and DBMS data. This reduces error-checking requirements and raises the level of confidence in the accuracy and consistency of important business data. DataLinks also enables DB2 to manage the overall security of these data and ensure properly-authorized access. A final benefit is the ability to maintain performance and speed of access by locating external files in close proximity to applications, and to continue using the file system directly for read and write operations.

WEB-BASED CONTENT MANAGEMENT

HotOffice provides Web-based Intranet services to small businesses

HotOffice Technologies, Inc. is a three-year-old company that takes advantage of the Internet's universal access and worldwide reach to provide an out-sourced, Web-based Intranet for small businesses and groups. With this Web-based service, HotOffice offers customers an Intranet on the Internet with access to the following suite of applications:

- Group communications such as e-mail, calendar, contact manager, reminders, and phone messages.
- Document management for central document access with a file viewer and complete version control. On request, HotOffice can convert standard document formats to HTML so users do not need word-processing software to view documents.
- Collaboration such as threaded bulletin boards (forums), online conferencing, and Internet phone calls.
- Third-party business services including office products and services, travel, technology and communication, government and legal, marketing, finance, news and reference, and many more.

Customers minimize their infrastructure investment

The user only needs a Web browser to access these applications over the Internet; the organization does not have to invest in networking, application software, Web servers, or Intranet applications development. The customer simply subscribes and pays a fee based on the number of users and amount of disk storage used. Users can communicate and share information via their HotOffice regardless of where they are located. The only additional software that is installed on the client machine is HotOffice's software for downloading documents and email attachments. This optional software provides data encryption and security during transmission of sensitive documents over the Internet. Customer documents are encrypted twice—once when they are sent across the Internet and again when they are stored on the HotOffice server.

Content management is a key requirement

Currently, HotOffice uses a multi-tier architecture: a browser on the front end, a Web server, a middle-tier application server, and DBMS and file servers on the back end. The Web server, application server, file servers, and the DBMS run on Windows NT. When a user publishes a document for inclusion in the HotOffice document-management system, the document is stored in an NT file server on the back end, and the file name and location are entered in the database. In addition to file references, the DBMS stores all user profile information. HotOffice has written its own custom software—a set of Java programs on the middle tier—to maintain integrity between the database references and the NT files.

HotOffice is not currently using DB2 UDB as the DBMS, but plans to migrate to UDB very soon for two reasons. One is the need to effectively handle a large volume of data by providing parallel query processing and fault-tolerant operation. Combining both database and file data, the company expects to be managing several terabytes of customer data. Based on its experience to date, HotOffice has concerns about the ability of the existing DBMS to ensure both performance and high-availability as data volume grows into the terabyte range.

DataLinks will manage the links between DBMS data and NT files

The second reason for moving to DB2 UDB is DataLinks. HotOffice “can’t wait” to implement DataLinks, and will use it to integrate into UDB the management of customer documents stored as NT files. The company will soon be beta-testing DataLinks on NT. HotOffice expects to reap the following benefits from DataLinks:

- DataLinks will enable HotOffice to eliminate the cost of maintaining its own custom software for keeping the database file references and NT files in synch.

- HotOffice can continue to store documents in the NT file system rather than moving them into the database. This will not require modification to the middle-tier applications that currently retrieve these documents.

- HotOffice will be able to rely on the DB2 DBMS for referential integrity between the database and the file system.

Integrity is a primary concern for the company, and it prefers to have the DBMS control integrity, not a user program. Given the number of files—and versions of those files—that HotOffice manages, it is difficult and costly to develop homegrown software that maintains the level of reliability and integrity required by HotOffice, and virtually impossible to achieve the same levels of reliability and integrity provided by DB2 UDB. “It is critically important that when a customer deletes something, it is deleted both from the file system and from the database. Today, we need to perform periodic maintenance procedures to assure that file references in the database accurately reflect the documents stored in the file system. With DataLinks, UDB will take care of this issue, so we can focus on continuing to build and add to our core-business applications for our clients.”

CAD/CAM APPLICATION SOFTWARE

Dassault Systèmes

Moving toward 3D modeling tools in CAD/CAM

Engineering software for computer-aided design and manufacturing (CAD/CAM) has been in existence for over 20 years. CAD/CAM applications are used by manufacturing companies to design, analyze, and manufacture parts and to assemble parts into entire products. Engineers use CAD/CAM tools to create both design documents, or engineering drawings, and part specifications.

The most recent CAD/CAM development has been the move from two-dimensional (2D), surface-oriented visualization of models to three-dimensional (3D), variational, knowledge-based, solids-modeling techniques. With 3D design tools, engineers can actually peer “inside” parts, rotate them 360 degrees, and create assemblies in 3D to see how components fit together. The software itself can identify the physical proximity of components—whether or not they touch or overlap and the amount of clearance, if any.

A singularly impressive example of the advantages of 3D design tools is Boeing’s newest aircraft, the huge, twin-engine 777. Earlier Boeing airplanes were designed with 2D tools, and required Boeing to build full-scale mockups of major airplane assemblies for testing. This was the only way to confirm without a doubt that the parts fit as required. With the 777, Boeing used 3D tools and did not have to make any full-size mockups to verify the airplane design before actual production began. The company simply made the first production plane and flew it. And the 777 fits together more precisely than any of its predecessors. For example, while the distance between the tip of each wing and the ground on a 747 may differ by as much as a foot, the difference on the 777 is a fraction of an inch. (Boeing accomplished all of this using Dassault Systèmes' CATIA CAD/CAM design environment; see below.)

RDBMS integration for coordinated management of data and files

Another major trend in the evolution of CAD/CAM software is the ability to associate engineering drawings (called models) and their contents, usually stored in flat files, with product data and data about the models stored in a relational database management system. Dassault Systèmes, one of the largest CAD/CAM software vendors in the world, already provides some level of integration between its CATIA CAD/CAM design environment and RDBMSs such as IBM's DB2 UDB. (CATIA, pronounced ka-TEE-ya, stands for Computer-Aided Three-Dimensional Interactive Application.) Data stored in the RDBMS are typically used for assembly management (i.e., a parts hierarchy). Dassault has many large customers with a total of over 100,000 individual users worldwide.

Dassault Systèmes now supports DataLinks

A central component of the new generation of Dassault's product line, which is now generally available, is ENOVIA$_{\text{VPM}}$ (VPM stands for Virtual Product Model). ENOVIA$_{\text{VPM}}$ is designed to provide a consistent, distributed model of products and their associated processes. This means that ENOVIA$_{\text{VPM}}$ can manage many relationships between data stored in the RDBMS and data stored in the file system, such as geometric data. The DataLinks technology has been implemented as part of ENOVIA$_{\text{VPM}}$ to help maintain the consistency of these relationships. For Dassault, DB2 UDB and the embedded DataLinks technology provide four primary capabilities:

- Logical data consistency—For example, an engineer cannot delete or rename a file which is referenced by its corresponding part description in the database.

- Transaction consistency—If a transaction is rolled back in the database, the link to the appropriate version of the file at this site is maintained.

- Security and access to files controlled by ENOVIA$_{\text{VPM}}$ with the possibility for external applications to have direct read-access to these files.

- Synchronized backup and recovery of database and file data—Using DB2 with DataLinks ensures consistent backup and recovery of ENOVIA$_{\text{VPM}}$ metadata and the associated CATIA files. This makes the overall process more automatic and less database administrator (DBA)-intensive. In the past, administrative tasks were performed outside the CATIA environment, requiring a separate backup strategy for external CATIA files and introducing a significant risk of inconsistencies between the database and related external files.

DB2 with DataLinks guarantees the integrity of links to external data

DB2's ability to *guarantee* the integrity of the link between a file and the database is one of the most powerful features of DataLinks. First, it is no longer necessary to explicitly backup CATIA files; DB2 and DataLinks automatically generate a backup of any file as soon as it is linked to the database. Second, if the database needs to be restored, DB2 automatically checks the validity of the file links and may retrieve the backup copy of a file if the current version is not in synch with the database. A file lost, perhaps following a human, software, or hardware error, can be retrieved either during restore or explicitly, using a new DB2 command called "reconcile." This command ensures that all DataLinks values reference valid files, retrieving the files from backup storage if necessary.

Another DB2 UDB feature with potential benefit that Dassault is exploring is DB2's object-modeling capabilities. ENOVIA$_{VPM}$ uses the new STEP standard, which includes an object data-definition language called Express to model the entire set of product data, including geometry, hierarchical parts explosions, etc. The use of DB2 UDB object-relational constructs, such as row objects and reference types, would provide an easier and more natural way to map CATIA objects to relational data.

Boeing

Boeing saw early need for DataLinks functionality

One of Dassault's long-time customers is The Boeing Company, the $23 billion aerospace company and commercial aircraft manufacturer. Boeing has used CATIA for engineering design for over ten years and currently has 8,000 CATIA users. The primary CATIA platform for the design of existing planes is the IBM mainframe. Boeing is moving to a distributed version of CATIA on a global network of Unix workstations and servers.

As background, Boeing was one of the test sites for IBM's prototype relational DBMS, System R, back in the late 1970s. At that time, Boeing was already aware of the need for functionality like that provided by DataLinks given the fact that the company had to manage information generated by a number of graphical CAD systems and various engineering-analysis applications.

The problem: synchronizing data on millions of parts

The biggest configuration problem for Boeing was keeping the metadata about the designs synchronized with the contents of the files. The 777 alone, for example, has 1.5 million parts managed by part number and hundreds of thousands of associated design files. As a result, Boeing was a pioneer in integrating CATIA with the DB2 relational DBMS. The company wrote its own application to associate DBMS data with CATIA model drawings. The application creates rows in a DB2 database and associates the data in each row with one or more drawings by storing the file names in selected rows. The application also stores DB2 data (such as the primary key of the associated row) within records in the CATIA system. This enables Boeing to cross-reference CATIA models with properties stored in the DBMS, such as the general characteristics of a drawing and the discrete properties of details in the drawing (e.g., the material of which a part is made or the airplanes to which the part applies).

The success of this integrated CATIA/DB2 database approach obviously depended on the ability to maintain the integrity of the links between the CATIA models and the DB2 data. Internally-developed validation routines helped here but implementation was limited because of concerns about the ability of the application to manage large numbers of these links.

Today, Boeing uses a variety of highly specialized engineering applications, including CATIA, each of which has its own unique file system. The company expects to continue operating in this environment and that it will become even more complex over time. Therefore, it makes sense to use a DBMS for overall management of both design documents and product data.

In August 1998, Boeing implemented Dassault's ENOVIA$_{VPM}$, which includes DataLinks, in both a test environment and a pilot production environment for the 737 in the Propulsion System Division. (The propulsion system includes the engine and all of the associated parts under the wing of the airplane.) The pilot project manages engine data that are still in the engineering-design phase and not yet released—i.e., they don't necessarily represent the configuration of the airplane that will be delivered; these are stored in a separate "release repository." Boeing plans to transition the release repository to ENOVIA$_{VPM}$ in the year 2000.

Improving data integrity with fewer resources

A major benefit of DataLinks is system-managed integrity. DataLinks enables DB2 to provide the required administrative infrastructure for managing the database world and the file-based world together as an entity. This is where the largest savings occur for the customer. The ability to have more information more accurately tied together via the DBMS also results in increased productivity for the designers.

The ultimate benefit is flexibility in structuring applications in the future. DataLinks can enable a higher degree of "plug and play" between the database and file systems, and enable a single application, such as a workflow application, to span components in both environments. Boeing sees the potential to "construct clever, flexible applications in the future with DataLinks as a starting point."

OTHER APPLICATIONS

Managing libraries of images, documents, video, programs

Maintenance of digital libraries of information is another major area where customers see significant value in DataLinks. One company is preparing a system to manage access to a large number of historical photographs. The photographs will be scanned into digital form and stored as flat files with metadata about the photos and references to the files stored in DB2 UDB. Users will be able to retrieve photo information through Web browser access to DB2 and non-DBMS-enabled tools will be able to access the photos for manipulation of the digital images.

Another company needs a way to associate metadata with a large file of video clips. Currently, a database contains information about the contents and location of videos and their corresponding thumbnails stored in the file system. Users can search through the database and use the file references to retrieve thumbnails and/or videos that match search criteria—clouds, for example. But because the database itself doesn't manage the files, mismatches can occur; the thumbnail may match the description but the movie itself does not, or vice versa. Using DataLinks technology would prevent these problems and eliminate the task of manually revalidating the references in the database.

A similar application is a publishing company's need to store large amounts of multimedia data stored in millions of files of up to 10MB each. This organization would like a mechanism for rolling back, recovering, and restoring these files in synch with the metadata and attributes which are maintained in DB2 tables.

IBM's Net.data takes advantage of DataLinks

Closer to home, IBM's own Net.data now exploits all of the DataLinks functionality. Net.data, bundled with DB2 UDB, "web-enables" UDB by providing essentially an intermediary between the Web and UDB. It allows any data stored in UDB to be accessed using HTML. Net.data receives HTML from a Web server, dynamically builds queries from the HTML, and passes the queries to UDB. When the results are received back by Net.data, it has access to each column in each row as variables and communicates the requested data back to the browser as appropriate.

With support for DataLinks, Net.data can now retrieve data from either the database or from an externally linked file; the location of the data is transparent to the user. A significant advantage here is the ability to continue to store files on the Web server for performance reasons (eliminating the need to transfer them from UDB to the Web server) yet manage them from within UDB.

HOW DATALINKS WORKS

DATALINKS COMPONENTS

DataLinks works with existing file-system APIs

Here is how DataLinks works from an application perspective. In a DataLinks environment, DB2 UDB provides the metadata repository for external data. Attributes and subsets of the data stored in external files are maintained in DB2 tables along with logical references to the location of the files (e.g., a server name and a file name). The application searches the DB2 metadata via the SQL API to identify external files of interest. Examples would be finding the following: engineering drawings that apply to a particular automobile model and include specific part numbers, a video clip used in a 60-second TV commercial within the last year that contains images of paper money, or all of the email attachments received within the last six months that concern customer profiles. DB2 provides the references for selected files and the application then accesses the file data directly using standard file-system API calls (file-open, etc.) Using standard file APIs is very important to support existing applications without having to modify either the application or the file system (see Figure 1).

Under the covers, DB2 UDB is responsible for managing the links between the database and the files in the areas of referential integrity, access control, coordinated backup and recovery, and transactional consistency.

DataLinks can reference files across multiple file systems

IBM’s implementation of DataLinks has several components:

- A DATALINK data type—This is a new built-in, base data type in DB2 UDB.
- The DB2 Data Links Manager software on the file server—This has two distinct software modules: the Data Links File Manager (DLFM) and the Data Links Filesystem Filter (DLFF).
- DBMS/DLFM APIs that DB2 uses to communicate with the DB2 Data Links Manager on the file server.

DB2 UDB and the DB2 Data Links Manager software can run on different platforms, and DB2 Data Links Manager itself can run on multiple, heterogeneous file systems (see Figure 2). So a single DB2 database can contain references to files stored in multiple, distributed file systems, allowing files to be located close to their application to reduce network traffic and enhance application performance.

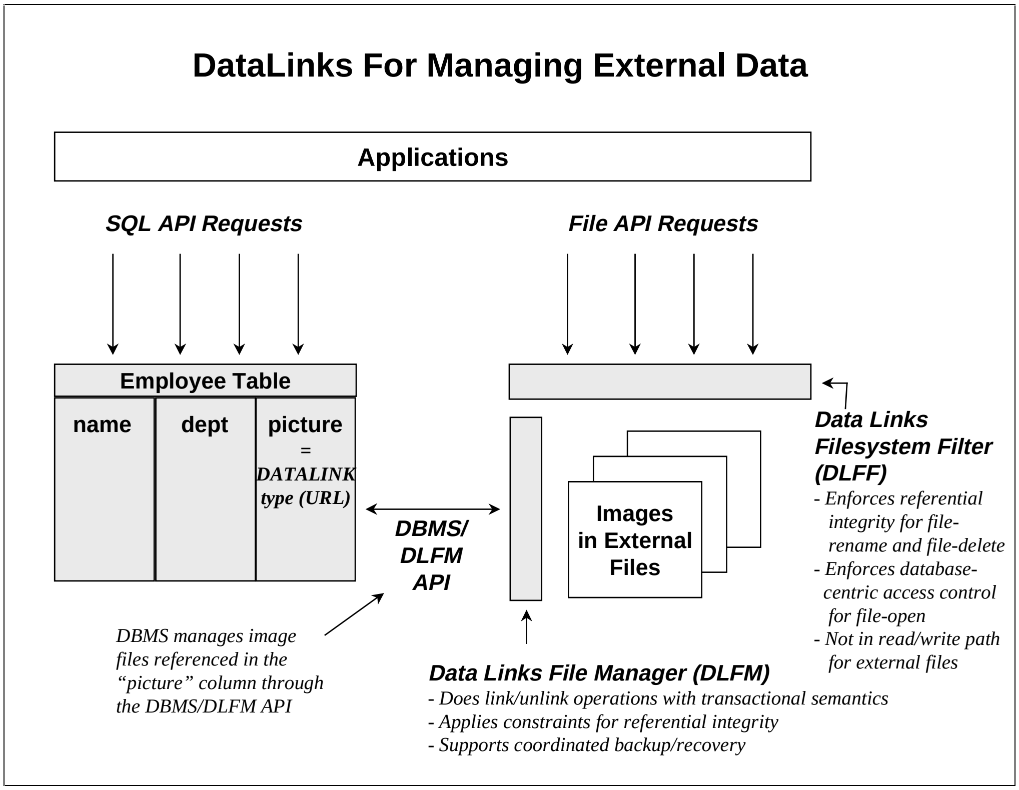
DataLinks For Managing External Data
Applications
SQL API Requests
File API Requests
Employee Table
name
dept
picture
=
DATALINK
type (URL)
DBMS/
DLFM
API
Images
in External
Files
Data Links
Filesystem Filter
(DLFF)
- Enforces referential integrity for file-rename and file-delete
- Enforces database-centric access control for file-open
- Not in read/write path for external files
DBMS manages image files referenced in the "picture" column through the DBMS/DLFM API
Data Links File Manager (DLFM)
- Does link/unlink operations with transactional semantics
- Applies constraints for referential integrity
- Supports coordinated backup/recovery

Figure 1. Applications use the SQL API to retrieve data and references to external files from DB2, and a standard file API to access the external files directly.

THE DATALINK DATA TYPE

A new DATALINK data type stores file URLs

The database-application developer uses the DATALINK type when creating any new column in a DB2 table that represents data in an external file. In the example in Figure 1, photographs of employees are stored externally and referenced in the "picture" column of the employee table. The picture column is a DATALINK type whose value points to an external image file.

DataLinks provides a range of options for controlling external files

Figure 3 shows sample SQL syntax for creating the picture column as a DATALINK type in the employee table. The value stored in the DATALINK column includes a pointer to the external file in the format of a URL (Uniform Resource Locator) as used in the Web environment—http://servername/pathname/filename. The DataLinks software understands how to parse the URL and identify the information it needs to locate the requested file. The table definer has a range of options for managing the external files referenced in the DATALINK column, from no control through full control by the DBMS. The following parameters are used to define the level of DBMS control over the linked files:

- Integrity—For any value provided in the DATALINK column, DB2 should expect to find DB2 Data Links Manager software installed on the referenced file server and DB2 Data Links Manager can, therefore, manage the file.

DB2 UDB can control read and write permissions over external files

- Read permission—Read permission can be retained by the file system or given to the DBMS. If the DBMS is used to grant read permission, access rules as defined in SQL (e.g., through SQL views) govern who can read the files. Read authorization is automatically checked when the application requests a file reference from the DBMS, and a token is embedded in the URL if the application has permission to read the file. This token is recognized by the DB2 DataLinks Manager software on the file server and the read request is passed through to the file system. Without a valid token, the read request is denied by the Data Links Filesystem Filter.

- Write permission—Write permission for managed files can be left with the file system or blocked. "Blocked" means that a user cannot update a referenced file in place; the user must create a new version and then switch the link in the database through a SQL "update" statement. If the file system retains write permission, files can be updated in place. In this case, there is no "update" statement issued against the DB2 table. Therefore, DB2 cannot support coordinated backup and recovery because it is not informed about updated files.

- Recovery—The "yes" option tells DataLinks to coordinate backup and recovery between the referenced files and the associated database.

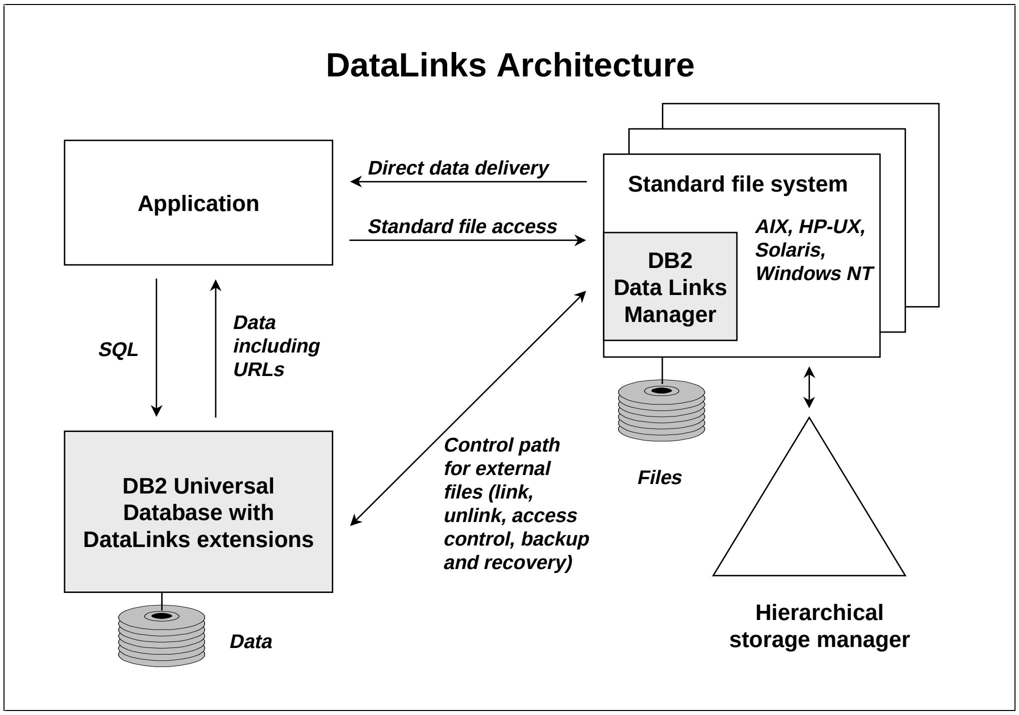

Figure 2. DB2 Universal Database with DataLinks extensions communicates with the DB2 Data Links Manager software on the file system to manage external files.

- On unlink—When a file is unlinked from the database, DB2 Data Links Manager can be instructed to delete the file from the file system or to restore control over the file to the file system. In the latter case, DB2 Data Links Manager restores the file-system ownership and access permissions that were in effect before the file was linked to the database.

SQL calls to the DBMS initiate file link and unlink operations

Before linking an external file to the database, the application first creates the file in the file system. (This process does not involve DB2 Data Links Manager.) Then to link the file with the database, the application simply inserts the URL value in a DATALINK column. This SQL “insert” statement prompts DB2 to initiate a *LinkFile* operation through DB2 Data Links Manager to establish the appropriate level of DBMS control over the file. Similarly, a SQL “delete” statement that deletes a row containing a DATALINK column will use the URL value to issue an *UnlinkFile* operation; the unlinked file will be deleted from or returned to the control of the file system. A SQL “update” statement that involves a URL value results in an *UnlinkFile* operation for the old URL and a *LinkFile* operation for the new URL.

THE DATA LINKS FILE MANAGER

The DLFM applies referential integrity constraints and access control

The Data Links File Manager (DLFM) component of DB2 Data Links Manager plays a key role in managing external files. It is responsible for executing the link/unlink operations with transactional semantics within the file system. To do this, the DLFM maintains its own repository about files that are linked to (referenced in) the database.

When a file is initially linked to the database, the DLFM applies the constraints for referential integrity, access control, and backup and recovery as specified in the DATALINK column definition. If the DBMS controls read access, for example, the DLFM changes the owner of the file to the DBMS and marks the file “read only” (file-rename and file-delete are disallowed at this point as mentioned above).

A Sample SQL3 DATALINK Type Definition

```
CREATE TABLE employee (
   name VARCHAR (30),
   dept INTEGER,
   picture DATALINK (200)
           LINKTYPE URL
           FILE LINK CONTROL
                      INTEGRITY all
                      READ PERMISSION DB
                      WRITE PERMISSION blocked
                      RECOVERY yes
                      ON UNLINK restore
);
```

Figure 3. Here is the SQL syntax for creating a column in a table as a DATALINK data type with read-access control over the referenced files by the DBMS (DB). This syntax as proposed by IBM has been accepted by ISO and ANSI for inclusion in the SQL3 standard.

All of these changes to the DLFM repository and to the file system are applied as part of the same DBMS transaction as the initiating SQL statement. If the SQL transaction is rolled back, the changes made by the DLFM on the file-system side are undone as well.

The DLFM also coordinates backup/recovery of files

The DLFM is also responsible for coordinating backup and recovery of external files with the database. When the DBMS transaction that includes a *LinkFile* operation commits and the DBMS is responsible for recovery of the file, the DLFM initiates a backup of the newly-linked file. This file backup is done asynchronously and is not part of the database transaction for performance reasons. In addition, note that by doing it this way, the database backup itself is not slowed down because the referenced file has already been backed up. This is particularly important in the case of very large files. Coordinated backup and recovery of external files with DB2 data can be done directly to disk or through an archive server supported by DB2 UDB, such as IBM's ADSTAR Distributed Storage Manager (ADSM).

IBM has taken into consideration within its implementation of the DataLinks technology the alternative ways of updating a file in order to ensure that 1) a file is available while it is being updated, 2) the correct file version is referenced in the database, and 3) transaction consistency is maintained. A referenced file can be updated in one of two ways, and DataLinks currently provides different levels of support for each.

Method of updating a file affects backup/recovery support

In the first update method, the user makes a copy of the referenced file, updates the copy, and saves it with a different file name. The user then issues a SQL "update" statement to reference the new version of the file in the DATALINK column, and to update the metadata stored in DB2 as appropriate. The DLFM unlinks the previous version of the file and links the new version, all within a transactional context. If the transaction aborts, the DLFM restores the link to the previous version of the file. (As a note, the new version of the file is not deleted by DataLinks even if the transaction aborts.) The DLFM also completes its standard link/unlink processing after the transaction commits; it makes a backup of the new version and deletes or restores the unlinked version. Therefore, if a file is updated multiple times within a database backup interval, each version is backed up.

In this method, users can continue to access the currently-referenced version of the file while it is being updated. This is particularly valuable in environments such as engineering, where a file may be updated over a period of days or weeks. This method also ensures that the correct version of the file is referenced in the database and that coordinated backup and recovery is possible. Because the metadata and the file references in the database are in sync with the file content, DataLinks restores the correct versions of the files in the file system if the database must be recovered.

The second update method is to update the file "in place;" that is, the user opens the file for update, makes changes, and closes the file. Updating a referenced file directly in this manner means that the DBMS is not aware that the file has been changed and, therefore, cannot coordinate recovery between the database and the file. Today, the file system must retain write permission over any files that can be updated in place. IBM is working on advanced solutions to support coordinated recovery for update-in-place. It is important to note, however, that DataLinks still provides referential integrity over such files, ensuring that the file is not renamed or deleted as long as it is referenced in the database.

Multiple versions of files are tracked for point-in-time recoverability

As mentioned above, the DLFM tracks different versions of a referenced file and maintains the backup status of each in order to support point-in-time recovery. The DBMS also provides the DLFM with a "recovery ID" for a file whenever it is linked or unlinked to help synchronize recovery of files with data. Otherwise, DataLinks would not be able to restore the file to the correct point of database consistency. IBM recognizes the importance of identifying all of the potential pitfalls in this area and ensuring that DataLinks is up to the challenge of coordinated backup and recovery of files across multiple file systems and multiple DLFMs.

When the DBMS does a backup of its data, it communicates with the DLFMs to ensure that all of the necessary asynchronous copy operations for referenced files are complete before declaring that the database backup has been successfully completed. The DBMS backup utility has been extended to handle this level of communication and to keep additional information in the backup image about which file servers and groups of files were involved in the complete backup.

Backup copies of unlinked files may be kept for a specific number of database backup cycles, again in case the database is restored to a point in time in which the file was still linked to the database. The DLFM is also responsible for “garbage collection” of backup copies of unlinked files that are no longer required by the DBMS.

THE DATA LINKS FILESYSTEM FILTER

The DLFF intercepts file-system requests for linked files

The Data Links Filesystem Filter (DLFF) is a thin, database-control layer on the file system that intercepts certain file-system calls (e.g., file-open, -rename, and -delete) issued by the application. If the file is referenced in a database, the DLFF is responsible for enforcing the referential-integrity constraints and access-control requirements defined for the file. This ensures that any access request meets DBMS security and integrity requirements.

DLFF will, for example, reject a user-level request to rename or delete a file referenced by the database. This avoids “dangling pointers” in which a file is referenced in the database but the actual file doesn’t exist. This is a typical problem for organizations that write their own software to link data in the database with data in external files.

Valid tokens are required for read access to DBMS-controlled files

The DLFF also validates any authorization token embedded in the file pathname for a file-open operation. In the example in Figure 1, when a user submits a SQL query to retrieve an employee picture, the DBMS checks to see if the user has permission to access the image (i.e., has select privilege on the table or view that is the target of the SQL query). If yes, the DBMS returns to the application the file name. If the DBMS is responsible for access control, an authorization token is embedded in the file name as part of the value returned for the DATALINK column. The application then uses the file API to retrieve the image. When the DLFF intercepts the file-open command, it validates the authorization token to determine whether or not to pass the file-open command through to the file system. If the validation fails, the file-open request is rejected. An important point here is that the DLFF is *not* in the read/write path for file access. Once access has been authorized by a valid token, the application interacts directly with the file system for delivery of the file. This guarantees the application the same speed of access as a pure file system without DataLinks.

THE BENEFITS OF DATALINKS AND ALTERNATIVES

BENEFITS

DB2 UDB becomes a platform for managing all business content…

Now that it is clear how DataLinks works and how customers are using it, let's summarize the major benefits of DataLinks. One of DataLinks' significant strengths is that it transforms DB2 UDB into a content-management platform. Organizations can continue storing data in external files to take advantage of file-system capabilities while coordinating the management of these files and their contents with associated data stored in DB2 UDB. DataLinks gives DB2 UDB comprehensive control over external data in the following areas: referential integrity, security and access control, coordinated backup and recovery, and transaction consistency.

…while leveraging the file system and existing applications

Other key benefits of DataLinks relate to its ability to leverage the capabilities of the file system. These include the following:

- Allows external files to be stored in close proximity to the application. This reduces network traffic and maximizes application performance because files don't have to be stored in the DBMS to be managed by the DBMS. Therefore, the files don't have to be transferred across the network to the application when requested.

- Maintains speed of file access by continuing to use the file system for this, not the DBMS. This is also critical to application performance. No DBMS can match the performance of the file system in this area.

- Allows applications that already access the external files directly to continue to do so without modification. This means that the applications can continue using their current APIs for file access.

- Provides the ability to leverage the functionality of any file system on Unix or Windows NT, such as support for hierarchical storage managers.

ALTERNATIVES

None of the alternatives to DataLinks provide the same benefits

There are alternatives to DataLinks that attempt to realize the same overall objective as DataLinks: effectively manage business data and content as if they were stored in the database (requirement 1) without losing the significant advantages of the file system (requirement 2). However, none of them are completely successful in doing this, with all of them missing the mark on one of the two requirements.

One obvious alternative is to maintain the status quo, i.e., continue to use the file system to manage external data. This meets requirement 2 but not requirement 1. We have already discussed the other disadvantages of this alternative earlier in this paper.

Another alternative is to move the external files into the database and store them as large-object (LOB) or abstract data types. While this provides the data-management functionality specified in requirement 1, existing applications would now have to be modified to access the files using SQL, the DBMS's API. Another potential disadvantage is the inability to maintain performance when accessing the files as

LOBs, particularly if the files are large and the DBMS is not physically located close to the application. A related performance issue is speed of backup. The database will be much larger if files are stored internally as LOBs, and backup will take longer.

A third alternative builds on the previous one: store external data in LOBs but also provide file-API access to the DBMS to access the LOB data/files. Thus, existing applications can continue to use their native APIs against the DBMS. However, this model doesn't necessarily solve the performance problem as the DBMS cannot match the file system's speed of access. This alternative also may not address the need to locate the file close to the application, again for better performance, because the "file system"—the DBMS—may not be installed everywhere. Third, this alternative does not support delivery of isochronous data such as video and audio. And finally, new developments in the file system cannot be leveraged.

AVAILABILITY

DataLinks shipped with DB2 UDB 5.2

One of IBM's goals is to make sure that new ORDBMS technology is implemented across all supported platforms and all members of the DB2 family. IBM shipped DataLinks with DB2 UDB version 5.2 in September 1998 on the AIX platform. DataLinks for NT is now in beta and will be delivered with the next release of UDB. Work is also in progress to support DataLinks on Solaris. IBM understands the importance of maintaining functional compatibility among the DB2 family, and DataLinks for the AS/400 will be included in Version 4 Release 4 of DB2 for OS/400. IBM will be porting DataLinks to other platforms, such as HP and OS/390 based on customer requirements.

CONCLUSIONS

DataLinks is an innovative solution to a challenging application requirement

IBM's DataLinks is an example of new and innovative DBMS capability. By providing tight integration of file-system data with the object-relational DBMS, DataLinks allows DB2 UDB to guarantee the integrity of data whether they are stored inside or outside the database. This functionality is a key requirement of an extensible, data-management architecture. Delivery of DataLinks clearly illustrates IBM's continued ability to provide robust, industrial-strength database management systems that meet real customer requirements.

While companies in the CAD/CAM application marketplace were the early supporters of DataLinks, other market segments clearly recognize that DBMS/file-system integration applies to a very broad range of application problems, and is a key requirement of effective content management. Web/Internet/e-commerce applications are important examples here. DataLinks will enable DB2 UDB to combine comprehensive object-relational extensibility with the ability to manage external data. This will provide a flexible and appealing application-development platform for a wide variety of organizations.

IBM white papers are available on the following IBM web site:

http://www.software.ibm.com/data/pubs/papers

ABOUT THE AUTHOR

Judith R. Davis is an independent consultant with over 15 years of experience in the business application of information technology. Her primary interest is database management systems and related technology. She is a respected consultant and analyst in this area and has worked extensively with users and major vendors. She has also authored many articles for IT publications and magazines. Most recently, she has focused on the technology and business requirements driving the development of object extensions to the relational DBMS, and the role that the object-relational DBMS plays in creating a fully-extensible data-management environment.