**IN THIS APPENDIX**

# APPENDIX E

# DB2 on Other Platforms

Although DB2 began its life on the mainframe, the advent of client/server technology and the success of competing RDBMS products (such as SQL Server and Oracle) caused IBM to create versions of DB2 for additional platforms. In short, DB2 is no longer just a mainframe product.

## The DB2 Family

Versions of DB2 exist for a large array of platforms, of which the mainframe (z/OS and OS/390) is only one (see Table E.1). These products are all collectively referred to by IBM as the *DB2 Family*. Individually, each DBMS is referred to as *DB2*, or *DB2 Universal Database Server*. The proper way to refer to any individual offering in the DB2 family is *DB2 for (operating system)* (for example, DB2 for z/OS or DB2 for Windows). If the version of the DB2 product has been extended with object/relational capabilities, it is referred to as *DB2 UDB for (operating system)*, where the UDB stands for Universal Database. Version 6 was the first version of DB2 for z/OS that earned the UDB moniker.

Many shops implement applications on several different platforms and interconnect them using client/server development methods.

**TABLE E.1** The DB2 Family of Products

| Platform | Operating System | AKA (Old Name) |
|---|---|---|
| iSeries (AS/400) | OS/400 | SQL/400 |
| Mainframe | MVS, OS/390, z/OS | DB2 |
| Mainframe | VM | SQL/DS |
| Mainframe | VSE | SQL/DS |
| Intel | Windows NT/95/98 | --- |
| | Linux | --- |
| | SCO UNIXware | --- |
| IBM pSeries | AIX | DB2/6000 |
| Hewlett Packard | HP-UX | --- |
| Sun | Solaris | --- |
| PDA | Palm Computing | --- |
| | PocketPC | Windows CE |

**NOTE**

The PDA version of DB2 is referred to as *DB2 Everyplace*. In addition to Palm and Windows PocketPC, DB2 Everyplace can run in the following environments: Symbian, embedded Linux, QNX Neutrino, Microsoft Win32 and Linux.

## Different Code Bases

There are four distinct code bases for the products under the DB2 brand. The mainframe has its own code base, as does the iSeries, and VSE/VM. The fourth code base is for Linux, Unix, and Windows (LUW) platforms—and all of the other DB2 offerings originate from this code base.

Having a separate code base means that each of these DB2 “products” was developed independently—at least in the beginning. So, for example, the process used by DB2 for z/OS to optimize SQL differs from the process used by DB2 for Linux. Hopefully, though, the result is similar—an efficient SQL statement.

So, there will be some major differences between the DB2s.

## Some Major Differences

It is obvious that the different DB2 products are not “plug and play” commodities simply because they all share the name *DB2*. There are some big differences among these products in their current releases. The biggest differences are relatively easy to detect and include the following:

- Differences imposed due to operating system constraints (OS/400 versus OS/390 versus AIX)
- Back-level compatibility issues

- Workstation orientation differences such as GUI interfaces and drag-and-drop menus
- Subsystem-centric implementation (z/OS and OS/390) versus database-centric implementation (workstation)

Most of these differences are minor and easy to handle. Indeed, IBM has slowly but surely been making these disparate implementations of DB2 more and more alike with each new release and version. However, there are some "gotchas" lurking under the covers that might be more difficult to find.

Of the basic differences mentioned earlier, the only one that might not be obvious is the focus of the DBMS implementation. DB2 for LUW is database-centric. This implies that each new database carries its own system catalog with it. Additionally, it is not possible to simply access tables across different databases; distributed access is required.

On z/OS, DB2 is subsystem-centric. A single system catalog spans databases. Each subsystem has a unique identification, and you can create multiple databases within it. Distributed requests are not required to access databases within the same subsystem (or, indeed, across multiple subsystems in a data-sharing environment).

### Directories

Another concept that is different at the workstation level is that of a directory. The DB2 for OS/390 Directory houses DBMS system-related information regarding DBD structure, skeleton plan and skeleton package tables, RBA log ranges, and utility control data. The information cannot be updated by the user but is managed and controlled by DB2.

At the workstation level, a directory is another matter altogether. For example, the directory structure used by DB2 for LUW controls the overall environment. The directories used by DB2 for LUW are as follows:

- The *System Database Directory* identifies the databases that can be accessed from the workstation and contains an entry for each local and remote one. Each database entry contains the database name, alias, entry type, and location.
- One *Volume Database Directory* is allocated per disk drive that contains a workstation database. Each entry identifies the location of a specific database on the drive.
- The *Workstation Directory* is used to make a connection to a remote database server. It is used in conjunction with the Database Connection Services Directory to make a connection to a remote host server.
- The *Database Connection Services Directory* is used by DB2 Connect to make a connection to a remote host server.

Not only is it possible for the user to update these directories, it is required. The workstation directories define the environment of DB2 for LUW. Without the proper information recorded in these directories, DB2 might not function in the desired manner. The information in these directories is somewhat analogous to DB2 for OS/390 DSNZPARMs and `SYSDDF`.

### Database Structures

Not all the objects available to DB2 for OS/390 users are supported at the workstation level. For example, hardware-specific DB2 objects such as tablespaces and storage groups are not available for DB2 on other platforms, at least as we are used to dealing with them. Partitioning and segmenting as it is done on the OS/390 flavor of DB2 is not possible. However, DB2 for common servers does provide a feature known as a segmented table. But this is not the same concept as a DB2 for OS/390 segmented tablespace. Common server segmented tables are used to span volumes, enabling DB2 to get around the 2 gigabyte file size limitations under AIX.

The file structure used for databases differs from platform to platform. For example, DB2 for OS/390 uses VSAM Linear Data Sets (LDS) or Entry Sequenced Data Sets (ESDS). A database deployed on DB2 for common servers uses two files for table data: one for normal data and a second to store long fields. These workstation files are flat files, not VSAM files.

Although tables are basically the same for all of the DB2 environments, not all of the DDL options are provided in all of the environments. For example, DB2 for OS/390 does not support triggers, and DB2 for common servers does not allow `VALIDPROC`s, `FIELDPROC`s, and `EDITPROC`s.

### Optimizer Differences

One of the most significant benefits of relational databases is that they provide built-in optimization. The DB2 for OS/390 optimizer is well-known to mainframe DB2 users, but how similar are the other DB2 optimizers?

The DB2 for common servers product uses the latest and greatest optimization technology from IBM's Almaden labs: the Starburst optimizer. Starburst is a database optimization research project that has been covered quite extensively in the academic press. Although some Starburst technology will find its way to DB2 for OS/390, the DB2 for OS/390 optimizer will never be completely replaced by Starburst technology. The DB2 for OS/390 optimizer has been finely tuned for its environment over the course of more than a decade.

Another interesting tidbit is that DB2 for iSeries provides an access method for programmers in which they can bypass the relational engine. This is not encouraged, but it is available.

### Other Differences

Other differences exist between the different implementations of DB2. Some of these are caused by the different release cycles IBM has created for the differing platforms. For example, DB2 for Linux, Unix, and Windows has supported recursive SQL for several releases, but DB2 for z/OS just recently added recursive SQL support in Version 8.

The bottom line is that you need to be aware that there are differences between the DB2s on different platforms. Whenever you use a specific implementation of DB2, you need to be aware of the features it supports that other DB2 platforms do not, as well as the features it does not support that other DB2 platforms do support.

## Packaging and Naming Issues

The actual name of the DB2 edition can be tricky to master on non-mainframe platforms. On the mainframe you just say "I want DB2," and that is what you get. Well, almost. You also have to decide whether you want IBM's utilities or not, too. But things are more difficult in the LUW world. The following packages are all available for DB2 on Linux, Unix, and Windows.

**DB2 Workgroup Server Edition (WSE)** is a multi-user, single host, Web-enabled database with included Java support, but without support for inter- or intra-partition parallelism. WSE fits most closely into the departmental DBMS architecture. It should be used for small systems with a limited number of users. WSE has a registration limit of a maximum of four processors.

**DB2 Workgroup Server Unlimited Edition (WSUE)** has the same functionality, and the same registration limit of a maximum four processors, but a different pricing model. So WSUE seems to fall somewhere between a departmental and an enterprise DBMS architecture—for larger numbers of users than WSE.

**DB2 UDB Enterprise Server Edition (ESE)** is the highest level of DB2 database version with intra-partition parallelism support (the database engine can process SQL statement segments in parallel), and inter-partition parallelism support (process a query in parallel across all of the nodes). ESE has Partitioning and Clustering options as additional add-on features. So, this is the enterprise DB2. This is what used to be known as EEE, sort of.

**DB2 UDB Personal Edition (PE)** is a single-user database engine ideal for deployment to PC based users. PE is IBM's personal architecture DBMS offering.

**DB2 UDB Developer's Edition (DE)** is a low cost package for a single application developer to design, build, and prototype applications for deployment on any of the DB2 client or server platforms. Here is another option for your developers. To save costs you can license DE instead of ESE, WSE, or WSUE, but you cannot use DE for production work.

**DB2 UDB Personal Developer's Edition (PDE)** enables a developer to design and build single user desktop applications. Similar to DE, PDE is a lower cost development option for single user application development.

Then there is **DB2 Express**. It was announced in the first quarter of 2003 and IBM touts it as "a specially tailored database offering for the worldwide small and medium business (SMB) market, with focus on enterprises of size 100 to 1000 employees." This "edition" is targeted at IBM's partners, encouraging them to build applications on top of DB2 Express targeting the SMB market.

The list of DB2s does not stop here. IBM has developed specialty editions of DB2 to handle heterogeneous, federated data and data that is not traditionally stored in databases.

### Federation and Information Integration

No matter how entrenched DB2 might be in your organization, you'll likely have and use other DBMS products, too. Homogeneous IT organizations are extremely rare. Given that fact, how do businesses handle the need to access data from multiple disparate data sources? IBM attacks this problem through federation.

Federation enables businesses to access and integrate data from multiple locations as if it were stored in a single location. Federation enables customers to abstract a common data model across data and content sources and to access and manipulate them as though they were a single source.

IBM's **DB2 Information Integrator** offers the ability to integrate data from multiple sources of different kinds of information. The product enables access to data from hierarchical DBMSs such as IMS, other relational products such as Microsoft SQL Server, and other sources. When it's returned to the application, the data appears as if it came from DB2.

## Content and Records Management

As businesses, government agencies, and other organizations continue to store more data of multiple types, applications will need to manage and unite all kinds of information—structured data plus emails, phone recordings, faxes, video, and other information sources—to solve business problems. RDBMSs traditionally focus on managing structured data, data that can be stored in rows and columns in a table. DB2 extenders expand the kinds of information it's possible to store; however, sometimes more in-depth manipulation and management of this unstructured, complex data is needed.

**DB2 Content Manager** provides support for two kinds of content management: media asset management and enterprise content management. Media asset management is the storage and management of collections of large multimedia objects, such as large collections of X-rays for a hospital, video and film content for movie studios, and scans of art collections for museums. Enterprise content management involves the storage and management of large collections of smaller multimedia objects—often scanned check images for banks or scanned invoices, bills, or similar data for many kinds of businesses.

**DB2 Information Integrator for Content** provides a programming layer above DB2 Content Manager that facilitates easy access to many kinds of information from a single interface.