

---

# Introduction to DB2

**D**ATABASE 2 (DB2) for Linux, UNIX, and Windows is a data server developed by IBM. Version 9.5, available since October 2007, is the most current version of the product, and the one on which we focus in this book.

In this chapter you will learn about the following:

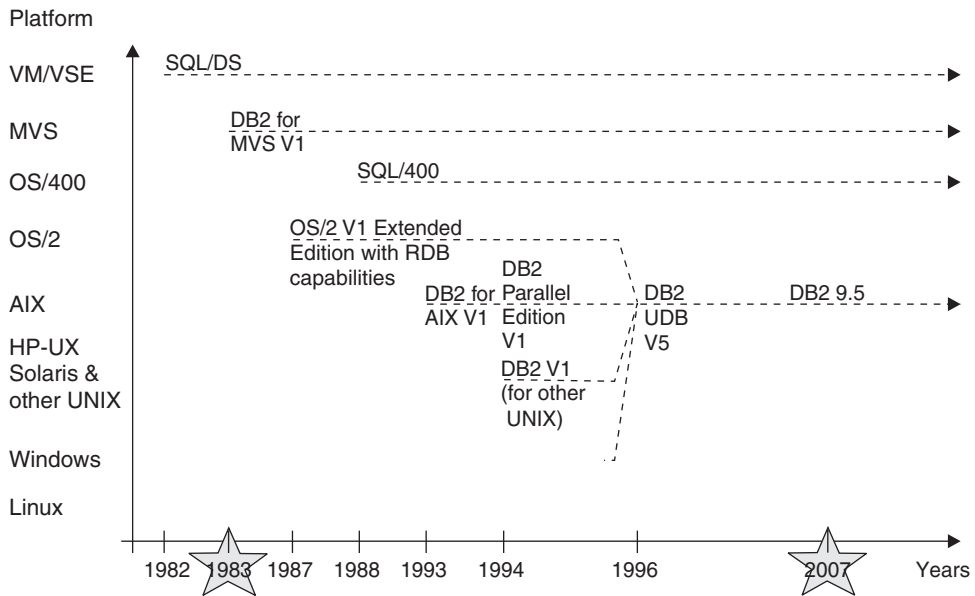
- The history of DB2
- The information management portfolio of products
- How DB2 is developed
- DB2 server editions and clients
- How DB2 is packaged for developers
- Syntax diagram conventions

## 1.1 BRIEF HISTORY OF DB2

Since the 1970s, when IBM Research invented the Relational Model and the Structured Query Language (SQL), IBM has developed a complete family of data servers. Development started on mainframe platforms such as Virtual Machine (VM), Virtual Storage Extended (VSE), and Multiple Virtual Storage (MVS). In 1983, DB2 for MVS Version 1 was born. “DB2” was used to indicate a shift from *hierarchical* databases—such as the Information Management System (IMS) popular at the time—to the new *relational* databases. DB2 development continued on mainframe platforms as well as on distributed platforms.<sup>1</sup> Figure 1.1 shows some of the highlights of DB2 history.

---

1. Distributed platforms, also referred to as **open system platforms**, include all platforms other than mainframe or midrange operating systems. Some examples are Linux, UNIX, and Windows.



**Figure 1.1** DB2 timeline

In 1996, IBM announced DB2 Universal Database (UDB) Version 5 for distributed platforms. With this version, DB2 was able to store all kinds of electronic data, including traditional relational data, as well as audio, video, and text documents. It was the first version optimized for the Web, and it supported a range of distributed platforms—for example, OS/2, Windows, AIX, HP-UX, and Solaris—from multiple vendors. Moreover, this universal database was able to run on a variety of hardware, from uniprocessor systems and symmetric multiprocessor (SMP) systems to massively parallel processing (MPP) systems and clusters of SMP systems.

Even though the relational model to store data is the most prevalent in the industry today, the hierarchical model never lost its importance. In the past few years, due to the popularity of eXtensible Markup Language (XML), a resurgence in the use of the hierarchical model has taken place. XML, a flexible, self-describing language, relies on the hierarchical model to store data. With the emergence of new Web technologies, the need to store unstructured types of data, and to share and exchange information between businesses, XML proves to be the best language to meet these needs. Today we see an exponential growth of XML documents usage.

IBM recognized early on the importance of XML, and large investments were made to deliver pureXML technology; a technology that provides for better support to store XML documents in DB2. After five years of development, the effort of 750 developers, architects, and engineers paid off with the release of the first hybrid data server in the market: DB2 9. DB2 9, available since July 2006, is a hybrid (also known as multi-structured) data server because it allows for

storing relational data, as well as hierarchical data, natively. While other data servers in the market, and previous versions of DB2 could store XML documents, the storage method used was not ideal for performance and flexibility. With DB2 9's pureXML technology, XML documents are stored internally in a parsed hierarchical manner, as a tree; therefore, working with XML documents is greatly enhanced. In 2007, IBM has gone even further in its support for pureXML, with the release of DB2 9.5. DB2 9.5, the latest version of DB2, not only enhances and introduces new features of pureXML, but it also brings improvements in installation, manageability, administration, scalability and performance, workload management and monitoring, regulatory compliance, problem determination, support for application development, and support for business partner applications.

**V9**

**NOTE** The term “Universal Database” or “UDB” was dropped from the name in DB2 9 for simplicity. Previous versions of DB2 database products and documentation retain “Universal Database” and “UDB” in the product naming.

Also starting in version 9, the term *data server* is introduced to describe the product. A data server provides software services for the secure and efficient management of structured information. DB2 Version 9 is a hybrid data server.

**NOTE** Before a new version of DB2 is publicly available, a code name is used to identify the product. Once the product is publicly available, the code name is not used. DB2 9 had a code name of “Viper”, and DB2 9.5 had a code name of “Viper 2”. Some references in published articles may still use these code names.

Note as well that there is no DB2 9.2, DB2 9.3 or DB2 9.4. The version was changed from DB2 9 directly to DB2 9.5 to signify major changes and new features in the product.

DB2 is available for many platforms including System z (DB2 for z/OS) and System i (DB2 for i5/OS). Unless otherwise noted, when we use the term DB2, we are referring to DB2 version 9.5 running on Linux, UNIX, or Windows.

DB2 is part of the IBM information management (IM) portfolio. Table 1.1 shows the different IM products available.

**Table 1.1** Information Management Products

| <b>Information Management Products</b>     | <b>Description</b>   | <b>Product Offerings</b>   |
|--|--|--|
| Data Servers                               | Provide software services for the secure and efficient management of data and enable the sharing of information across multiple platforms.   | IBM DB2<br>IBM IMS<br>IBM Informix<br>IBM U2   |
| Data Warehousing and Business Intelligence | Help customers collect, prepare, manage, analyze, and extract valuable information from all data types to help them make faster, more insightful business decisions.   | DB2 Alphablox<br>DB2 Cube Views<br>DB2 Warehouse Edition<br>DB2 Query Management Facility  |
| Enterprise Content Management & Discovery  | Manage content, process, and connectivity. The content includes both structured and unstructured data, such as e-mails, electronic forms, images, digital media, word processing documents, and Web content. Perform enterprise search and discovery of information. | DB2 Content Manager<br>DB2 Common Store<br>DB2 CM OnDemand<br>DB2 Records Manager<br>FileNet P8 and its add-on suites<br>OmniFind  |
| Information Integration                    | Bring together distributed information from heterogeneous environments. Companies view their information as if it were all residing in one place.  | IBM Information Server integration software platform, consisting of: <ul style="list-style-type: none"> <li>- WebSphere Federation Server</li> <li>- WebSphere Replication Server</li> <li>- WebSphere DataStage</li> <li>- WebSphere ProfileStage</li> <li>- WebSphere QualityStage</li> <li>- WebSphere Information Services Director</li> <li>- WebSphere Metadata Server</li> <li>- WebSphere Business Glossary</li> <li>- WebSphere Data Event Publisher</li> </ul> |

## **I.2 THE ROLE OF DB2 IN THE INFORMATION ON DEMAND WORLD**

IBM's direction or strategy is based on some key concepts and technologies:

- On-Demand Business
- Information On Demand (IOD)
- Service-Oriented Architecture (SOA)

Web Services  
XML

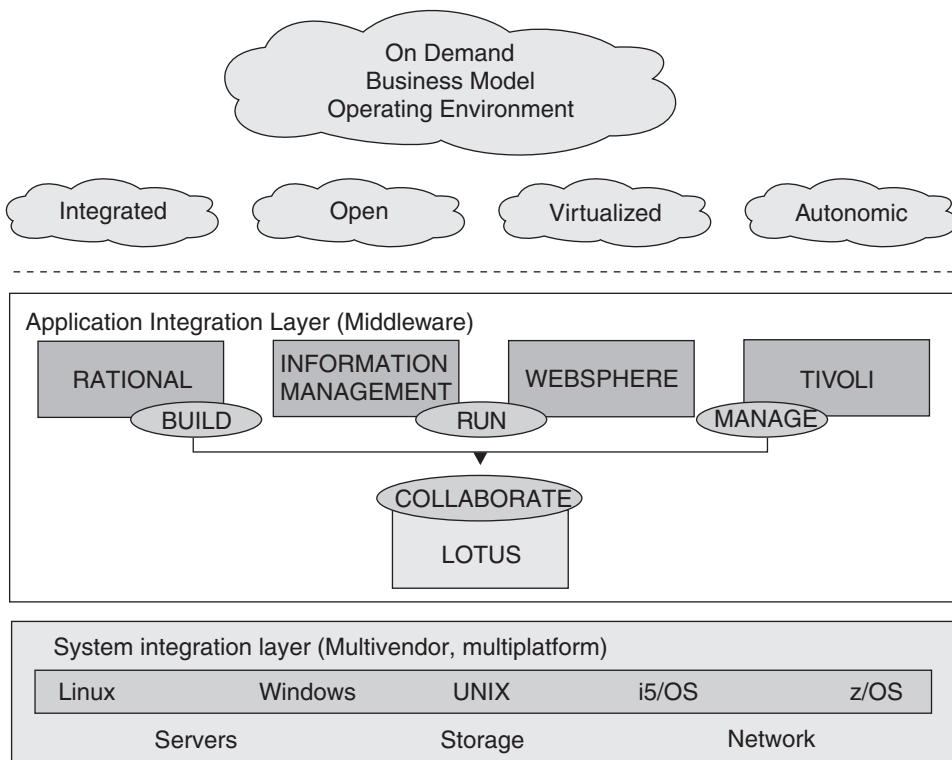
In this section we describe each of these concepts, and we explain where DB2 fits in the strategy.

### 1.2.1 On-Demand Business

We live in a complex world with complex computer systems where change is a constant. At the same time, customers are becoming more demanding and less tolerant of mistakes. In a challenging environment like this, businesses need to react quickly to market changes; otherwise, they will be left behind by competitors. In order to react quickly, a business needs to be integrated and flexible. In other words, a business today needs to be an on-demand business.

An **on-demand business**, as defined by IBM, is “an enterprise whose business processes—integrated end to end across the company and with key partners, suppliers and customers—can respond with speed to any customer demand, market opportunity, or external threat.”

IBM’s on-demand business model is based on this definition. To support the on-demand model, IBM uses the e-business framework shown in Figure 1.2.



**Figure 1.2** The IBM e-business framework

In Figure 1.2 the dotted line divides the logical concepts at the top with the physical implementation at the bottom. Conceptually, the IBM e-business framework is based on the on-demand business model operating environment, which has four essential characteristics: It is integrated, open, virtualized, and autonomic. These characteristics are explained later in this section.

The area below the dotted line illustrates how this environment is implemented by the suite of IBM software products.

- Rational is the “build” software portfolio; it is used to develop software.
- Information Management (where DB2 belongs) and WebSphere are the “run” software portfolios; they store and manipulate your data and manage your applications.
- Tivoli is the “manage” software portfolio; it integrates, provides security, and manages your overall systems.
- Lotus is the “collaborate” software portfolio used for integration, messaging, and collaboration across all the other software portfolios.

The IBM DB2 software plays a critical role in the on-demand operating environment. All elements of the Information Management portfolio, including DB2, are developed with the four essential characteristics of the on-demand business model in mind.

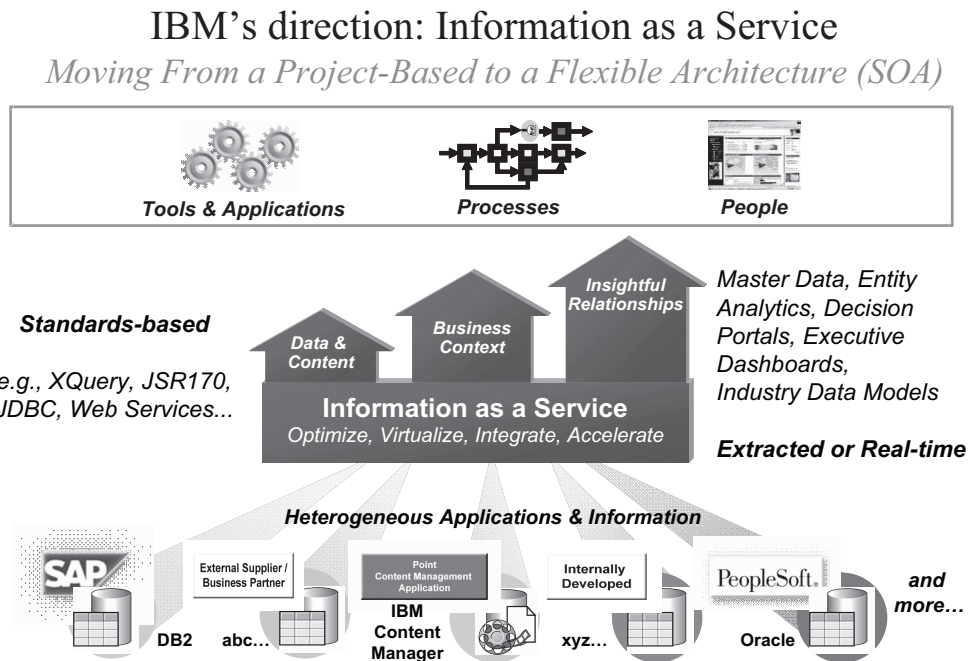
- **Integrated:** DB2 software has built-in support for both Microsoft and Java development environments. It is also integrated into WebSphere, Tivoli, Lotus, and Rational products. In addition, the DB2 family has cross-platform capabilities and can be integrated natively with Web services and message-queuing technologies. It also provides support for heterogeneous data sources for both structured and unstructured information, including pureXML support.
- **Open:** DB2 software allows for different technologies to connect and integrate by following standards. Thus, it provides strong support for the Linux operating system and for Java, XML, Web services, grid computing, and other major industry applications.
- **Virtualized:** Grid computing technology, a type of distributed computing, collects and shares resources in a large network to simulate one large, virtual computer. DB2 software products support grid computing technology through federation and integration technologies. Both of these are discussed in more detail later in this chapter.
- **Autonomic:** An autonomic computing system manages, repairs, and protects itself. As systems become more complex, autonomic computing systems will become essential. DB2 provides self-tuning capabilities, dynamic adjustment and tuning, simple and silent installation processes, and integration with Tivoli for system security and management.

The bottom of Figure 1.2 shows the operating systems in which the IBM software suite can operate: Linux, UNIX, Windows, i5/OS, and z/OS. Below that, the servers, storage, and network represent the actual hardware used to support the framework.

An on-demand business depends on having information available on demand, whenever it is needed, by people, tools, or applications. Information On Demand is discussed in the next section.

## 1.2.2 Information On Demand

Information On Demand, as its name implies, is making information available whenever people, tools, or applications demand or request it. This can be made possible by providing information as a service. IBM commonly uses the illustration in Figure 1.3 to explain what “information as a service” means. Let’s use the following example to explain this concept in a more interesting way. Assume you are the general manager of a supermarket, and your main goal is to make this business profitable. To accomplish this, you must make good decisions, such as how to display items on shelves so that they sell more. In order to make good decisions, you need to have up-to-date, reliable information.



**Figure 1.3** Information as a service

As depicted at the bottom of Figure 1.3, many businesses today have a large number of heterogeneous sources of information. For this particular example let's assume your suppliers use SAP and DB2, your sales department uses an internally developed application, your smaller supermarket clients use Peoplesoft, and Oracle, and so on. Thus, you see several heterogeneous applications with semi-raw data, which will only be valuable to you if you can integrate them all. In order to integrate the data, it needs to be provided as a service, and this is possible through the

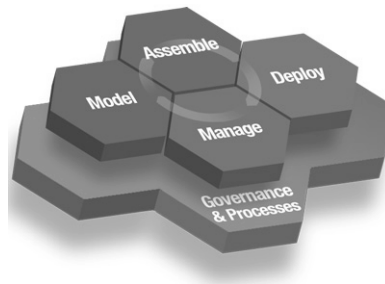
use of standards such as JDBC and ODBC, and wrapping each of these applications as a Web service. Once the data are integrated, you may come up with decisions that might not have been logical otherwise, such as putting beer and diapers in the same aisle in order to sell more of both products.

With the data integrated you can further massage it to perform some additional analysis and get insightful relationships. This further massaging of the data can be performed by other software, such as entity analytics, master data, and so on as shown on the right side of the figure. Finally, this integrated data can be passed to other processes, tools and applications, and people for further analysis.

### 1.2.3 Service-Oriented Architecture

Service-Oriented Architecture (SOA), as its name implies, is an architecture based on services—mainly Web services. SOA is not a product, but a methodology, a way to design systems that allow for integration, flexibility, loosely coupled components, and greater code reuse. With this architecture, business activities are treated as services that can be accessed on demand through the network.

Figure 1.4, which is also used in many IBM presentations, depicts the SOA lifecycle. It consists of four iterative steps or stages—Model, Assemble, Deploy, Manage—and a fifth step that provides guidance throughout the cycle: Governance & Processes.



**Figure 1.4** The SOA Lifecycle

A more detailed explanation of each stage in the SOA lifecycle is provided in Table 1.2.

**Table 1.2** The SOA Lifecycle Stages

| SOA stage | Description   | IBM Tools That Can Be Used  |
|-----------|---|---|
| Model     | This stage is used to model and optimize your business processes. It is also used to determine the kinds of services needed and the type of data these services would access. | WebSphere Business Integration Modeler<br>Rational Software Architect |



**Table 1.2** The SOA Lifecycle Stages (Continued)

| SOA stage  | Description   | IBM Tools That Can Be Used   |
|------------|---|--|
| Assemble   | This stage is about building new services and/or reusing existing ones, and assembling them to form composite applications.   | WebSphere Integration Developer<br>Rational Application Developer  |
| Deploy     | In this stage your services and applications are deployed into a secure environment that integrates people, processes, and information within your business.  | WebSphere Process Server<br>WebSphere Message Broker<br>WebSphere Partner Gateway<br>WebSphere Portal<br>WebSphere Everyplace Deployment<br>Workplace Collaboration Services<br>WebSphere Information Integrator<br>WebSphere Application Server |
| Manage     | In this stage, you need to manage and monitor your system, find and correct inefficiencies and problems, deal with security, quality of service, and general system administration.   | WebSphere Business Monitor<br>Tivoli Composite Application Manager for SOA<br>Tivoli Identity Manager  |
| Governance | Governance underpins all the lifecycle stages. It ensures that all the services from inside and outside the organization are controlled so the system does not spin out of control. Governance provides both direction and control. | N/A  |

### 1.2.4 Web Services

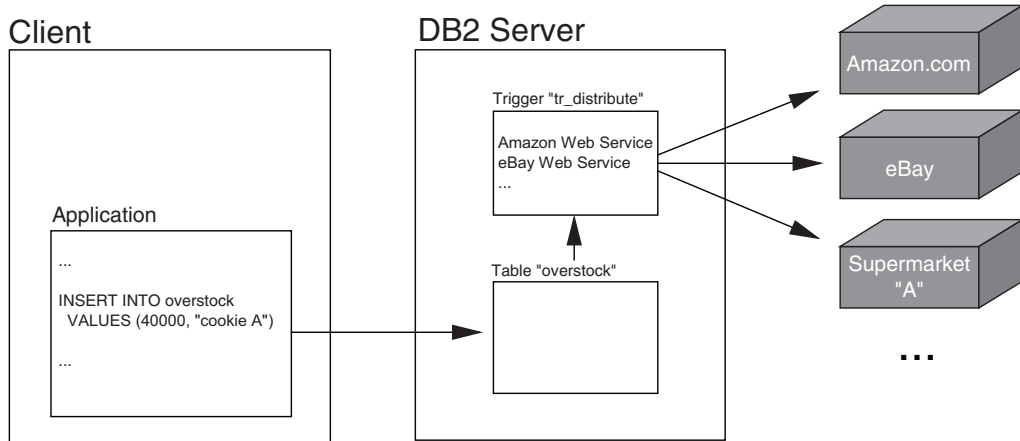
A Web service, as its name implies, is a service made available through the Web. A more formal, but still simple definition states that a Web service is a way for an application to call a function over the network; however, there is no need to know

- The location where this function will be executed
- The platform in which the function will run (for example Linux, UNIX, Windows, the mainframe, Mac OS/X, etc.)
- The programming language in which the function was created (for example Java, Cobol, C, etc.)

Web services are powerful because they allow businesses to exchange information with minimal or no human intervention. Let's go back to the supermarket example to see the power of Web services in a more realistic scenario:

Let's say you order 100,000 cookies from a supplier, expecting all of them to be sold in one month. After the month passes only 60,000 are sold, so you are left with 40,000. Because these

are cookies of a special kind, they will spoil in two weeks. You need to act fast and sell them to other smaller supermarkets or Internet companies such as Amazon.com or eBay. You can grab the phone and spend an entire morning calling each of the smaller supermarket clients, offering them as many cookies as they would want to buy from you; or you could take a more “technical” approach and develop a simple application that would do this for you automatically. Assuming each of these smaller supermarket clients provide Web services, you could develop an application (in any programming language) that allows you to SQL insert overstocked items, such as the 40,000 cookies, into a DB2 database table *overstock*. You could then define a trigger on this table which invokes a DB2 stored procedure (more about triggers and stored procedures in Chapter 7, Working with Database Objects) that could consume Web services provided by the Internet companies or the smaller supermarket clients. This scenario is depicted in Figure 1.5.



**Figure 1.5** Using a Web service

As you can see from Figure 1.5, the simple act of inserting 40,000 cookies through your application into the table *overstock* in the DB2 server allows the systems of many smaller supermarkets and Internet companies, through the use of their Web services, to make the cookies available on their systems quickly, opening new sales channels. In Figure 1.5, DB2 is behaving as a Web service consumer, because it is using or “consuming” the Web services, while the smaller supermarket clients and Internet companies are behaving as the Web service providers, because they are making these Web services available for others to use. For simplicity purposes, we have omitted in Figure 1.5 the call to a stored procedure. This scenario shows the power of Web services: business-to-business exchange of information using applications. There is no need for human intervention. DB2 and Web services will be discussed in more detail in Chapter 10, Mastering the DB2 pureXML Support.

## 1.2.5 XML

XML stands for eXtensible Markup Language. XML's popularity and use has grown exponentially in the past few years, as it is a core component of many new technologies. The easiest way to understand how XML works is by comparing it to HTML, given that many people today are familiar with HTML. Let's take a look at the following line in an HTML document:

```
<b>Raul</b>
```

In the above line, the tag **<b>** indicates the way you would like to *display* the text, in this case, **Raul** in bold. Now Let's take a look at the following line in an XML document:

```
<name>Raul</name>
```

In the above line, the tag **<name>** *describes* the text **Raul**. The tag is saying that **Raul** is in fact a name. See the difference? In HTML, tags are used to indicate how you would like to display the data; in XML, tags are used to actually describe the data. Table 1.3 describes the characteristics of XML.

**Table 1.3** Characteristics of XML

| XML Characteristic                    | Description  |
|---------------------------------------|--|
| Flexible                              | XML is a flexible language because it is easy to modify or adapt. XML is based on a hierarchical model, which is most appropriate to store unstructured types of information such as financial information, life sciences information (for example Genome, DNA), and so on.  |
| Easy to extend                        | XML is easy to extend; that is, you can create your own tags. For example, in addition to the <b>&lt;name&gt;</b> tag in the example above, you could create new tags such as <b>&lt;address&gt;</b> , <b>&lt;email&gt;</b> , <b>&lt;phone&gt;</b> , and so on. This means you can create your own language or protocol based on XML.  |
| Describes itself                      | XML can describe itself; another document called an XML Schema (which itself is an XML document) is used to provide rules and descriptions as to what each of the tags in a document mean and restrict the type of data the tags can contain. An older method, but still widely used today, is to use DTD documents. In the above example, an XML Schema or DTD document can indicate that the tag <b>&lt;name&gt;</b> can only be used to store characters. |
| Can be transformed to other formats   | XML can be transformed to other formats like HTML, using Extensible Stylesheet Language Transformations (XSLT), a language used for the transformation of XML documents.   |
| Independent of the platform or vendor | XML is independent of the platform or vendor; after all, XML documents can be stored in text files containing tags. Text documents are supported everywhere.   |
| Easy to share                         | XML is easy to share with other applications, businesses, and processes given that it can be stored as a text document. Because it is easy to share, it's appropriate as the core of Web services.   |

XML is also at the core of Web 2.0 development technologies. Web 2.0, as defined in Wikipedia.org “refers to a perceived second generation of web-based communities and hosted services—such as social-networking sites, wikis, and folksonomies—which facilitate collaboration and sharing between users”. Wikis, blogs, mash-ups, RSS or atom feeds, and so on, which are part of Web 2.0 development technologies, are all based on or related to XML. This makes DB2 9.5 the ideal data server platform for Web 2.0 development. Table 1.4 describes the different technologies that are part of Web 2.0.

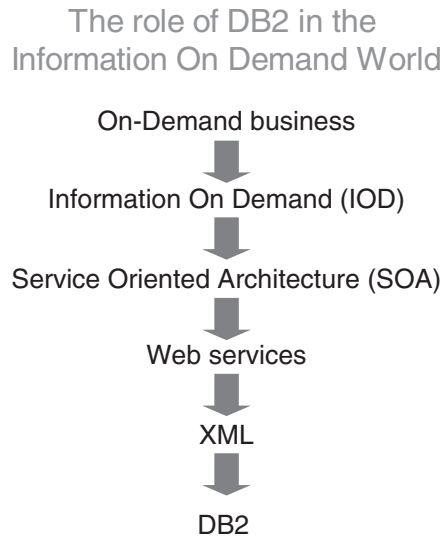
XML is discussed in more detail in Chapter 10, Mastering the DB2 pureXML Support.

**Table 1.4** Web 2.0 Technologies

| <b>Web 2.0 Technology</b> | <b>Description</b>  |
|---------------------------|---|
| AJAX                      | Asynchronous Javascript and XML: A technique for creating interactive Web applications, which can update parts of a Web page without refreshing the entire page.  |
| AdSense                   | This is an advertisement serving program where Web site owners can enable text, image, and video advertisement on their site, and ads will appear based on the Web site content, the user’s geographic location, and other factors. |
| Blog                      | Also known as a Web log, this is a Web-based publication of periodic articles in journal style displayed in chronological order. It is often used to provide personal commentaries on a subject or as personal online diaries.      |
| Mashup                    | This is a Web application hybrid, that is, a Web site or application that is derived by aggregating components or services from other sources such as RSS or atom feeds, and JavaScript.  |
| REST                      | Representational State Transfer: An architectural style for distributed hypermedia systems like the World Wide Web.   |
| RSS and Atom              | RSS (Really Simple Syndication) and Atom are XML file formats for Web syndication, which provides a way to distribute information.  |
| Tagging                   | Bookmarks that provide a way to attach keywords to pages or images on the Web, helping categorize and making things easier to find (i.e., meta-data).   |
| Wiki                      | A type of Web site that allows for community authoring (add/delete/edit content).   |

### 1.2.6 DB2 and the IBM Strategy

Now that you understand the key concepts of the overall IBM strategy, you may be asking yourself, how do these concepts relate to each other? Where does DB2 fit in the overall strategy? To answer these questions, let’s take a look at Figure 1.6.



**Figure 1.6** The role of DB2 in the IBM strategy

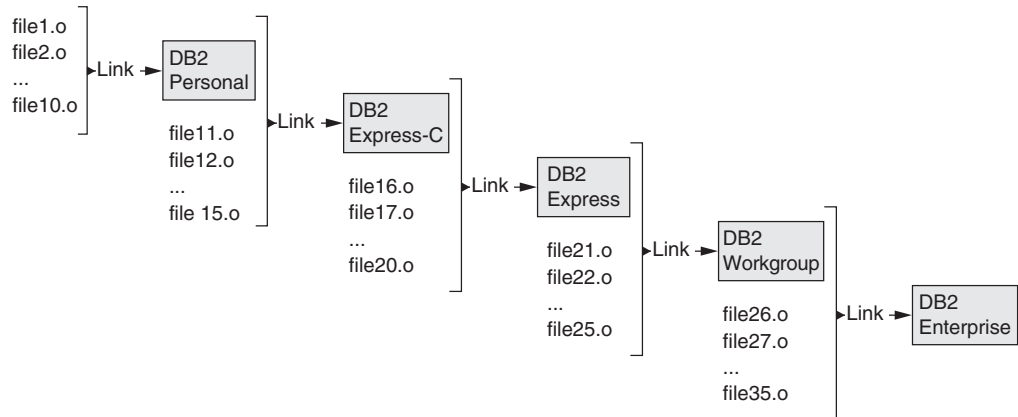
As shown in Figure 1.6, an on-demand business depends on Information On Demand, which depends on Service-Oriented Architecture, which depends on Web services, which depends on XML. The figure shows why XML is so important: XML is at the base of all of these concepts; without it, they would be hard to implement.

XML documents need to be stored in a safe place that allows for storing, managing, and retrieving large volumes of these documents. Persistency, scalability, security, recovery, and good performance are all important features to consider when selecting a repository for your XML documents. All of these features have already been provided for many years in databases; therefore, a database is probably the best repository for XML documents. DB2 has been providing support to work with XML documents for several years. Starting in version 9, as described earlier in this chapter, pureXML technology is made available, which greatly enhances performance by internally storing XML documents in a parsed-hierarchical manner, as a tree. Thus, in summary, the role of DB2 in the Information On Demand world is to be *the* repository of XML documents. This is why DB2 is shown at the bottom of the figure, supporting all the other technologies.

### 1.3 DB2 EDITIONS

DB2 for Linux, UNIX, and Windows (sometimes referred to as **LUW**) is developed using the C/C++ language. More than 90 percent of the code is common among these platforms. The remaining code is unique to take full advantage of the underlying platform architecture; however, the database functionality on all of these platforms is the same.

Like any other C/C++ application, DB2 is written in separate modules—.c/.C source files—that have been separately compiled to obtain object files (.o files). These object files are later linked to obtain an executable file. Figure 1.7 shows a *simplified* view of how each edition is built.



**Figure 1.7** How DB2 editions build on top of each other

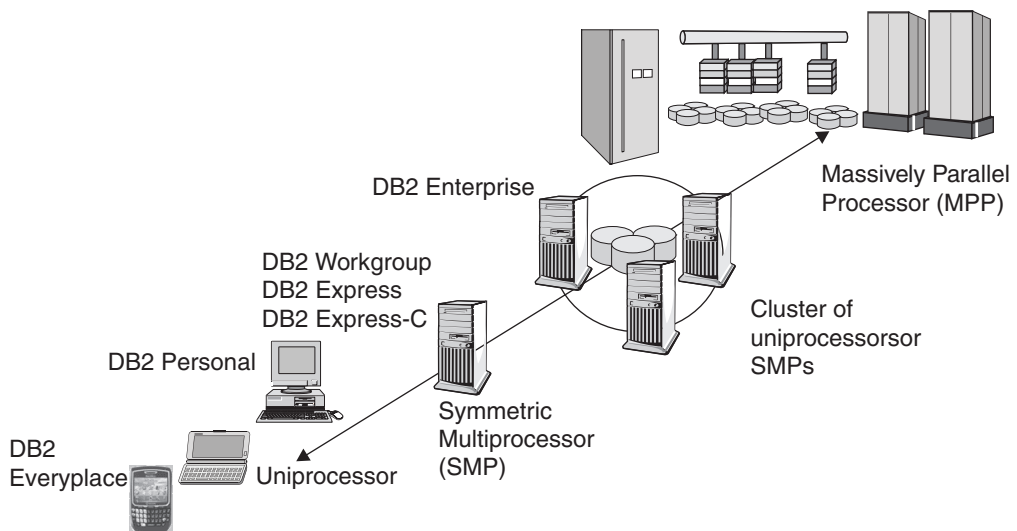
As you can see in Figure 1.7, each edition (other than DB2 Everyplace, which is not shown in the figure) builds on top of the other by linking modules or object files that contain additional functionality. The core of the DB2 code is common across all editions, which greatly benefits application development. For example, if you are developing an application for DB2 Personal, this application will also work on DB2 Express-C, DB2 Express, DB2 Workgroup, and DB2 Enterprise on any of the supported platforms.

From the above explanation, it should be clear that the DB2 LUW editions are mainly packaging and licensing distinctions that let you choose the appropriate features or functions you need for the right price. The underlying technology is always DB2, so choose the appropriate edition based on the features and functions you need.

**NOTE** DB2 for z/OS, DB2 for VM/VSE, and DB2 for i5/OS use a different code base from that used by DB2 LUW. Note, however, that the Linux operating system extends across all of IBM's servers: System x, System p, System i, and System z. DB2 for Linux on all of these server platforms is the same. Thus, DB2 for Linux on System z uses the same code base and is licensed in the same way as DB2 for Linux on a System x (Intel) platform.

**NOTE** Refer to Appendix C, IBM Servers, for a description of the System x, p, i, and z servers.

Figure 1.8 illustrates the different editions and the types of servers they typically run on. By default, DB2 takes advantage of all the processing power it is given. The figure also shows that DB2 is a scalable product. With the exception of DB2 Everyplace, the functions, features, and benefits of an edition shown on the bottom of the figure are included in each subsequent edition as you move up the figure. The following sections provide more detail on the functionality of each edition.



**Figure 1.8** DB2 editions

### 1.3.1 DB2 Everyplace Edition

As its name implies, the DB2 Everyplace edition (DB2 Everyplace) can run anywhere, anytime, and in all kinds of small devices, such as personal digital assistants (PDAs), handheld computers, embedded devices, and laptops. DB2 Everyplace, though only about 350K in size, is a true relational database that uses a subset of the DB2 server SQL functionality. If you know how to code an application for a DB2 server edition, you know how to code for DB2 Everyplace. Applications can be developed using ODBC, CLI, JDBC, and .NET.

Typically, users of DB2 Everyplace store information in the mobile database and later replicate it to a back-end database server using the DB2 Everyplace Sync Server installed on another machine.

This edition supports the following operating systems that run on mobile devices:

- Embedded Linux
- Linux distributions
- J2ME devices
- Palm OS
- QNX Neutrino
- Symbian OS
- Microsoft Windows 32-bit operating systems
- Windows Mobile for Pocket PC, Windows CE.NET

DB2 Everyplace can be licensed as a fully synchronized environment or as a standalone embedded database.

### 1.3.2 DB2 Personal Edition

The DB2 Personal Edition (DB2 Personal) is a complete product for a single user. It has all the functionality of a database server, including the administration graphical tools. It also comes with the Spatial Extender, and the Net Search Extender. While this edition can also be used as a client to connect to other DB2 servers, it does not support database incoming connections from other computers. Only Windows and Linux operating systems, which are the most commonly used platforms in *personal* computers, support DB2 Personal.

Figure 1.9 shows DB2 Personal installed on Machine 2. The local DB2 client (the client component of Machine 2) can connect to a database in the DB2 Personal server on Machine 2, but the remote DB2 client in Machine 1 cannot connect to a database in the server on Machine 2 because DB2 Personal does not accept remote (inbound) connections. The figure also shows DB2 Personal on Machine 2 as the remote client to other DB2 server editions installed on machines 3, 4, and 5.

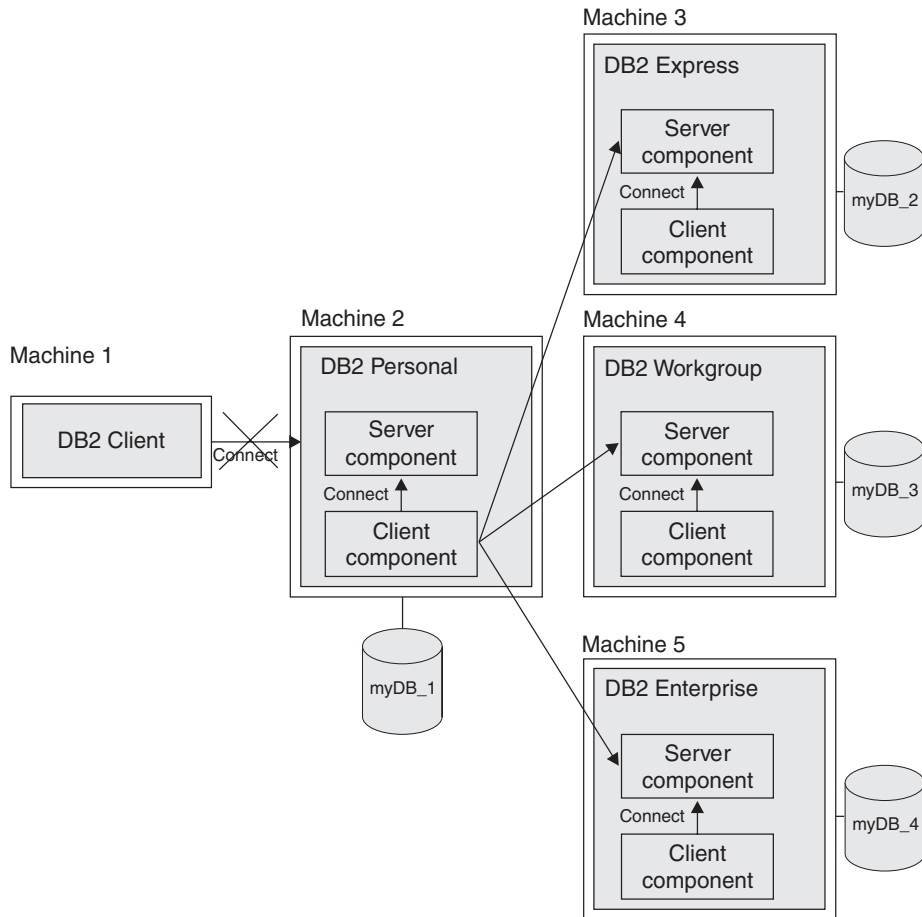
**NOTE** A DB2 data server is considered a server when it can accept inbound client connections for data retrieval purposes. Hence, DB2 Personal is not considered a DB2 server.

### 1.3.3 DB2 Express-C

DB2 Express-C is a version of the DB2 Express edition for the community. Businesses developing an application that needs to connect to a data server can use DB2 Express-C for free. Note that we refer to it as a version: It is not an edition of DB2 because it is free. Also note that the core code of DB2 Express-C is the same as the other DB2 editions as shown in Figure 1.7. DB2



Express-C can be used in production or in a commercial setting. In addition, businesses can embed and distribute DB2 Express-C as part of their application also for free. DB2 Express-C does not impose limits on the database size, the number of instances per server, or the number of users.



**Figure 1.9** DB2 Personal as a (local) server and a remote client

The supported operating systems for this version are Windows and Linux (on Intel and AMD), running on a system with any amount of processors and memory, however the total resource utilization by all instances of DB2 Express-C on a given system cannot exceed 2GB of RAM and 2 processor cores on 32- or 64-bit SMP hardware computers. An optional renewable 12-month subscription license can be purchased for DB2 Express-C to obtain IBM DB2 Technical support (24/7), and also support for the SQL replication and the High Availability and Disaster Recovery (HADR) feature. This subscription edition of DB2 Express-C is licensed to run on up to two dual-core processors and no more than 4GB of memory. Without this license, the product is free but without the mentioned features, and with support provided only through a free community-based online forum. For further details, refer to the DB2 Express Web site: [www.ibm.com/db2/express](http://www.ibm.com/db2/express).

### **1.3.4 DB2 Express Edition**

The DB2 Express Edition (DB2 Express), a low-cost, full-function data server, is ideal for a business that requires a database, but has minimal in-house database skills. This edition provides the same support as DB2 Workgroup, but it also features simple installation, enhanced self-management, and other ease-of-use features. Businesses developing applications that require a database can embed DB2 Express as part of their solution.

The supported operating systems for this edition are Windows and Linux (on Intel and AMD), running at most on 4GB of RAM and two CPUs, which can be dual core on 32- or 64-bit SMP hardware computers.

DB2 Express can be licensed per user (ideal for applications with just a few users) or per processor value unit (ideal for applications with many users, like a Web application).

DB2 Express can be extended by purchasing several Feature Packs. A Feature Pack includes several features, and can be purchased in the same manner as a DB2 data server license. There are available Feature Packs for workload management, performance optimization, high availability, pureXML, and homogeneous federation.

### **1.3.5 DB2 Workgroup Server Edition**

The DB2 Workgroup Server Edition (DB2 Workgroup) is a full-function data server designed for deployment in a department or small business environment. Linux, UNIX, and Windows platforms support DB2 Workgroup running on at most 16GB of RAM and four CPU servers with a 32- or 64-bit operating system. It can be licensed per user or per processor value unit. All the Feature Packs mentioned for DB2 Express apply also to DB2 Workgroup.

### **1.3.6 DB2 Enterprise Server Edition**

The DB2 Enterprise Server Edition (DB2 Enterprise) is the most complete data server offering. It provides unparalleled scalability, accessibility, and extensibility features, and is the edition of choice for most enterprises. Some of the features included in the Feature Packs available with DB2 Express or DB2 Workgroup are free with DB2 Enterprise. DB2 Enterprise has its own Feature Packs as well. For example, to use the database partitioning feature (DPF) you need to purchase the Feature Pack “Database Partitioning Feature for DB2 Enterprise.” DPF allows you to partition your data within a single server or across multiple servers running the same operating system. This means that your databases can grow to sizes that are limited only by the number of servers available.

DB2 Enterprise can be used in SMP systems, and DB2 Enterprise with DPF can be used in either SMP or clustered server systems. The supported operating systems are Linux, UNIX, and Windows.

Table 1.5 obtained at the time of publication from the DB2 for Linux, UNIX, and Windows main Web page at <http://www-306.ibm.com/software/data/db2/9/>, provides a comparison of the DB2 server editions.

**Table 1.5** Comparing the DB2 Editions**Find the DB2 9 edition that meets your needs**

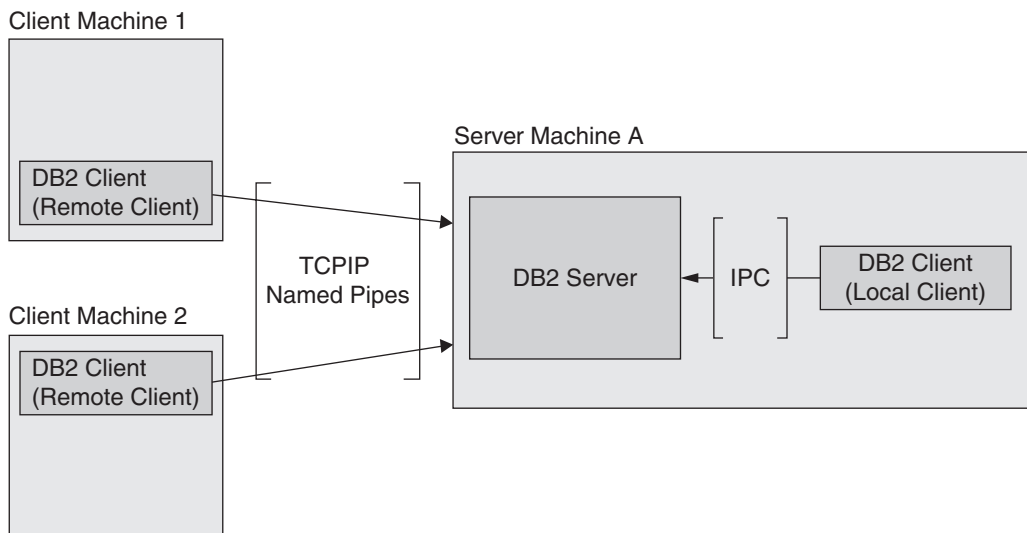
|                 | <b>DB2 Express</b>   | <b>DB2 Workgroup</b>   | <b>DB2 Enterprise</b>   |
|-----------------|--|--|---|
| Function        | DB2 Express is a full-function hybrid data server, which provides very attractive entry-level pricing.   | Includes all of the features of DB2 Express with scalability to larger servers.  | Includes all of the features of DB2 Workgroup plus features required to provide the scalability to handle high user loads and provide 24x7x365 availability, including:<br><br>High Availability Disaster Recovery (HADR) |
|                 | Simple installation including silent installation capability   |  | Tivoli System Automation  |
|                 | Self managing  |  | Table Partitioning  |
|                 | Optimized interfaces and tools for application developers  |  | Multi-dimensional data clustering   |
|                 | Supports wide array of development paradigms   |  | Materialized Query Tables   |
|                 | Minimal disk space requirements  |  | Full intra-query parallelism  |
|                 | Worldwide 24x7 Service and Support   |  | Connection concentrator   |
| Customizable    | Expandable with pureXML and optional enterprise class features to preserve and improve performance, workload management, and high availability | Expandable with pureXML and optional enterprise class features to preserve and improve performance, workload management, and high availability | Expandable with pureXML and advanced features like storage optimization, performance optimization, advanced access control, scale-out clustering, geodetic data, and more   |
| Scalable        | 2 CPUs / 4GB RAM maximum (may run on machines with more than 4GB)  | 4 CPUs / 16GB RAM maximum  | Unlimited   |
| Platforms       | Linux, Solaris x86 and Windows   | Linux, UNIX, and Windows   | Linux, UNIX, and Windows  |
| Pricing Metrics | Authorized User (Min. 5 per server), or Per Processor  | Authorized User (Min. 5 per Server), or Per Processor  | (Authorized User (Min. 25 per CPU) or Per Processor   |

## 1.4 DB2 CLIENTS

To connect from a client to a DB2 data server database, you usually need to install DB2 client software on the client machine. This isn't always required; for example, this isn't necessary for a JDBC application using the Type 4 driver running on the client machine. We provide more detail about connectivity scenarios in Chapter 6, Configuring Client and Server Connectivity.

A DB2 client installed on a different machine than the DB2 server is known as a **remote client**. A remote client can establish communication to the server using TCP/IP or Named Pipes (Windows only).

If the DB2 client is installed on the same machine as a DB2 server, then it is known as a **local client**, and it connects to the server using inter-process communication (IPC). Note that since all DB2 servers come with a local client component, you don't need to install the DB2 client separately after installing a DB2 server. Figure 1.10 shows local and remote clients. Client Machine 1 and 2 are remote clients with the DB2 client code installed and are accessing Server Machine A, which has a DB2 server installed. The DB2 server has a client component that is the local client.



**Figure 1.10** Local and remote clients

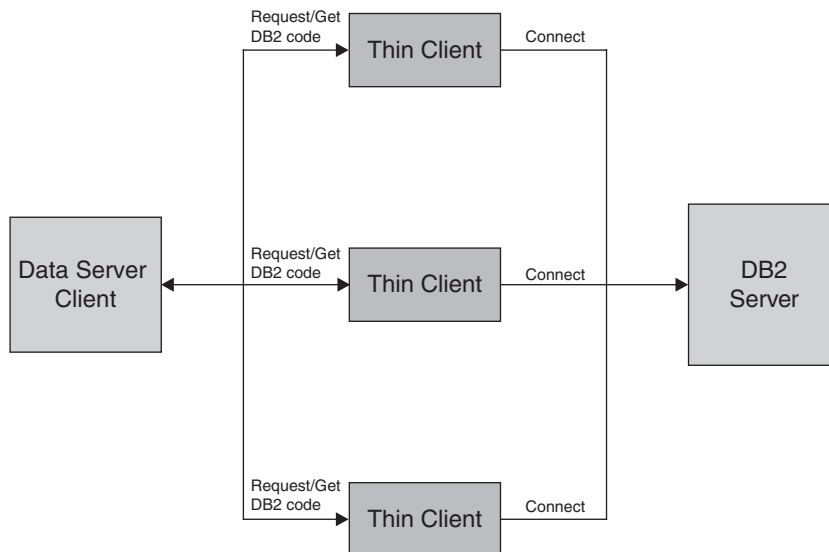
**V9.5** There are three types of clients in DB2:

- Thin client
- IBM Data Server Runtime client
- IBM Data Server client

and there are two drivers:

- IBM Data Server Driver for JDBC and SQLJ
- IBM Data Server Driver for ODBC, CLI, and .NET (DSDRIVER)

A **thin client**, also known as a **dumb terminal**, is a machine with no operating system or DB2 client code. A thin client has to first get from another machine all the libraries and modules it requires to fulfill a request. Figure 1.11 shows an example of thin clients, where the thin client first requests, and then gets the libraries and modules it needs and the code of the DB2 client from the Data Server Client machine on the left side of the figure. This can only work on Windows platforms and will not work if the machine on the left side of the figure was an IBM Data Server Runtime client. It must be an IBM Data Server client.

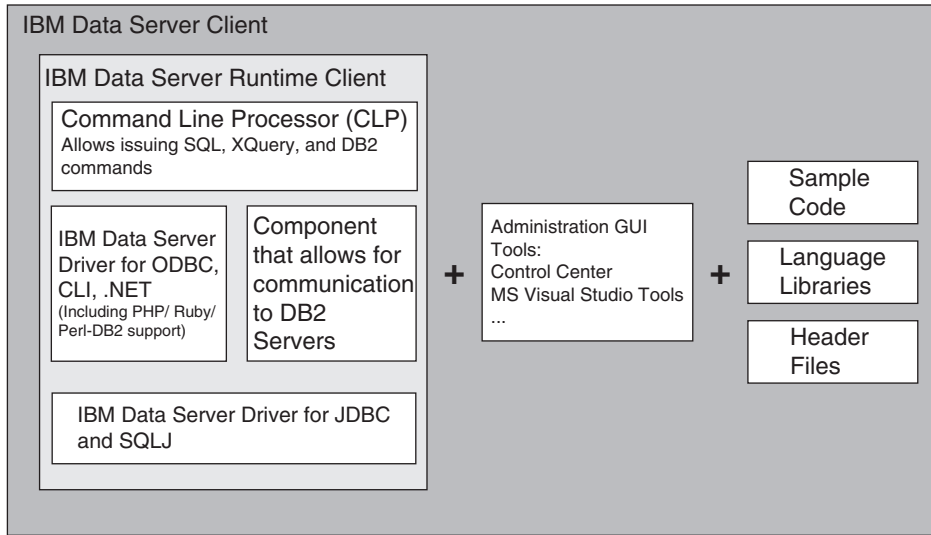


**Figure 1.11** A thin client

The **IBM Data Server Runtime client** has the minimum basic requirements to connect to DB2 databases and to provide basic client functionality. It includes the drivers for ODBC, CLI, .NET, PHP, Ruby, Perl-DB2, JDBC, and SQLJ. There are also IBM Data Server runtime client merge modules for Windows, which allow you to add IBM Data Server runtime client functionality to any product that uses the Windows installer. The **IBM Data Server client** comes with everything the IBM Data Server Runtime client has, plus graphical administration tools and development tools and libraries.

Prior to DB2 9, the drivers for ODBC, CLI, JDBC, and SQLJ could only be obtained by downloading and installing a DB2 client. With DB2 9, the **IBM Data Server Driver for JDBC and SQLJ** was introduced as a separate downloadable image. Now with DB2 9.5, the **IBM Data Server Driver for ODBC, CLI and .NET (DSDRIVER)** client is introduced to deploy drivers for applications to access remote IBM data servers. It includes support for PHP, Ruby, and

Perl-DB2 as well. Both drivers are free and can be redistributed. Figure 1.12 illustrates the client progression.



**Figure 1.12** Client progression

**NOTE** The DB2 9 Runtime Client has been renamed IBM Data Server Runtime Client starting in DB2 9.5. The DB2 9 Client has been renamed IBM Data Server Client starting in DB2 9.5. These name changes reflect changes in the client code, as these clients not only can connect to DB2 data servers, but can also connect to Informix data servers.

## 1.5 TRY-AND-BUY VERSIONS

A DB2 9.5 Try-and-Buy version allows you to try DB2 for an evaluation period of 90 days. DB2 will stop working after the 90 days unless you buy and apply the full license. Other than this time limitation, the Try-and-Buy version contains all the features of a fully licensed version. During or after the Try-and-Buy period you can buy a permanent license by calling 1-800-IBM-SERV. An IBM representative will direct you to the License Key Center. Once you receive the license file, you can use it to upgrade the Try-and-Buy version to the fully licensed product level without reinstalling it. Use the following command from a command-line window to install the license file:

```
db2licm -a file_name
```

where **file\_name** stands for the name of the license file, which normally has a .lic extension.

Issuing the **db2licm -l** command lists the software products installed on your machine and the current license. After adding a full license, the Expiry Date field will have a value of *Permanent*. Chapter 3, Installing DB2, discusses more about the **db2licm** command and the License Center.

**NOTE** Licensing policies, as well as the evaluation period, are subject to change. The information in this section is accurate as of the time of publication.

## 1.6 HOST CONNECTIVITY

**DB2 Connect** is a software product containing the license files required to communicate from a DB2 distributed client, also known as the **DRDA Application Requester**, to a host DB2 server, a **DRDA Application Server**. (DRDA—Distributed Relational Database Architecture—is the standard that defines formats and protocols for providing transparent access to remote data.) Host DB2 servers include DB2 for z/OS and OS/390, DB2 for VM/VSE, and DB2 for i5/OS. DB2 Connect also includes an IBM Data Server Runtime client.

**NOTE** DB2 Connect software is only required when connecting from DB2 LUW to a host DB2 server, such as DB2 for z/OS; it is *not* required in the other direction, for example, when DB2 for z/OS behaves as the client, and DB2 LUW is the server.

DB2 Connect comes in two main editions. Other editions not listed below refer to licensing options:

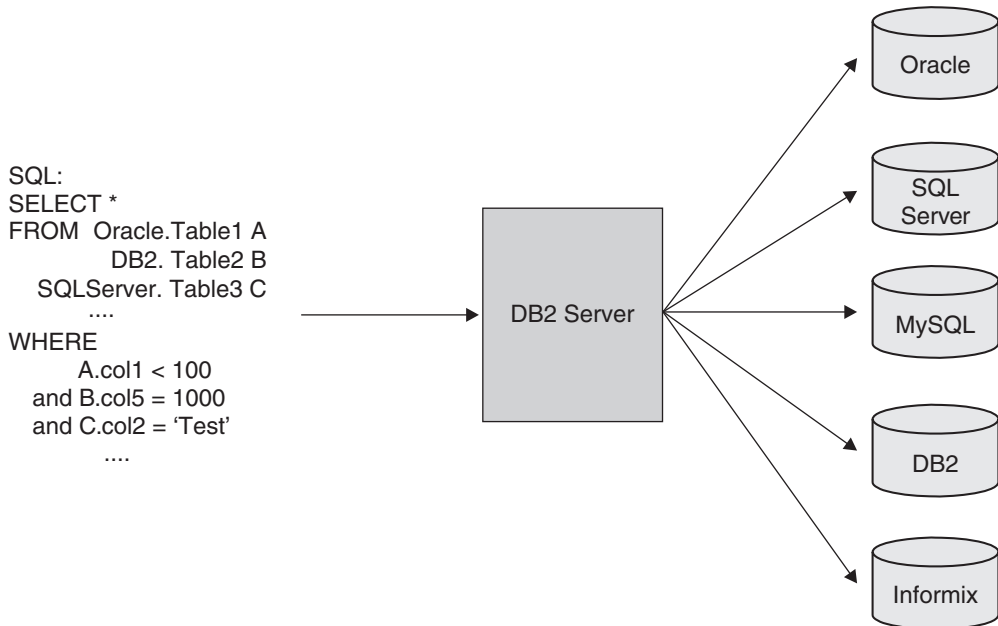
- The Personal Edition supports the direct connection of one DB2 client to multiple host DB2 servers.
- The Enterprise Edition supports the connection of multiple DB2 clients to multiple host DB2 servers via a DB2 Connect server.

## 1.7 FEDERATION SUPPORT

**Federation** allows you to query and manipulate data stored on other servers and in other data servers. When you issue an SQL statement in a federated environment, you may actually be accessing information from multiple databases and potentially multiple data servers (see Figure 1.13).

Federated support is included in DB2 when the other databases being accessed are part of the IBM DB2 family, that is, another DB2 database or an Informix database. For accessing

databases from other vendors, refer to the IBM WebSphere Federation Server product described in Section 1.9.



**Figure 1.13** DB2 federation

## 1.8 REPLICATION SUPPORT

**Replication** lets you propagate data to different servers to keep multiple databases synchronized. This can be useful in situations where a single server is used for day-to-day transaction operations, and where issuing reporting queries at the same time would be costly for performance. By replicating the data to another server, this second server could be used for reporting without disturbing the first server. Figure 1.14 illustrates how the data changes captured at one server are later applied to another (target) server. In the figure, the first box shows the source server and the fourth box shows the target server. The second and third boxes contain the “capture” and “apply” components, respectively.

DB2 has built-in support for replication when source and target databases are part of the IBM family, which includes Informix. For databases from other vendors, such as Oracle or SQL Server, the IBM WebSphere Replication Server software is required.



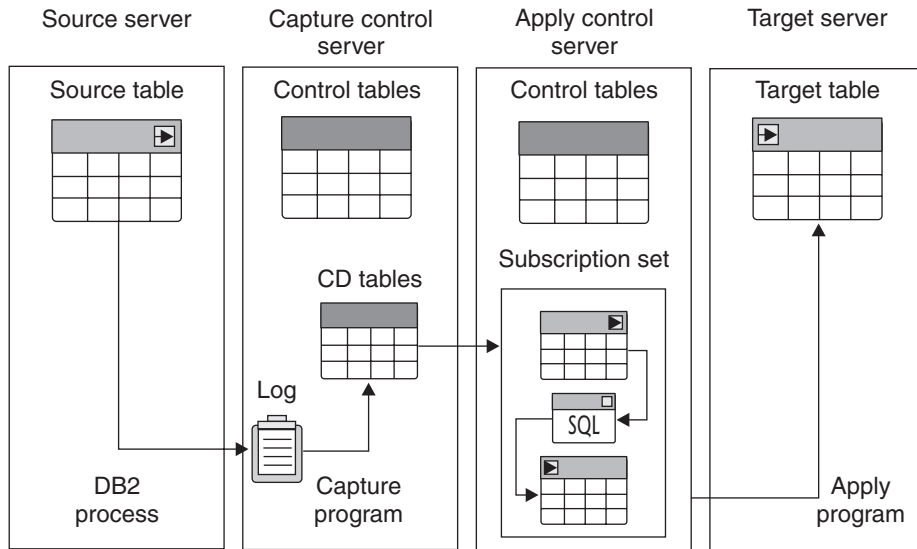


Figure 1.14 DB2 replication environment

## 1.9 IBM WEBSHERE FEDERATION SERVER AND WEBSHERE REPLICATION SERVER

IBM WebSphere **Federation Server** provides federated support by making remote data sources from IBM or different vendors appear as if they were part of the same database. The Federation server uses wrappers to communicate with and retrieve data from those other data sources; it encapsulates any conversions required from the source database and presents them to the target database as tables.

IBM WebSphere **Replication Server** provides replication support from one data server to another, even if they are not part of the IBM family. Replication Server includes both Q-replication and SQL replication (SQL replication is free with DB2 Enterprise).

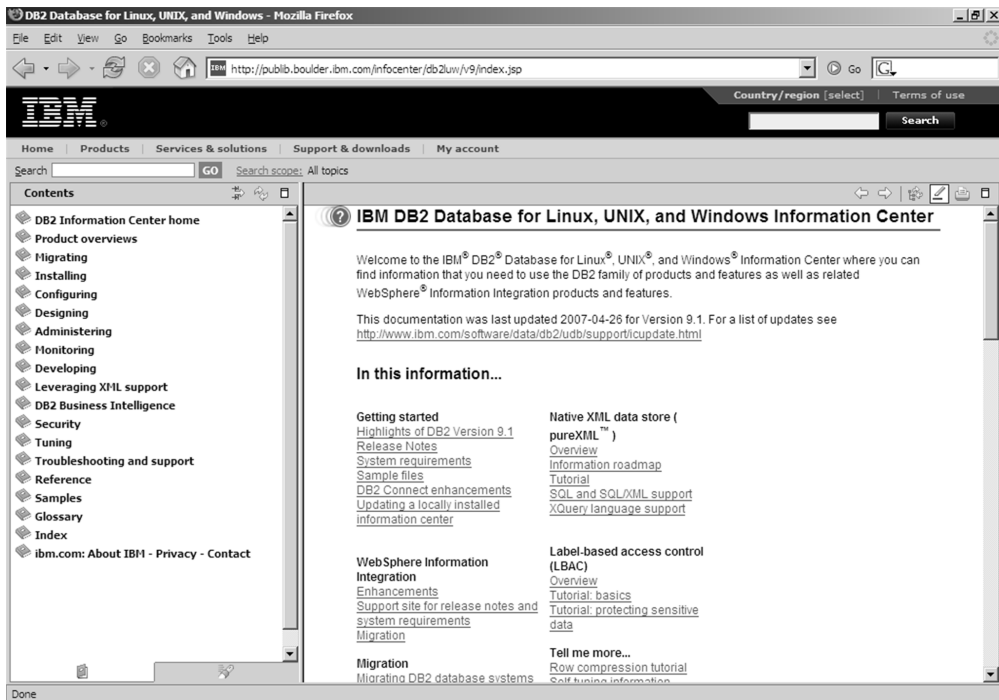
IBM WebSphere Federation and Replication server software were formerly called WebSphere Information Integrator.

## 1.10 SPECIAL PACKAGE OFFERINGS FOR DEVELOPERS

The Database Enterprise Developer's Edition (DEDE) provides developers with a package that includes several DB2 editions and features, Informix products, and DB2 Connect. This package is offered at a reduced price, and is restricted to the development, evaluation, demonstration, and testing of application programs. The package is licensed on a per developer basis.

## 1.11 DB2 SYNTAX DIAGRAM CONVENTIONS

DB2 supports a comprehensive set of statements for data access and manipulation. These statements are documented online in the DB2 Information Center, which gives you access to all information about DB2 as well as major DB2 features and components. It can be conveniently accessed by using a browser, as shown in Figure 1.15. The DB2 Information Center is also available at <http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/index.jsp>.



**Figure 1.15** The DB2 Information Center

You can find the syntax of any commands or statements we introduce in this book in the DB2 Information Center. Understanding how to read the syntax diagrams will help you use the numerous options available with many of the statements.

Syntax diagrams are all read from left to right and top to bottom, following the path of each line. Table 1.6 summarizes a few of the most commonly used symbols in a syntax diagram.

**Table 1.6** Summary of Some Symbols Used in Syntax Diagrams

| Symbol | Description  |
|--------|--|
| >>---  | Indicates the beginning of a syntax diagram.                   |
| --->   | Indicates that the syntax is continued on the next line.       |
| >---   | Indicates that the syntax is continued from the previous line. |
| ---<<  | Indicates the end of a syntax diagram.                         |

When a mandatory field is required, it appears on the horizontal line (the main path) like this.

```
>>-mandatory_field-----<<
```

Optional fields appear below the main path.

```
>>-mandatory_field--+-----+-----<<
                    '-optional_field-'
```

If an optional field appears above the main path, it means that it is the default option.

```
                    .-default_field-.
>>-mandatory_field--+-----+-----<<
                    '-optional_field-'
```

If two or more mandatory choices are available, one of the mandatory choices will appear in the main path, and the rest will appear in a stack. You must choose one of these options.

```
>>-mandatory_field--+mandatory_choice1+-----<<
                    '-mandatory_choice2-'
```

An arrow returning to the left, above the main line, indicates an option can be repeated. In this example, repeated options are separated by one or more blanks.

```
                    .-----
                    v                |
>>-mandatory_field----repeatable_field+-----<<
```

If the repeat arrow contains a comma, you must separate repeated items with a comma.

```
                    .-,-----
                    v                |
>>-mandatory_field----repeatable_field+-----<<
```

You should now feel comfortable reading syntax diagrams in the DB2 documentation. Browse through the DB2 online documentation and review some examples there.

## 1.12 CASE STUDY

John recently graduated from Pennsylvania State University, where he learned DB2 as part of the IBM Academic Initiative program, a program that provides DB2 and other IBM software for free to teach database skills.

While at school, he worked with a DB2 Enterprise Server Edition installed on a System p machine at the university computer lab. He was given SYSADM authority, and was able to see connections from many different clients to all created databases using the **list applications** command. John wanted to develop a Java application using the JDBC Type 4 driver, so he downloaded and installed on his laptop the 90-Day Try-and-Buy version of DB2 Personal Edition. With this edition, he was able to build, test, and run his application and connect to the database created on his laptop. Since the client and data server were both on his laptop, he was dealing with a local client connection.

John wanted to test whether his application would work as a remote client, so he used the client software that comes with DB2 Personal Edition to test his application against the database he had created earlier on the university's System p machine. This also worked, and John was feeling like a DB2 guru.

Eager to show his program to his colleagues, he e-mailed the executable to his friend Peter, who had just bought a new laptop with Microsoft Windows Vista installed. Peter detached the file and tried to run the application against John's database on the University's pSeries server. After spending a few hours trying to figure out why he couldn't, he dropped by John's place. John realized that Peter had to download and install either an IBM Data Server Runtime client, an IBM Data Server client, or just the IBM Data Server JDBC and SQLJ driver, as he needed the JDBC Type 4 driver on his laptop. Given that Peter was neither going to develop a new program nor administer a database, John asked Peter to download just the IBM Data Server JDBC and SQLJ driver from the IBM Web site, and after installation, voilà!, the program successfully ran. Peter then asked John to perform the test from his laptop against the database on John's laptop, but John said it would not work because he had installed DB2 Personal Edition, which is not a database server, and it cannot accept inbound remote client connections.

After the graduation ceremony, John received a PDA as a gift from his dad. His dad had heard John praise DB2, so he had had DB2 Everyplace installed on the PDA. Since John was going to take six months off to travel before looking for a job, John decided to take his PDA with him rather than his laptop. John's account at the university was going to be active for the next eleven months, so while he was traveling he could connect to his "old" database on the System p server and use his application (which he had installed on his PDA) to transfer information about all the friends he met and places he visited during his trip. This way he was able to save information in another place should he lose his PDA.

After his trip, John applied for a position at a medium-sized company in his hometown. To prepare for his interview, John again tested the program he had written against his laptop database,

but the Try-and-Buy evaluation period had expired. John figured that he would always need DB2 on his laptop, so he decided to buy the permanent license. When John received the license file after the purchase, he installed it on his laptop with the command `db2licm -a filename`. Once this problem was resolved, John demonstrated his program during the job interview and was immediately hired.

Company ABC, which was using DB2 Workgroup Server Edition, asked John to modify his application so that it would connect to a DB2 for z/OS host machine. John responded that he did not need to make any modifications, but since DB2 Workgroup Server Edition does not come with the DB2 Connect software component, the company could purchase this software, or get the Database Enterprise Developer Edition (DEDE), as it would be cheaper and has most of the software for all DB2 editions. DEDE is licensed per developer, and this company did not have that many developers, so DEDE was good for them. If Company ABC wanted to use DB2 in a production environment, they would not be able to use DEDE, but would have to buy the appropriate edition and license.

Company ABC was also interested in the DB2 Express-C Edition, because it was free, and one of their applications needed a database to be embedded as part of their solution. Other than John, there were not many skilled DB2 personnel in the company, so DB2 Express-C was also ideal because of its ease-of-use features. John indicated that whilst DB2 Express-C was free, it would not have 24/7 IBM support unless the 12-month subscription license was purchased. Without this license, questions could be posted in a free community-based forum, but community participation was voluntary.

Three months after John was hired, he was promoted. John is well on his way to a very successful career with DB2!

## 1.13 SUMMARY

This chapter introduced DB2 and its history. IBM pioneered relational database management systems and invented SQL. IBM's technology in the relational database management system area has been around for more than twenty years. Its legacy is visible in the Information Management product line that includes database management software, data warehousing and business intelligence software, enterprise content management and discovery software, and information integrator software.

This chapter also discussed the types of clients and servers available with DB2. Although different editions are available to provide varying functionality, the core DB2 product is the same; therefore, application development on any edition will work on all editions. The various editions allow you to choose the functions that best suit your needs.

In addition, the chapter explained federated support, replication, the WebSphere Federation Server, the WebSphere Replication server, and packaging options available for application developers. These packaging options allow developers to obtain DB2 software at a reduced price.

## I.14 REVIEW QUESTIONS

1. IBM added the term “Universal” to the DB2 name with Version 5 of the product. Why was this term added?
2. Which five software brands support the IBM on-demand strategy?
3. Can an application developed for DB2 Personal Edition work with DB2 Enterprise Server Edition?
4. Is DB2 Connect required to connect from a DB2 for z/OS client to a DB2 for Linux, UNIX, and Windows server?
5. What is the term used to describe DB2 9’s unique support for XML?
6. Is IBM WebSphere Federation server required to set up a federation environment between a DB2 server and an Informix server?
7. Provide an example when replication support may be required.
8. Does DB2 for Linux, UNIX, and Windows have one single file that is used for installation in any of these platforms?
9. What does the Database Partitioning Feature (DPF) allow you to do?
10. What should you do when your Try-and-Buy license period expires and you would like to buy a permanent license?
11. Which of the following products is the minimum required on the Windows client to *run* a DB2 application accessing a DB2 database on UNIX?
  - A. DB2 Enterprise
  - B. DB2 Personal Edition
  - C. DB2 Connect
  - D. IBM Data Server Runtime Client
12. Which of the following products is the minimum required to *run* a DB2 application using JDBC Type 4?
  - A. IBM Data Server Client
  - B. IBM Data Server Runtime Client
  - C. Data Server Driver for JDBC and SQLJ
  - D. Data Server Client
13. Which of the following products does not allow applications to connect to its databases from remote clients?
  - A. DB2 Express
  - B. DB2 Personal Edition
  - C. DB2 Enterprise Server Edition
  - D. DB2 Workgroup
14. Which of the following products is *not* considered a DB2 server?
  - A. DB2 Workgroup Server Edition
  - B. DB2 Express-C
  - C. DB2 Personal
  - D. DB2 Enterprise

15. Which of the following DB2 clients provide the DB2 graphical administration tools?
  - A. Thin client
  - B. IBM Data Server client
  - C. Thick client
  - D. IBM Data Server Runtime client
16. Which of the following DB2 editions is the most appropriate for sales personnel who need a basic database to store contacts and business leads made during business trips?
  - A. DB2 Everywhere
  - B. DB2 Satellite Edition
  - C. DB2 Everyplace
  - D. DB2 Personal Edition
17. A software development company would like to test an application that connects to both DB2 for LUW as well as DB2 for z/OS. Which of the following would suit its needs the best?
  - A. DB2 Enterprise Server Edition
  - B. DB2 Workgroup Server Edition
  - C. DB2 Connect Enterprise Edition
  - D. DEDE
18. Which of the following data servers can run on a System z server?
  - A. DB2 for Linux, UNIX, and Windows
  - B. DB2 for iSeries
  - C. DB2 Connect
  - D. IBM Websphere Information Integrator
19. Which of the following products allows ten clients to connect from DB2 LUW to DB2 for z/OS?
  - A. Database Enterprise Developer's Edition
  - B. DB2 Universal Developer's Edition
  - C. DB2 PE
  - D. DB2 LUW
20. Which of the following products can be used to collect, prepare, and analyze your data to allow you to make better business decisions?
  - A. DB2 Content Manager
  - B. DB2 Warehouse Manager
  - C. IBM DB2 WebSphere Federation Server
  - D. DB2 LUW

