IBM DB2 Universal Database Extended Enterprise Edition for UNIX** Quick Beginnings Version 5

Document Number S99H-8314-00

IBM DB2 Universal Database
Extended Enterprise Edition
for UNIX**

Quick Beginnings

Version 5

S99H-8314-00

IBM DB2 Universal Database
Extended Enterprise Edition
for UNIX**

Quick Beginnings

Version 5

S99H-8314-00

Before using this information and the product it supports, be sure to read the general information under Appendix G, “Notices” on page 321.

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

Order publications through your IBM representative or the IBM branch office serving your locality or by calling 1-800-879-2755 in U.S. or 1-800-IBM-4YOU in Canada.

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

Contents

Welcome to DB2 Universal Database

This book describes how to install and use the basic functions of DB2 Universal Database Extended Enterprise Edition.

Part 1 gives an overview of DB2 Universal Database Extended Enterprise Edition.

Part 2 provides planning, installing, and basic configuration information for DB2 Extended Enterprise Edition.

Part 3 describes how to install the DB2 Client Application Enabler on any supported operating system.

Part 4 provides information on configuring communications between DB2 servers and its clients.

Part 5 describes how to use DB2 Universal Database.

Part 6 describes how to perform advanced configuration tasks, such as performing administration tasks using the Client Configuration Assistant, and using the command line processor for setting up server communications. This part also describes deinstalling DB2 products.

Part 7 contains reference information and describes viewing, printing, and ordering manuals.

Conventions

This book uses these highlighting conventions:

- **Boldface type** indicates commands or graphical user interface (GUI) controls such as names of fields, folders, icons, or menu choices.
- *Italics* indicates variables that you should replace with a value. It is also used to indicate file names, directory paths, book titles and to emphasize words.
- *Monospace* indicates examples of text you enter exactly as shown.
- DB2 Extended Enterprise Edition is commonly referred to as DB2 server or DB2 Universal Database throughout this book.

This is a fast path icon. It tells you if you can skip sections to get to the "how to" information. It is shown to indicate where you should go next.

This icon marks a Tip – additional information that can help you complete a task.

Road Map

Use the following table to find the information you need quickly.

If you want to...	**Refer to...**
Understand the relationship between DB2 and related products,	Chapter 1, “About DB2 Universal Database” on page 3.
Plan your network configuration,	Chapter 4, “Planning for Installation” on page 35.
Confirm that your system meets the DB2 disk and software prerequisites,	“Disk Requirements” on page 39 and “Software Requirements” on page 41.
Migrate from a previous release of DB2,	Chapter 7, “Migrating from Previous Versions” on page 91.
Install DB2 Extended Enterprise Edition from a CD-ROM,	Chapter 5, “Installing and Configuring DB2 Universal Database Extended Enterprise Edition for AIX” on page 51 or Chapter 6, “Installing and Configuring DB2 Universal Database Extended Enterprise Edition for Solaris” on page 73.

If you want to...	**Refer to...**
Install the client products,	Chapter 8, “Installing DB2 Clients” on page 111.
Set up communications on the client to access databases,	Chapter 14, “Configuring Client-to-Server Communications Using the CCA” on page 143 or Chapter 15, “Configuring Client-to-Server Communications Using the Command Line Processor” on page 147.
Use the DB2 components,	Chapter 16, “Getting Started with DB2 Universal Database” on page 167.
Create the sample database,	“Creating the Sample Database” on page 168.
Connect to a database,	“Connecting to a Database” on page 169.
Print or view online documentation,	“Printing the PostScript Books” on page 283 or “Viewing Online Books” on page 284.
Set up a DB2 Call Level Interface or ODBC environment to enable ODBC applications such as Lotus Approach to access DB2 data,	“Running CLI/ODBC Programs” on page 191.
Work with the System Administrative (SYSADM) group,	“Working with the System Administrative Group” on page 170.
Use the DB2 Command Center or the DB2 command line processor to enter SQL statements or DB2 commands interactively,	Chapter 20, “Entering DB2 Commands and SQL Statements” on page 207.
Learn about last-minute changes to the product,	The Installation Notes on the CD-ROM or the Release Notes that are installed with the products.

To locate information on other topics, see Appendix A, “Obtaining Information” on page 275 for a complete description of the DB2 library.

Part 1. Introduction to DB2 Universal Database

Chapter 1. About DB2 Universal Database

DB2 is a *relational* database management system that is web-enabled with Java support; scalable from single processors to clusters of symmetric multiprocessors; and multimedia capable with image, audio, video, and text support.

The DB2 products and components that are provided on your CD-ROM include *DB2 Extended Enterprise Edition* and the *DB2 Client Application Enabler.* These are the features of each product:

DB2 Universal Database Extended Enterprise Edition
Provides the ability for a database to be partitioned across multiple independent machines of a common platform. To the end-user and application developer, it still appears as a single database on a single machine. This fully scalable database system enables an application to use a database that is simply too large for a single machine to handle efficiently. SQL operations and utilities can operate in parallel on the individual database partitions, thereby speeding up the execution time of a single query or command. This product also includes the DB2 Connect feature that allows access to databases on AS/400, MVS/ESA, OS/390, VM, or VSE systems. DB2 Extended Enterprise Edition was formerly known as DB2 Parallel Edition. The DB2 Client Application Enabler component is provided on the CD-ROM. It provides the ability for a client workstation to access a DB2 server.

Note: DB2 Universal Database Extended Enterprise Edition is commonly referred to as DB2 server or DB2 Universal Database throughout this book.

DB2 Client Application Enabler
The DB2 Client Application Enablers CD-ROMs are also included with DB2. They contain all the latest DB2 Client Application Enablers. With DB2 Client Application Enablers, clients from a variety of platforms can connect to DB2 databases.

There are several other DB2 products, which you can order separately:

DB2 Universal Database Workgroup Edition
A relational database management system that enables local and remote clients and applications to create, update, control, and manage relational databases using Structured Query Language (SQL), ODBC, or CLI. The DB2 Client Application Enabler component is provided on the CD-ROM. It provides the ability for a client workstation to access a DB2 server.

This product is available for OS/2 and Windows NT only.

DB2 Universal Database Enterprise Edition
Includes all the functions of the DB2 Universal Database Workgroup Edition and the DB2 Connect support that allows access to databases on host systems (for example: AS/400, MVS/ESA, OS/390, VM, or VSE). The DB2 Client Application Enabler component is provided on the CD-ROM. It provides the ability for a client workstation to access a DB2 server.

DB2 Connect Enterprise Edition
Provides access from clients on the network to DB2 databases residing on host systems such as AS/400, MVS/ESA, OS/390, VM, and VSE. DB2 Connect Enterprise Edition is available for the AIX, HP-UX, OS/2, Solaris, and Windows NT operating systems. This product was formerly known as DDCS Multi-User Gateway. The DB2 Client Application Enabler component is provided on the CD-ROM. It provides the ability for a client workstation to access a DB2 server.

DB2 Application Developer's Kit (AD Kit)
Contains a collection of DB2 Universal Database products, clients, DB2 Connect products, DB2 Software Developer's Kits, and application development tools for all supported operating systems.

The AD Kit gives you all the tools you need to create multimedia database applications that can run on a variety of platforms and can connect to any DB2 server, including DataJoiner.

Datajoiner provides access from DB2 clients on the network to databases residing on various heterogeneous data sources such as DB2, Informix, Microsoft SQL Server, Oracle, and Sybase.

DB2 Universal Database Personal Edition
Allows you to create and use local databases. DB2 Personal Edition can also act as client to access remote DB2 servers. This product is available for the OS/2, Windows 95, and Windows NT operating systems only.

DB2 Connect Personal Edition
Provides access from a single workstation to DB2 databases residing on host systems such as MVS/ESA, OS/390, OS/400, VM, and VSE, as well as access to DB2 databases. DB2 Connect Personal Edition is available for the OS/2, Windows 3.1x, Windows 95, and Windows NT operating systems. This product was formerly known as DDCS Single-User.

DB2 Client Application Enablers
Contains all the latest DB2 Client Application Enablers. This is not a separately orderable product, but it is available with DB2 products.

Working with Data Using DB2 Universal Database

As well as providing a relational database to store your data, DB2 lets you administer requests to query, update, insert, or delete data using local or remote client applications.

DB2 Universal Database includes tools that enable you to tune performance, access remote DB2 servers, manage all servers from a single site, develop powerful applications, and process SQL queries. For a description of these tools, see "Administering Databases with DB2 Tools" on page 8.

Accessing DB2 Data from Remote Clients

The DB2 Client Application Enabler provides a run-time environment that enables client applications to access one or more remote databases. Local applications and all Java applications (either local or remote) access a database through the DB2 Client Application Enabler component on the server. All remote applications that are not Java applets must have the DB2 Client Application Enabler installed on the client machine before they can access the remote database.

The *DB2 Client Application Enablers CD-ROMs* provide support for clients on the following operating systems:

- Macintosh
- OS/2
- UNIX (AIX, HP-UX, SCO OpenServer, SGI IRIX, SINIX, Solaris)
- Windows 3.x, Windows 95, Windows NT

The Client Application Enabler Version 1.2 for DOS is only available from the Web. See Chapter 8, "Installing DB2 Clients" on page 111 for information on how to download the DOS client.

Figure 1 on page 6 shows a DB2 server with local applications and applications that run on remote clients. This is the system that contains your databases. Any applications that run on this system are known as local applications. The client systems need to have the appropriate DB2 Client Application Enabler installed to enable applications to access the data on the remote server system.

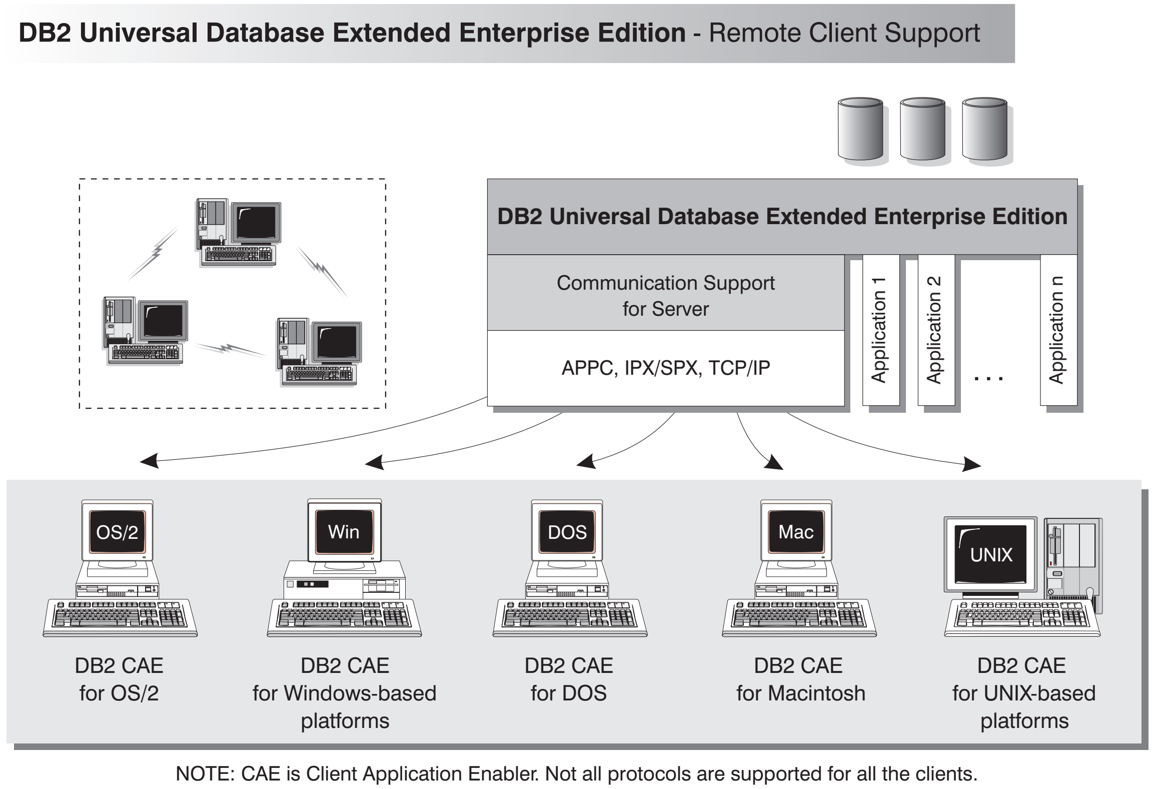

DB2 Universal Database Extended Enterprise Edition - Remote Client Support
DB2 Universal Database Extended Enterprise Edition
Communication Support for Server
APPC, IPX/SPX, TCP/IP
Application 1
Application 2
. . .
Application n
OS/2
Win
DOS
Mac
UNIX
DB2 CAE for OS/2
DB2 CAE for Windows-based platforms
DB2 CAE for DOS
DB2 CAE for Macintosh
DB2 CAE for UNIX-based platforms
NOTE: CAE is Client Application Enabler. Not all protocols are supported for all the clients.

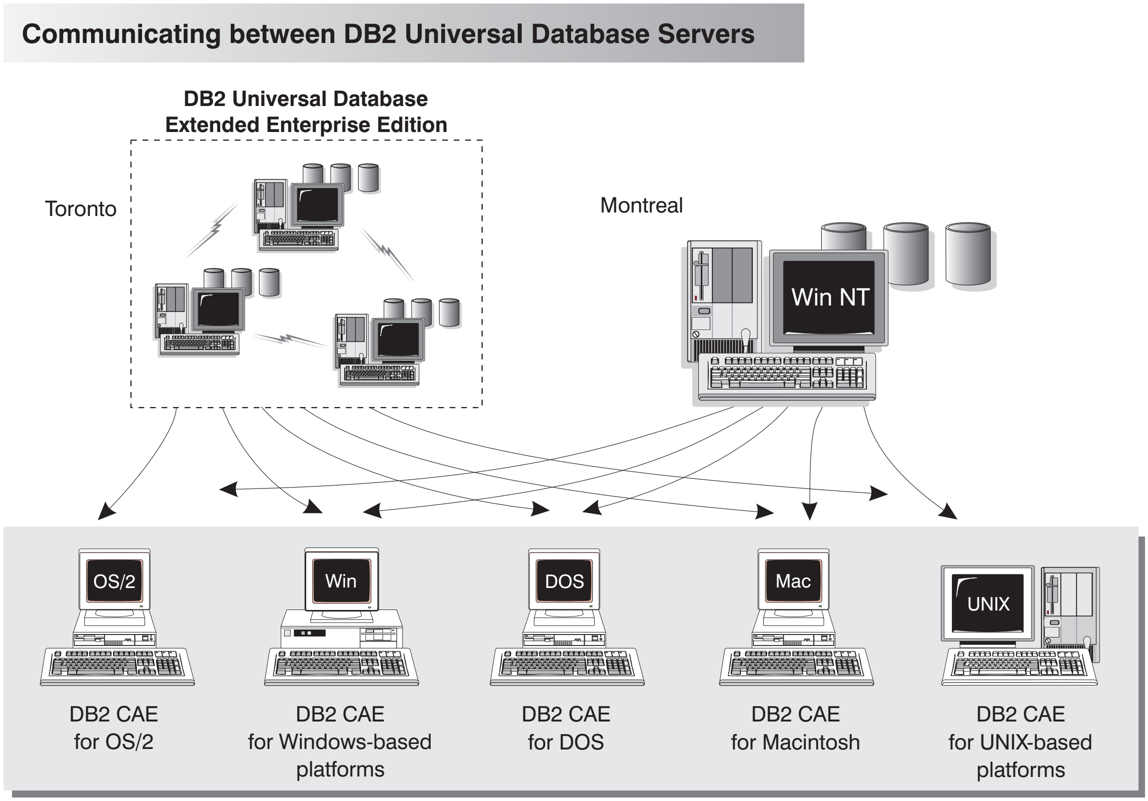
Figure 1. DB2 with Local Applications and Remote Clients

In addition to accepting requests from the clients that are listed above, DB2 Universal Database also has the DRDA Application Server function built in. It accepts requests from MVS, OS/400, VM, and other clients which are DRDA Application Requesters. See Chapter 25, “Using the DB2 Workstation as a Server for Host Applications” on page 255, or refer to the *Administration Guide* for more information.

Accessing Multiple DB2 Servers

Once a network is operational and protocols are functional on the workstations, LAN-to-LAN connections between DB2 servers and clients require no additional software.

For example, you can have a DB2 server that is connected to a LAN located in Montreal, and a DB2 Extended Enterprise Edition server connected to a LAN located in Toronto. As long as there is a connection between the two LANs, clients on either network can access either server. See Figure 2 on page 7.


Communicating between DB2 Universal Database Servers
DB2 Universal Database
Extended Enterprise Edition
Toronto
Montreal
Win NT
OS/2
Win
DOS
Mac
UNIX
DB2 CAE for OS/2
DB2 CAE for Windows-based platforms
DB2 CAE for DOS
DB2 CAE for Macintosh
DB2 CAE for UNIX-based platforms

NOTE: CAE is Client Application Enabler.

Figure 2. Accessing Data on Multiple Servers

Within a single transaction, databases on both servers are accessed and updated, and the integrity of the data on both servers is maintained. This is commonly known as two-phase commit, or distributed-unit-of-work access. Refer to the *Administration Guide* for more information.

Administering Databases with DB2 Tools

You can administer local databases on DB2 servers by using the command line processor. You can also perform database administration tasks remotely from an OS/2, Windows 95, or Windows NT client by using the DB2 administration tools. Use the *Control Center* to perform administration tasks such as configuring, backing up and recovering data, managing directories, scheduling jobs, and managing media, all from a graphical interface.

In a partitioned database system, you must also have a *listener* daemon that runs on each machine in the instance. This daemon is called `db2cclst`, and is used by all instances that are on the machine. The Control Center uses the listener daemon to retrieve status, connection, and snapshot information from each database partition server. The daemon is not associated with a specific instance; rather, it functions as a global server for the machine.

The listener daemon requires a predefined named port. The named port must be called `db2ccmsrv` and must be defined in the */etc/services* file on every machine. The named port can be assigned any unused port number, but the same number must be used for all machines.

You should set up the `db2cclst` daemon to start at machine reboot time. You can do this by changing the */etc/inittab* file as follows:

For AIX:
Run the following command:

```
mkitab "db2cclst:2:once:DASINSTHOME/sqllib/bin/db2cclst"
```

For Solaris:
Edit the */etc/inittab* file and add the following:

```
cl:234:once:DASINSTHOME/sqllib/bin/db2cclst
```

where *DASINSTHOME* is the home directory of the Administration Server.

Note: If you are using DB2 Extended Enterprise Edition for AIX, for serviceability reasons, you should not enable the *inittab* autorestart option for the `db2cclst` daemon.

To see if the daemon is running, issue the following command:

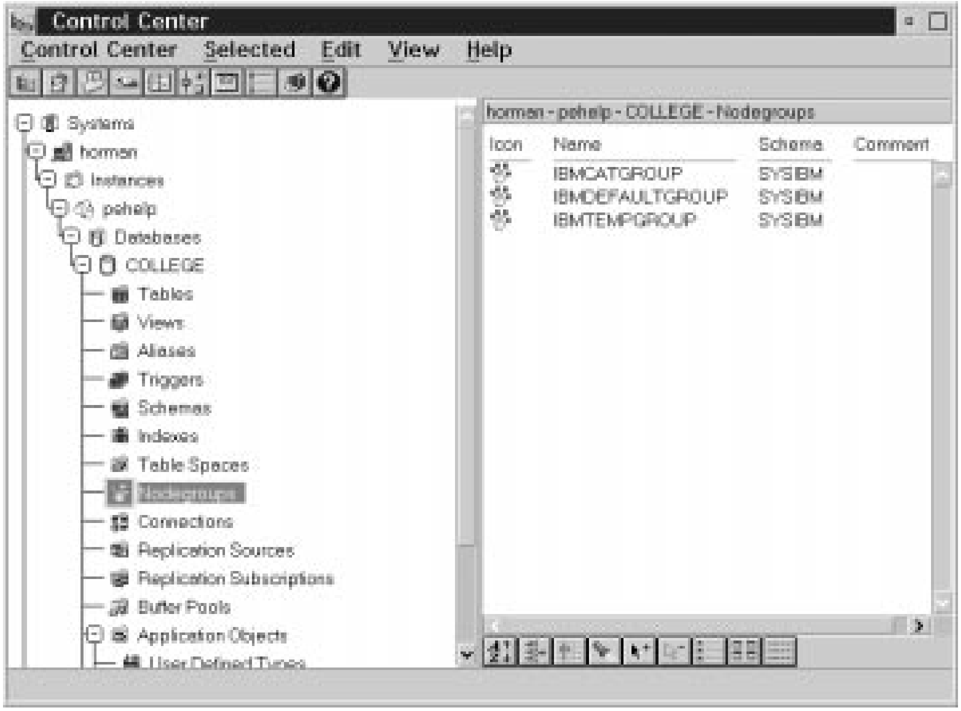
```
ps -aef | grep db2cclst
```

Error information for the `db2cclst` daemon is returned to the *syslog* file. Errors are written to this file, instead of the *db2diag.log* file, because the daemon runs independent from the instance.

Managing Databases Using the Control Center

The Control Center displays database objects (such as database partitions, table partitions, and packages) and their relationships to each other. Using the Control Center, you can manage a local database server or multiple remote DB2 servers and the database objects within them, all from a single point of control. See Figure 3 for an example of the main Control Center window. You can choose to install the Control Center on OS/2, Windows 95, or Windows NT workstations.

DB2 Universal Database Extended Enterprise Edition distinguishes between single-partition and multipartition database systems via *Discovery*. Discovery uses the *DB2SYSTEM*, *DB2ADMINSERVER*, and *DB2COMM* registry values. For more information on these registry values, see Chapter 21, "Controlling Your DB2 Environment" on page 213.



Icon	Name	Schema	Comment
	IBMCATGROUP	SYSIBM	
	IBMDEFAULTGROUP	SYSIBM	
	IBMTEMPGROUP	SYSIBM	

Figure 3. Control Center Main Window

From the Control Center, you can perform operations on database objects. These operations include:

- Create and drop a database
- Create, alter, and drop a table space or table
- Create, alter, and drop an index

- Create nodegroups
- Backup and recover a database partition or table space partition.
- Define the replication sources and subscriptions to replicate data between systems

For more information on objects in a partitioned database system, see Chapter 3, “Introduction to DB2 Extended Enterprise Edition” on page 21.

You can also control database manager environments (which are known as DB2 instances) by:

- Maintaining the communications protocols
- Setting database manager configuration values that affect performance

The Control Center provides additional features to assist you in managing your DB2 servers:

	Use the **Command Center** to enter DB2 commands and SQL statements in an interactive window and see the execution result in a result window. You can scroll through the results and save the output to a file.
	Use the **Script Center** to create mini applications called scripts, which you can store and invoke at a later time. These scripts can contain DB2 commands, SQL statements, as well as operating system commands. Scripts can be scheduled to run unattended. These jobs can be run once or set up to run on a repeating schedule; a repeating schedule is particularly useful for tasks like backups.
	Use the **Journal** to view all available information about jobs that are pending execution, executing, or that have completed execution. You can also view the recovery history log, the alerts log, and the messages log; and review the results of jobs that are run unattended.
	Use the **Alert Center** to monitor your system for early warnings of potential problems or to automate actions to correct problems discovered.
	Use the **Tools Setting** to change the settings for the Control Center, Alert Center, Replication, and Node Status.

You can run these functions from the Control Center toolbar or from icons in the Administration Tools folder.

You can install the following options for the Control Center:

DB2 Performance Monitor	Use the **DB2 Performance Monitor** to monitor the performance of your DB2 system. You can monitor activity using snapshots of data over a period of time or using data for a particular event. See “Monitoring Databases Using DB2 Performance Monitor” on page 11 for more information.
Visual Explain	Use **Visual Explain** to graphically analyze and tune SQL statements and to analyze query access plans. See “Viewing SQL Access Plans Using Visual Explain” on page 12 for more information.

You can find additional information in the *Administration Getting Started* guide or in the Control Center's online help.

Understanding the Administration Server

The Administration Server satisfies requests from the DB2 Administration Tools. It enables the DB2 Administration Tools to start, stop, catalog, and configure instances of the DB2 server. The Administration Server must reside on every DB2 server that you want to administer and detect. For more information, see Chapter 18, “Using the Administration Server” on page 185.

Monitoring Databases Using DB2 Performance Monitor

With the DB2 Performance Monitor, you can:

- Identify and analyze performance problems in the database applications or the database manager.
- Identify exception conditions based on thresholds you define.
- Use the early warning system to detect potential problems.
- Automate actions to correct problems that are discovered.
- Define your own statistics, in addition to the default set that is provided.

You can choose to monitor snapshots or events. The Snapshot Monitor allows you to capture point-in-time information at specified intervals. The Event Monitor allows you to record information over the duration of an event such as a connection.

Snapshot Monitoring Considerations

The snapshot monitoring function in the Control Center is split between the administrative client and the database server. A separate process is started on behalf of the user to gather snapshot information on the server and return the data to the client. This separate server process runs locally on the server and therefore it must be locally authenticated; because of this, the user must have a valid user ID and password on each server that is monitored. If the server is configured for SERVER authentication, this is transparent to the user. If the server is configured for CLIENT authentication, the user will be prompted for a user ID and password that will be authenticated locally on the server.

Note: DCE authentication for snapshot monitoring is not supported at this time.

You can find additional information in the *Administration Getting Started* guide or in the online help for DB2 Snapshot Monitor.

Viewing SQL Access Plans Using Visual Explain

Visual Explain helps database administrators and application developers to:

- View the access plan chosen by the database manager's optimizer for a given SQL statement.
- Tune SQL statements for better performance.
- Design application programs and databases.
- View all the details of the access plan, including the statistics in the system catalogs.
- Decide whether or not to add an index to a table.
- Identify the source of problems by analyzing the access plan or performance of SQL statements.
- Use the portable snapshot function to view snapshots from any remote DB2 server.
- Display access plans for partitioned and SMP systems.

You can find additional information in the *Administration Getting Started* guide or in the online help for Visual Explain.

Managing Communications on the Server

The Control Center allows you to view, update, and reset server protocol settings. These functions are accessed by clicking with the right mouse button on an instance and selecting the **Setup communications** option from the pop-up menu. This tool helps database administrators to:

- Configure communication parameters in the database manager.
- Generate database information in a profile that can be used to configure clients.

You can find additional information in the online help for the Control Center.

Managing Connections to Databases Using the Client Configuration Assistant

The Client Configuration Assistant helps you manage your database connections to remote DB2 servers. The Client Configuration Assistant is available on the OS/2, Windows 95, and Windows NT operating systems. Use the command line processor to set up DB2 clients on UNIX platforms.

With the Client Configuration Assistant, you can:

- Catalog the location and name of databases so that they can be used by applications. Three methods are available:

  - Search the network for available databases and select one. Client access is automatically set up for that database.
  - Use a database access profile provided by a database administrator to automatically define your connections. Client access is automatically set up for that database.
  - Manually configure a connection to a database by entering a few required connection parameters.
- Remove cataloged databases, or change the properties of a cataloged database.
- Test connections to local or remote databases identified on your system.
- Bind applications to a database by selecting utilities or bind files from a list.
- Tune the client configuration parameters on your system. Parameters are logically grouped and hints are provided on the interface as parameters are selected.
- If DB2 Connect is installed, set up a connection to a DRDA server.

Developing Applications Using the DB2 Software Developer's Kit

The DB2 Software Developer's Kit is a collection of tools that are designed to meet the needs of database application developers. It includes libraries, header files, documented APIs, and sample programs to build character-based, multimedia, or object-oriented applications.

You can install the DB2 Software Developer's Kit on a DB2 server or on a remote workstation. If you install it on the server, it can access both remote and local databases.

There is a platform-specific version of the DB2 Software Developer's Kit available for each of the supported operating systems. The *DB2 Application Developer's Kit (DB2 AD Kit)* provides the DB2 Software Developer's Kits for all supported platforms, the DB2 Extender development toolset, and VisualAge for Basic. Applications that are developed with the DB2 Software Developer's Kit will run on any platform where the equivalent DB2 Client Application Enabler component is installed. Through the DB2 Client Application Enabler, these applications can access all DB2 servers and, by using DB2 Connect, they can also access DRDA Application Servers.

The DB2 Software Developer's Kit allows you to develop applications that use the following interfaces:

- Embedded SQL
- Call Level Interface (CLI) development environment (which is compatible with ODBC from Microsoft)
- Java Database Connectivity (JDBC)
- Application programming interfaces to access database utilities.

The DB2 Software Developer's Kit supports several programming languages (which include COBOL, C, and C++) for application development, and provides precompilers for the supported languages. Refer to the *Building Applications for UNIX Environments* or the *Building Applications for Windows and OS/2 Environments* manuals for complete information on using the DB2 Software Developer's Kit, and for a list of the supported compilers.

Accessing Host Data from the Desktop Using DB2 Connect Enterprise Edition

DB2 Connect Enterprise Edition gives clients on your LAN access to data that is stored on host systems. It provides applications with transparent access to host data through a standard architecture for managing distributed data. This standard is known as Distributed Relational Database Architecture (DRDA). DRDA allows your applications to establish a fast connection to databases on MVS, OS/390, OS/400, VM, and VSE host systems.

A great deal of the data in many large organizations is managed by DB2 for AS/400, DB2 for MVS/ESA, DB2 for OS/390, or DB2 for VSE & VM. Applications that run on any of the supported platforms can work with this data transparently, as if a local database server managed it.

The following tools and products can access host data easily using DB2 Connect:

- *Spreadsheets*, such as Lotus 1-2-3 and Microsoft Excel, to analyze real-time data without the cost and complexity of data extraction procedures and import procedures.
- *Decision support tools*, such as Business Objects, Intersolv Q+E Database Editor, and Crystal Reports, to provide real-time information.
- *Personal database products*, such as Lotus Approach and Microsoft Access.
- *Development tools*, such as IBM VisualAge, PowerBuilder, Microsoft VisualBasic, and Borland Delphi to create client/server solutions.

DB2 Connect Enterprise Edition enables multiple clients to connect to host data and can significantly reduce the effort that is required to establish and maintain access to enterprise data. See Figure 4 on page 15 for an example of clients connecting to host databases through DB2 Connect Enterprise Edition.

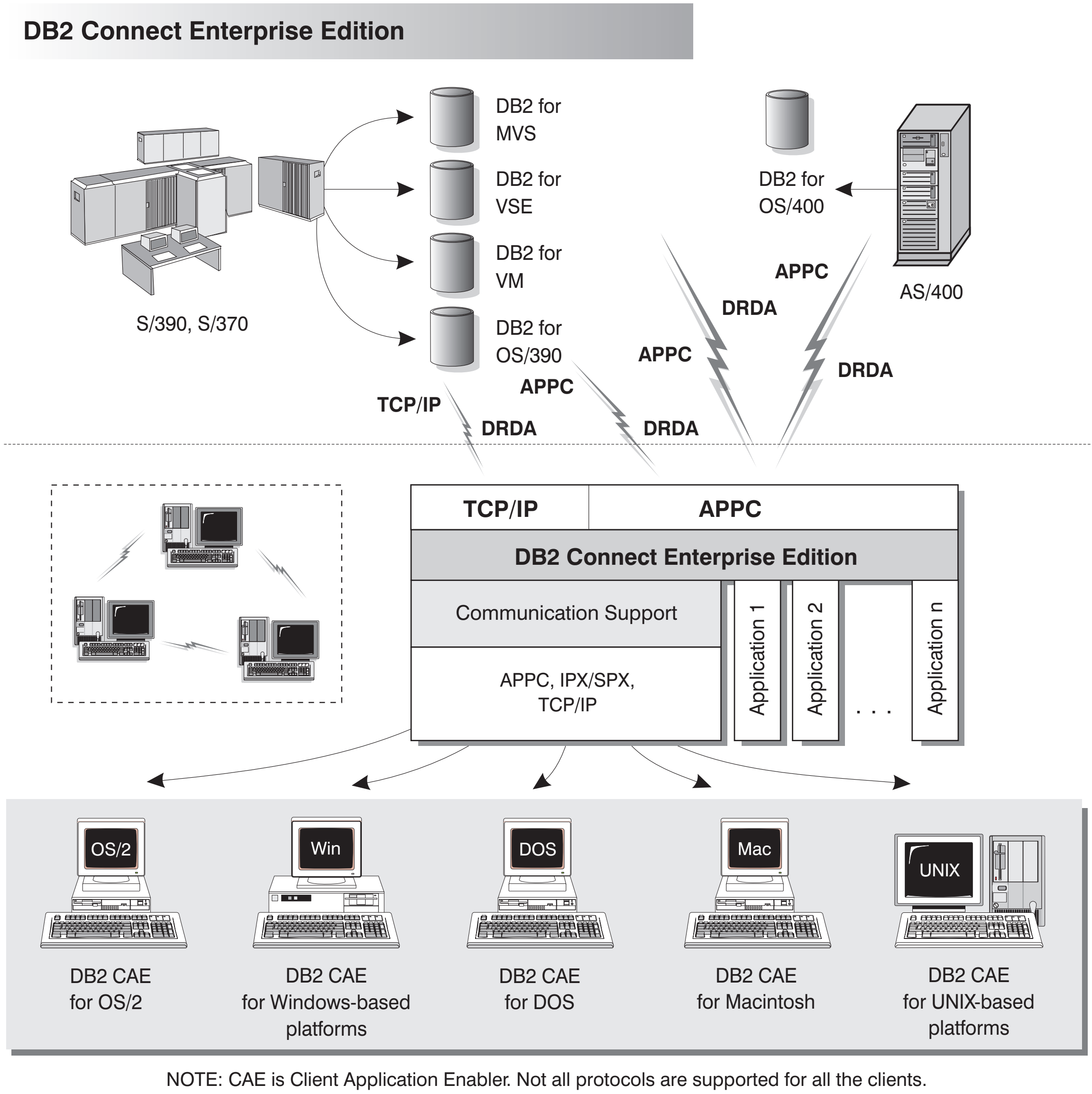


Figure 4. DB2 Connect Enterprise Edition

Accessing DB2 Data from the Web

The popularity of the Internet and the World Wide Web has created a demand for Web access to enterprise data. Java Database Connectivity (JDBC) and Net.Data are provided with DB2 to allow you to create applications that access data in DB2 databases.

Use *Java Database Connectivity* to create applications or applets that access data in DB2 databases. You can run JDBC applets inside HTML (Hypertext Markup Language) web pages on any system with a Java-enabled browser, regardless of the platform of your client. Your client system requires no additional software beyond this browser. The client and the server share the processing of JDBC applets.

The JDBC server and the DB2 Client Application Enabler must reside on the same machine as the Web server. The JDBC server calls the DB2 Client Application Enabler to connect to local, remote, or host databases. When the applet requests a connection to a DB2 database, the JDBC client opens a TCP/IP socket to the JDBC server on the machine where the Web server is running. See Figure 5 for an example of Java-enabled browsers accessing data from remote DB2 databases.

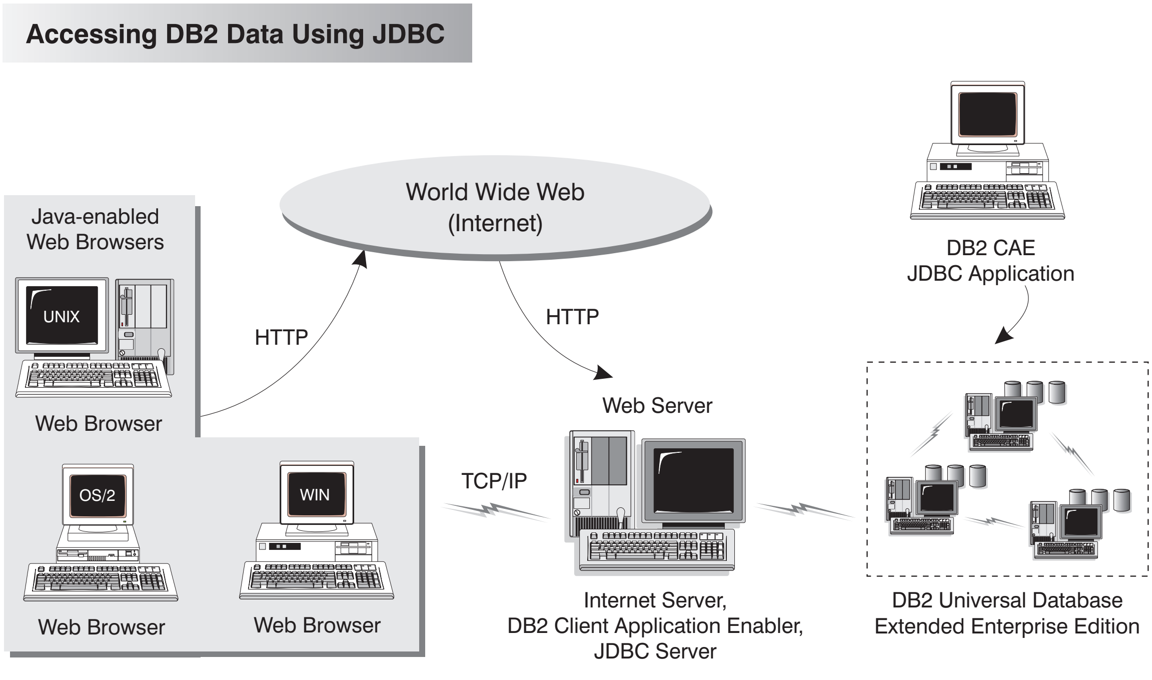

Accessing DB2 Data Using JDBC
Java-enabled Web Browsers
UNIX
Web Browser
OS/2
Web Browser
WIN
Web Browser
World Wide Web (Internet)
HTTP
HTTP
Web Server
TCP/IP
Internet Server, DB2 Client Application Enabler, JDBC Server
DB2 CAE JDBC Application
DB2 Universal Database Extended Enterprise Edition

Figure 5. Accessing Internet Data Stored on DB2 Using JDBC

JDBC applications can be run from any system that has a DB2 Client Application Enabler installed; a Web browser and a Web server are not required.

For more information on Java enablement, refer to the IBM Software Page under the Data Management link (http://www.software.ibm.com/data/db2/java/). For more information on the JDBC API, refer to the URL http://splash.javasoft.com/.

Use *Net.Data* to create applications that are stored on a Web server and viewable from any Web browser. While viewing these documents, users can either select automated queries or define new ones that retrieve the specified information directly from a DB2 database.

Automated queries do not require user input; they are links in the HTML document and, when selected, they trigger existing SQL queries and return the results from a DB2 database. These links can be triggered repeatedly to access current DB2 data. Customized queries require user input. Users define the search characteristics on the Web page by selecting options from a list or by typing values in the entry fields. They submit the search by clicking on a push button. Net.Data uses the information that is supplied by the user to dynamically build a complete SQL statement, and it sends the query to the database.

A demonstration of Net.Data applications is available from the IBM Software Page (http://www.software.ibm.com/) under the Data Management link.

Net.Data can be installed with a DB2 server to allow local access to databases. Net.Data can be installed with a DB2 Client Application Enabler to allow remote access to databases. In both cases, Net.Data and a web server must be installed on the same system. See Figure 6 on page 18 for an example of Net.Data that is being used to access data from remote DB2 databases.

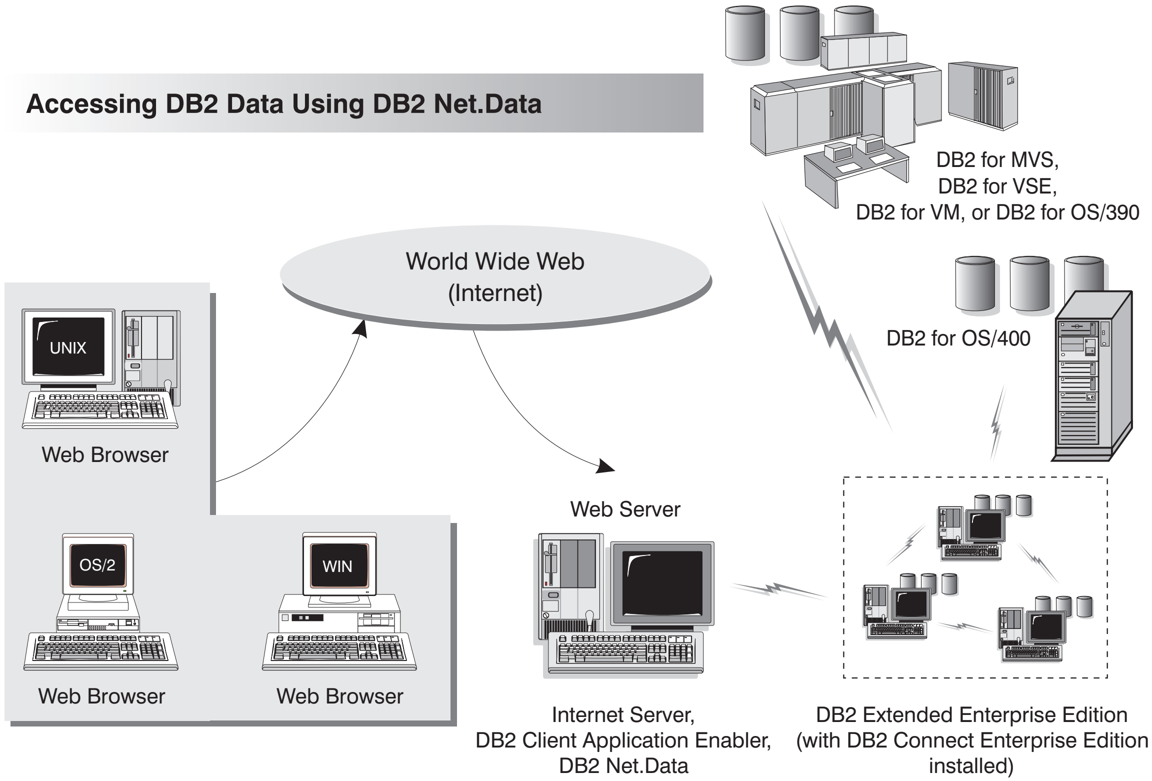

Accessing DB2 Data Using DB2 Net.Data
DB2 for MVS,
DB2 for VSE,
DB2 for VM, or DB2 for OS/390
World Wide Web
(Internet)
DB2 for OS/400
UNIX
Web Browser
Web Server
OS/2
WIN
Web Browser
Web Browser
Internet Server,
DB2 Client Application Enabler,
DB2 Net.Data
DB2 Extended Enterprise Edition
(with DB2 Connect Enterprise Edition
installed)

Figure 6. Accessing Internet Data Stored on DB2 Using Net.Data

Chapter 2. What's New in DB2 Version 5?

Version 5 of DB2 Universal Database Extended Enterprise Edition is the follow-on product to DB2 Parallel Edition Version 1.2 (which was available on AIX only); it includes all the features and enhancements of DB2 Version 2 and Database Server Version 4, and offers many new features. This section describes some of the major changes for this version, and points you to sources of more information in the DB2 library. The numerous enhancements include:

Exploitation of symmetric multiprocessors (SMP) for improved scalability and performance
DB2 now exploits the ability of an SMP system to share resources across multiple processors, performing your SQL queries more quickly. Commands like **load**, **backup**, and **restore** also take advantage of the multiprocessor environments.

For more information on SMP parallelism, refer to the *Administration Guide*. Commands are described in more detail in the *Command Reference*.

New graphical tools on OS/2, Windows 95, and Windows NT that make it easy to install, configure, and administer DB2 databases
From the Control Center, you can accomplish just about any administrative task. You can also use the graphical tools to administer any DB2 server on any of the available platforms. For more details, refer to the online help available with the Control Center.

From the Client Configuration Assistant, you can configure communications for clients to access remote or local DB2 servers. For more details, refer to the online help available with the Client Configuration Assistant.

Comprehensive online help and the Information Center help you easily find the information you need to use DB2.

Extensions to SQL, including support for Online Analytical Processing (OLAP)
With the new CUBE and ROLLUP aggregations, you can now create *super groups*, like *overall total* and *cross tabulation*, for improved online analytical processing. You can also perform outer joins, rename tables, define unique constraints on tables, and create your own table functions. DB2 has refined its authorization support, so you can grant authority at a more granular level; you can also create schemas to grant various attributes and privileges. For more details on these and other new SQL features, refer to the *SQL Reference*.

Performance, capacity, and memory improvements
DB2 now provides global caching for SQL statements, creating a public repository that improves performance. You can create multiple buffer pools of various sizes to better control the data in memory. Client/server communications are more efficient, crash and roll-forward recovery is faster, and **load** features numerous performance enhancements. For information on how to exploit these enhancements, refer to the *Administration Guide*.

Security enhancements, including support for Open Software Foundation's Distributed Computing Environment (DCE)
You can now use the DCE architecture to manage users, passwords, and groups more easily, and authenticate users more securely. DB2 also provides a "Trusted Clients" option so you can choose whether to trust all clients or only those that come from an operating system with inherent security. Security considerations are described more fully in the *Administration Guide*.

Additional support for communicating with host databases using DB2 Connect and the Distributed Relational Database Architecture (DRDA)
In addition to communicating with host systems using SNA, you can now use TCP/IP to communicate with host systems that support this protocol. (Currently, only DB2 for OS/390 Version 5.1 supports TCP/IP.) In addition, DB2 servers can accept requests from host systems using TCP/IP, allowing you to use your DB2 workstation server as an application server to a host application. On OS/2 and UNIX platforms, you can perform two-phase commits using DB2 Connect and TCP/IP; if you're using SNA, you can invoke a two-phase commit involving both the host and the DB2 server. For more details about connecting to host databases, refer to the *DB2 Connect Enterprise Edition Quick Beginnings* and the *DB2 Connect User's Guide*.

Enhancements to application programming for DB2
The DB2 Call Level Interface (DB2 CLI) now reflects the Microsoft Open Database Connectivity 3.0 specifications, allowing you to connect to DB2 databases from ODBC 3.0 applications. Additions to user defined functions (UDFs) include scrollable cursors and the UCT_UNIQUE function to return a unique value to use in a table column. You can also invoke external UDFs that are methods of object linking and embedding (OLE) automation servers. For more details on these and other improvements, refer to the *CLI Guide and Reference*, *Embedded SQL Programming Guide*, and the *API Reference*.

There are many other new features, for example:

- The DB2 Governor which you can use to control application behavior.
- New table space features, such as point-in-time roll-forward recovery and selective restore.
- Improved error messages.

For a comprehensive list of what's new, refer to the Information Center.

Chapter 3. Introduction to DB2 Extended Enterprise Edition

A *database* is simply a collection of data. A *database manager* is the software that allows users to store and access data in a database. It achieves this function by using system resources, including CPU, memory, disk, and communications. In a partitioned database system, a single database manager and the collection of data and system resources that it manages are referred to collectively as a *database partition server*. A *partitioned database system* is the collection of all the database partition servers that you create to handle data requests.

In DB2 Extended Enterprise Edition, the quality, functionality, reliability, and robustness of the database manager is extended to the entire partitioned database system.

In a partitioned database system, multiple database partition servers can be assigned to a machine (or to multiple machines), and the database manager at each machine is responsible for a portion of the database's total data (each database partition server houses a portion of the entire database). This portion of the database is known as a *database partition*. The fact that databases are partitioned across database partition servers (or even machines) is transparent to users of SQL statements.

The system can maintain very large databases, and open opportunities for new applications. DB2 Extended Enterprise Edition provides a fast response for both decision-support (DSS) and online transaction processing (OLTP) applications.

DB2 Extended Enterprise Edition can be configured to execute on a *shared-nothing* hardware architecture, in which machines do not have to compete for resources. Each machine has exclusive access to its own disks and memory, and the machines communicate with each other through the use of messages. For a database system to exploit shared-nothing architecture, typically one database partition server is assigned to each machine. Another possible configuration is running multiple logical nodes, in which more than one database partition server runs on a machine. For more information, see “Multiple Logical Nodes” on page 28.

Processing in the Partitioned Database Environment

In DB2 Extended Enterprise Edition, the database is distributed across multiple machines, and database partition servers are installed on a set of machines. Because the database is partitioned across multiple machines, you can use multiple CPUs to satisfy requests for information. The retrieval and update requests are decomposed automatically into sub-requests and executed in parallel on the database partition servers on each machine.

As an illustration of the power of processing in a partitioned database system, assume that you have 100 000 000 records that you want to scan in a single-partition database. This scan would require that a single database manager search 100 000 000 records. Now suppose that these records are spread evenly over 20 database partition servers; each database manager only has to scan 5 000 000 records. If each database manager scans at the same time and with the same speed, the time required to do the scan should be approximately 5% of that of a single-partition system handling this task.

User interaction with DB2 Extended Enterprise Edition is handled through one database partition server (node). This database partition server is known as the *coordinator node* for that user. The coordinator node runs on the same machine as the application, regardless of whether the user or application is remote or local. Any database partition server can be used as a coordinator node. You should consider spreading out users across database partitions servers to distribute the coordinator function.

Cost-Based Query Optimization

DB2 Extended Enterprise Edition uses a *cost-based query optimizer*, which compares different methods for doing a unit of work, and selects the most efficient one. The optimizer provides the following features:

Transparent parallelism
Both new and existing applications that use data-manipulating SQL statements do not have to be changed when they are migrated to DB2 Extended Enterprise Edition. You only have to rebind them so the optimizer can generate the best plans for existing SQL queries.

Comprehensive use of data partitioning information
The optimizer uses information about how base tables, and the intermediate tables that result from queries, are partitioned across database partitions. This information is used to determine the best execution strategy.

Full-fledged, cost-based optimization
The optimizer considers different execution plans and chooses the one with the lowest cost. While comparing different strategies, it accounts for the inherent parallelism of different operations, and the costs introduced by messages.

Inter-partition parallelism of all relational operations.
All operations, including index and table scans, aggregation, set operations, joins, inserts, deletes, and updates can employ *inter-partition parallelism*. (Inter-partition parallelism means that the operator is executed in parallel by each database partition server; *intra-partition parallelism* means that different operators in the same query can be executed in parallel by the same database partition server.) When generating plans, the optimizer considers different parallel joining methods, including *collocated*, *directed*, and *broadcast* joins.

As a simple example, assume that you want to fetch specific columns from rows that meet some condition. The coordinator node sends a request to the other database partition servers to select the columns from rows that meet that condition. The other database partition servers then send this data to the coordinator node, which does the final processing.

If a row is being added to a table, the database partition server checks the *partitioning map*, which specifies the database partition server where the row is stored. The row is only sent to that database partition server, with the result that only the interested database partition servers take part in the insert. This keeps communications overhead as low as possible.

Configuration

Figure 7 shows an example of a DB2 Extended Enterprise Edition hardware configuration.

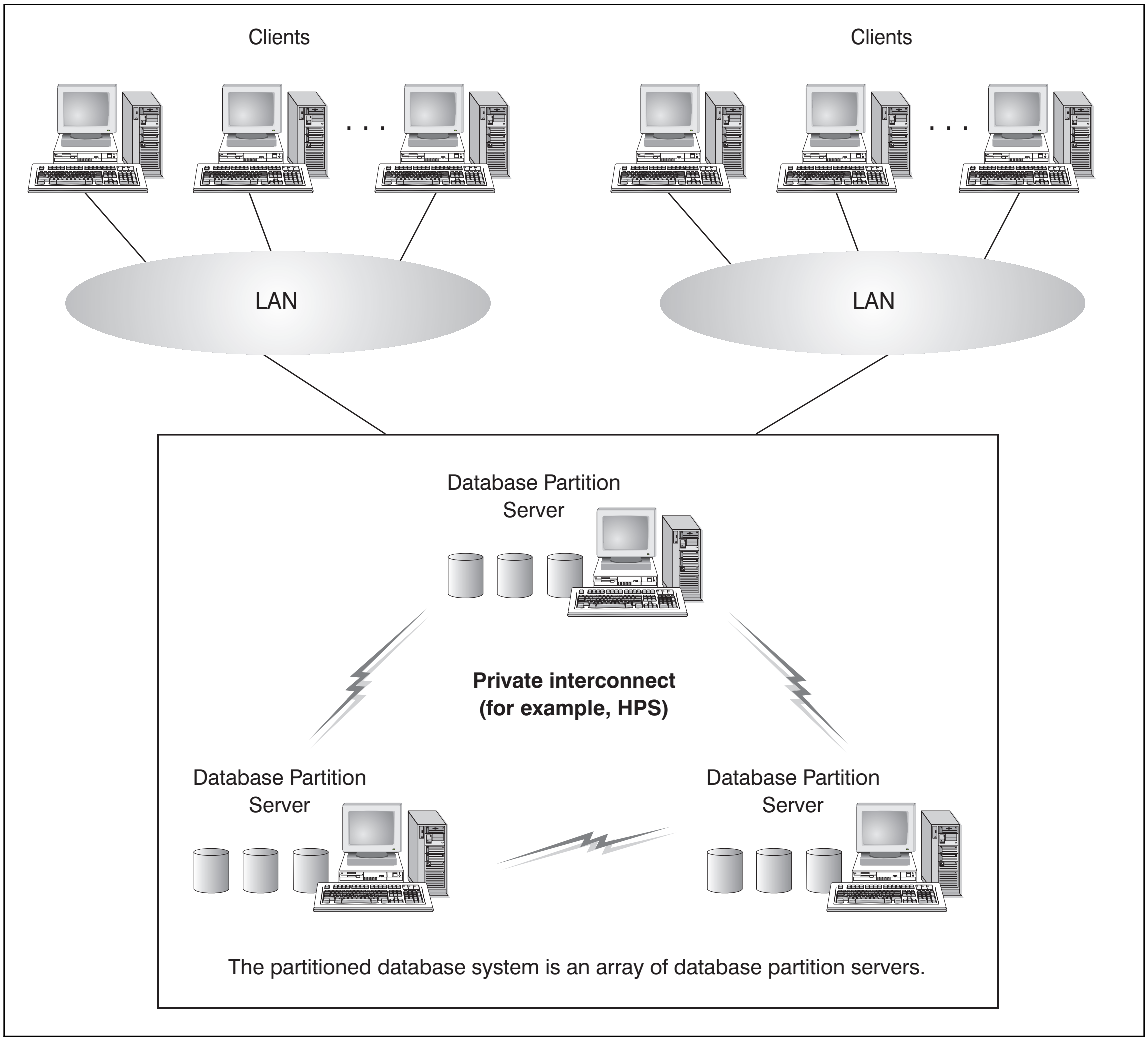

Clients
Clients
. . .
. . .
LAN
LAN
Database Partition Server
Private interconnect (for example, HPS)
Database Partition Server
Database Partition Server
The partitioned database system is an array of database partition servers.

Figure 7. DB2 Extended Enterprise Edition Hardware Configuration

DB2 Extended Enterprise Edition can run on a cluster of individual CPUs interconnected by shared memory (symmetric multiprocessors (SMP)), a dedicated high-speed communications switch (for example, HPS), or a LAN. The number of database partition servers in a configuration varies by platform. You should limit the number of database partition servers that communicate over a LAN to 16.

In practice, the number of database partition servers in a configuration is limited by the platform and the management tools available on each platform. For more information about configuration, refer to the *Administration Guide*.

For example, in an IBM RISC/6000 Scalable POWERparallel Systems (RISC/6000 SP) environment that is running AIX, the number of database partition servers is only limited by the possible size of an AIX RISC/6000 SP system.

In a Solaris environment, the number of database partition servers is limited by the size of the machines and the number of those machines that are clustered together. Forty database partition servers could be run on a clustered system of four Ultra Enterprise 6000s with ten CPUs each.

The following sections provide information that you should be familiar with before you configure your partitioned database system. Specifically, they describe:

- Physical storage
- How to combine database partitions into nodegroups, and how data is stored in them
- Logical nodes and their uses
- Instances
- Communications between database partition servers
- Agents

Machines and Storage

DB2 Extended Enterprise Edition implements a shared-nothing architecture, therefore each database partition server is the equivalent of a single-partition database system. Thus, the database storage capacity for the partitioned database system is equal to that provided by the single-partition database system multiplied by the number of nodes. You can store tables of up to 64 GB (gigabytes) per database partition, where giga is 10^9. For example, in a 128-database partition (node) system, the maximum size of one table is 8 TB (terabytes, where tera is 10^{12}).

Nodegroups and Data Partitioning

You can define named subsets of one or more database partitions in a database. Each subset you define is known as a *nodegroup*. Each subset that contains more than one database partition is known as a *multipartition nodegroup*. Multipartition nodegroups can only be defined with database partitions that belong to the same database.

Three default nodegroups are created when you create a database: IBMDEFAULTGROUP, IBMCATGROUP and IBMTEMPGROUP.

If you want, you can create tables in the default nodegroups, IBMDEFAULTGROUP and IBMCATGROUP.

The IBMDEFAULTGROUP nodegroup contains all the database partitions for the database. When you create a database, a database partition is created at each database partition server (node) that is defined in the node configuration file (*db2nodes.cfg*).

The IBMCATGROUP nodegroup for the database is created at the database partition server where you issued the **create database** command. This nodegroup only contains the database partition that is local to the database partition server where the command was issued. This database partition server is referred to as the *catalog node* of the database because the IBMCATGROUP nodegroup contains the catalog tables for the database.

You cannot directly work with the third default nodegroup, IBMTEMPGROUP. Like the IBMDEFAULTGROUP nodegroup, it also contains all the database partitions of the database. This nodegroup is used to contain all temporary table spaces.

Figure 8 on page 27 shows an example of a database in which there are three nodegroups. Nodegroup 1 is a multipartition nodegroup made of four database partitions, and nodegroups 2 and 3 are both single-partition nodegroups.

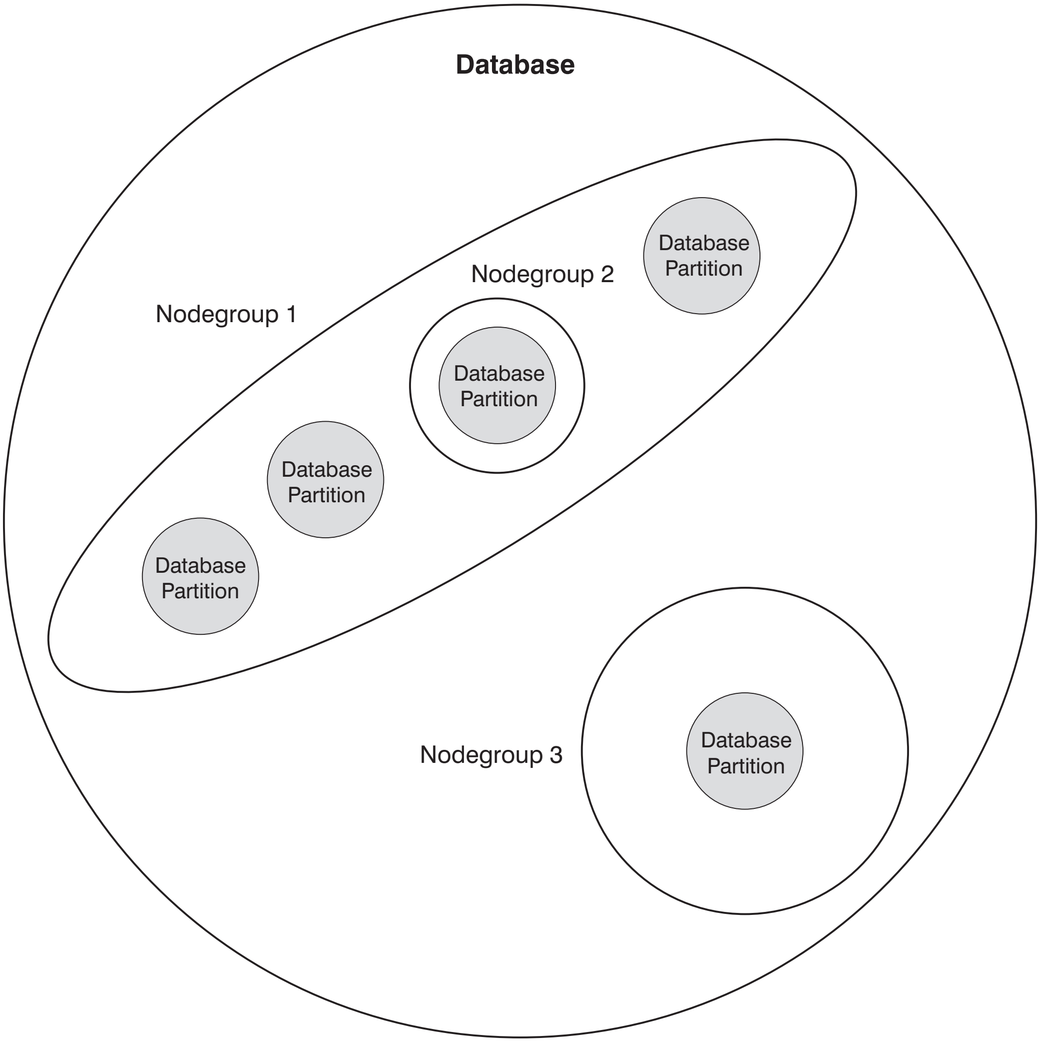


Figure 8. Nodegroups in a Database

When you want to create table spaces for a database, you first create the nodegroup where the table spaces will be stored, then create a table space in the nodegroup. After this, you create the tables in the table space.

You can drop database partitions from a nodegroup, or if new nodes have been defined in the *db2nodes.cfg* file, you can add them to a nodegroup in a database. For information about adding and dropping nodes in nodegroups, refer to the *Administration Guide*.

As your database increases in size, you can add database partition servers to the database system for improved performance. This is known as scaling the database system. When you add a database partition server, a database partition is created for each database that already exists in the database system. You then add the new database partition to an existing nodegroup that belongs to that database. Then you redistribute data in that nodegroup to utilize the new database partition. For information about scaling a database system, refer to the *Administration Guide*.

Each table defined in a multipartition nodegroup has a *partitioning key* associated with it. The partitioning key is an ordered set of columns whose values are used in conjunction with a *partitioning map* to determine the database partition on which a row of a given table resides. The partitioning map is an array of 4 096 database partition numbers.

Columns of any data type (except LONG VARCHAR, LONG VARGRAPHIC, or LOB) can be used as the partitioning key. A table defined in a single-partition nodegroup may or may not have a partitioning key. Tables with only long-field columns can be defined only in single-partition nodegroups, and they cannot have a partitioning key. For more information about creating tables, refer to the *SQL Reference*.

The use of nodegroups and partitioning keys means that:

- Data can be distributed across multiple database partitions to reduce I/O and processing bottlenecks.
- Data can be redistributed when large volumes of system activity or increases in table size require the addition of more machines.

For more information about creating nodegroups, refer to the *SQL Reference*. For more information about using nodegroups, refer to the *Administration Guide*.

Multiple Logical Nodes

Typically, you can configure DB2 Extended Enterprise Edition to have one database partition server assigned to each machine. There are situations, however, in which it would be advantageous to have several database partition servers running on the same machine. This means that the configuration can contain more logical nodes than machines. In these cases, the machine is said to be running *multiple logical nodes* if they participate in the *same* instance. If they participate in different instances, this machine is *not* running multiple logical nodes.

With multiple logical node support, you can choose from three types of configurations:

- A standard configuration, where each machine has only one database partition server.
- A multiple logical node configuration where a machine has more than one database partition server.

- A configuration where several logical nodes run on each of several machines.

Configurations that use multiple logical nodes are useful when the system runs queries on a machine that has symmetric multiprocessor (SMP) architecture. The ability to configure multiple logical nodes on a machine is also useful if a machine fails. If a machine fails (causing the database partition server or servers on it to fail), you can restart the database partition server (or servers) on another machine, using the **db2start** command with the *nodenum* parameter. This ensures that user data remains available. Another benefit is that multiple logical nodes can exploit SMP hardware configurations. In addition, because database partitions are smaller, you can obtain better performance when performing such tasks as backing up and restoring database partitions and table spaces, and creating indexes.

The *DB2NODE* environment variable is used to route requests to a target logical node within a machine. This environment variable must be set in the session in which the application or command is issued and *not* in the DB2 profile registry. If this variable is not set, the target logical node defaults to the logical node which is defined with port *0* on the machine. For more information, see “Connecting to a Logical Node” on page 211.

For more information about setting up logical nodes, see “Step 5. Create the Node Configuration File” on page 64 (for AIX) or “Step 8. Create the Node Configuration File” on page 84 (for Solaris), or refer to the *Administration Guide*.

Instances

In DB2 Extended Enterprise Edition, an instance contains all the database partition servers (nodes) that were defined to take part in a given partitioned database system. The database partition servers are defined in the *db2nodes.cfg* file as *nodes*. An instance has its own databases and instance directory. The instance directory contains the database manager configuration file, system database directories, node directories, and node configuration file. See “Instance Directory” on page 173 for more information.

Each instance has different security from other instances on the same machine. This is shown in Figure 9 on page 30, which shows two separate instances. Instance 1 contains six database partition servers and Instance 2 contains eight database partition servers. The two instances appear to overlap, but this is due to the assignment of two database partition servers to each of the three machines in the middle of the figure. (Multiple database partition servers are indicated when more than one line is shown between a database partition server and the instance directory.) The *db2nodes.cfg* file of Instance 1 will not list the database partition servers that belong to Instance 2, and the converse.

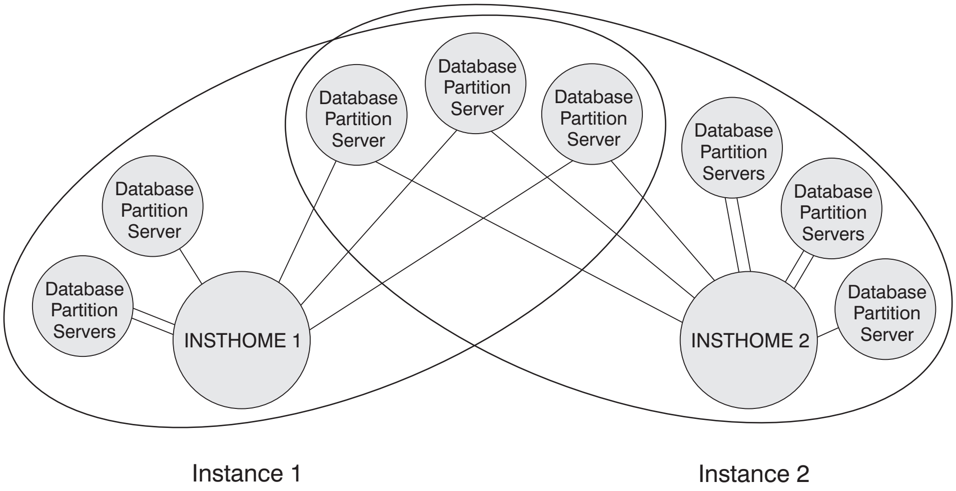

Database Partition Server
Database Partition Server
Database Partition Server
Database Partition Servers
Database Partition Server
Database Partition Servers
Database Partition Servers
Database Partition Server
INSTHOME 1
INSTHOME 2
Instance 1
Instance 2

Figure 9. Two Instances

You can have multiple instances on the same machine, with each configured differently:

- To have distinct test and production environments
- To use different versions of DB2; for example, one instance can use DB2 Parallel Edition Version 1.x and another can use DB2 Extended Enterprise Edition Version 5.x
- To restrict access to specific databases
- To exploit different database configurations

Each instance is owned by a user known as the *instance owner*. For information about generating instances, see Chapter 17, “Working with Instances” on page 173.

The instance owner has *SYSADM* authority for all databases that belong to the instance. Because the instance owner has almost complete control over the instance, this user ID can:

- Issue the **db2start** and **db2stop** commands on the instance
- Modify database configuration parameters
- Modify database manager configuration parameters
- Grant privileges to and revoke them from other users

The instance owner cannot remove an instance. This requires *root* authority.

A one-to-one correspondence exists between an instance and an instance owner; that is, a user cannot own more than one instance. (However, an instance owner may possess authorizations for other instances, up to, and including, *SYSADM*). In addition to this, each instance must have a separate home directory.

Fast Communication Manager

The fast communication manager (FCM) provides communication support for DB2 Extended Enterprise Edition. It consists of:

- A communications process, known as the FCM daemon (`db2fcmdm`)
- Requester functions that run within database manager processes
- Initialization and termination functions

Each database partition server has one FCM daemon to provide communications between database partition servers to handle agent requests, and to deliver message buffers.

You can specify the number of FCM message buffers with the *fcm_num_buffers* database manager configuration parameter. For a description of this and other FCM parameters, refer to the *Administration Guide*. The FCM daemon is started when you issue the **db2start** command. When the daemon starts, it reads the node configuration file (*INSTHOME/sqllib/db2nodes.cfg*) and defines a well-known address to use in communications.

If communications fail between database partition servers or if they re-establish communications, the FCM daemon updates information (that you can query with the database system monitor) and causes the appropriate action (such as the rollback of an affected transaction) to be performed. You can use the database system monitor to help you set the FCM configuration parameters. For information about FCM-related database system monitor output, refer to the *System Monitor Guide and Reference*.

Agents

All connection requests from client applications, whether they are local or remote, are allocated a corresponding coordinator agent (`db2agent`). In symmetric multiprocessor (SMP) systems, multipartition database systems, and single-partition database systems in which the *intra_parallel* database manager configuration parameter is enabled, the coordinator agent distributes the database requests to subagents (`db2agntp`), and these subagents perform the requests for the application. Once the coordinator agent is created, it handles all database requests on behalf of its application, coordinating the subagents that perform requests on the database.

A coordinator agent may be:

- Connected to the database with an alias. For example, `db2agent (DATA1)` is connected to the database alias `DATA1`.
- Attached to an instance. For example, `db2agent (instance)` is attached to an instance.

Idle and associated agents reside in an agent pool. These agents are available for requests from coordinator agents operating on behalf of client programs, or from subagents operating on behalf of existing coordinator agents. The number of available agents is dependent on the *maxagents* and *num_poolagents* database manager configuration parameters.

For additional information about agents and the configuration parameters used to control them, refer to the *Administration Guide*. For information about the DB2 process model, refer to the *Troubleshooting Guide*.

Part 2. DB2 Server Installation and Initial Configuration

Chapter 4. Planning for Installation

If you know that your system meets DB2's hardware and software requirements, and you want to start installing DB2 right away, go directly to Chapter 5, “Installing and Configuring DB2 Universal Database Extended Enterprise Edition for AIX” on page 51 or Chapter 6, “Installing and Configuring DB2 Universal Database Extended Enterprise Edition for Solaris” on page 73.

DB2 provides many components that you might want to use in your environment. Use the product and planning information in this section to confirm that your system fulfills the system requirements, and to decide which components you want to install.

Memory Requirements

Before you install DB2 Extended Enterprise Edition, decide what system configuration you need.

This section shows diagrams of the most common configurations that use DB2. Each diagram shows the products or components that are required for each system in the network (excluding operating system and communication requirements), and the approximate amount of memory required for such a configuration.

Configuration for Accessing DB2 Servers on the LAN

If you want to access data across a LAN from a remote database server, use the configuration shown in Figure 10 on page 36. This diagram shows DB2 Universal Database Extended Enterprise Edition installed on the server machine and various clients connected to this server to access data. You must install the appropriate DB2 Client Application Enabler on each client.

Clients connect to a DB2 Universal Database Extended Enterprise Edition server using one of the supported protocols: APPC, IPX/SPX, or TCP/IP.

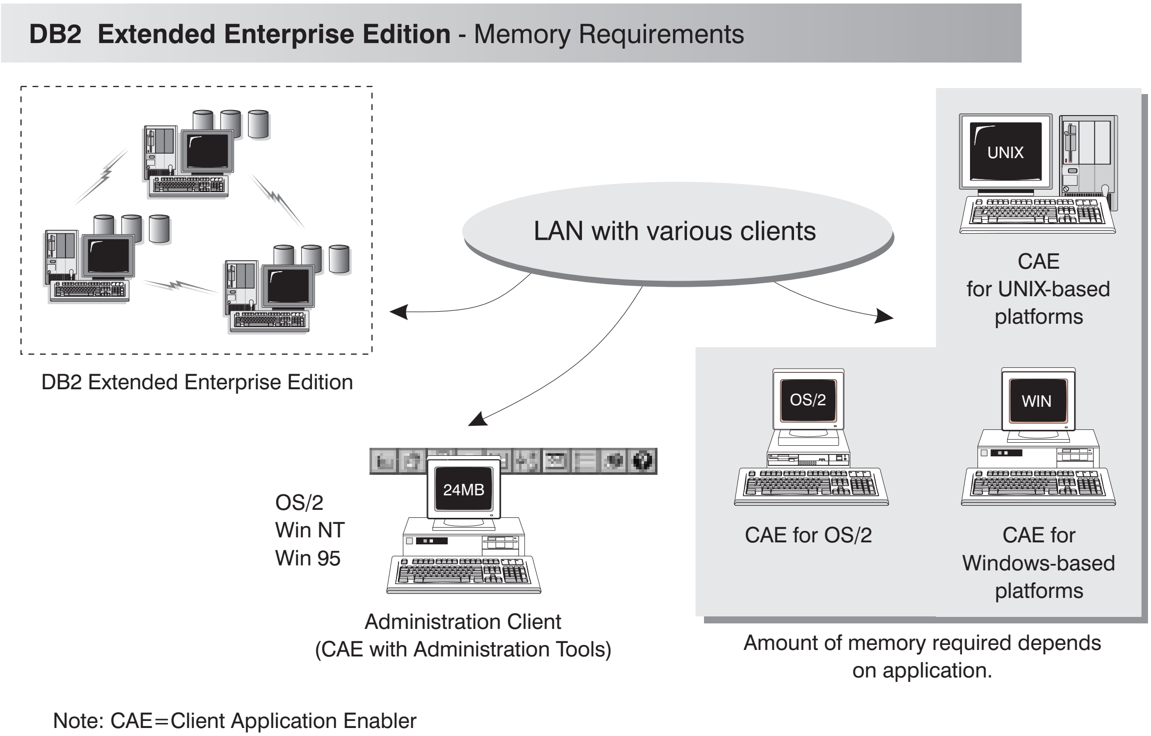

DB2 Extended Enterprise Edition - Memory Requirements
LAN with various clients
DB2 Extended Enterprise Edition
UNIX
CAE
for UNIX-based
platforms
OS/2
WIN
CAE for OS/2
CAE for
Windows-based
platforms
24MB
OS/2
Win NT
Win 95
Administration Client
(CAE with Administration Tools)
Amount of memory required depends
on application.
Note: CAE=Client Application Enabler

Figure 10. Configuration for Accessing DB2 Servers on the LAN

The amount of virtual memory (RAM) you need depends on the configuration that you want. Typically, you will require at least 256 MB of RAM per machine. If you have more than 32 database partition servers in the instance, or more than 30 clients, you will require at least 512 MB of RAM per machine. If your system contains more database partition servers, more users, or you have specific performance requirements, you may require more memory. Use these guidelines to determine the amount of memory necessary in your environment.

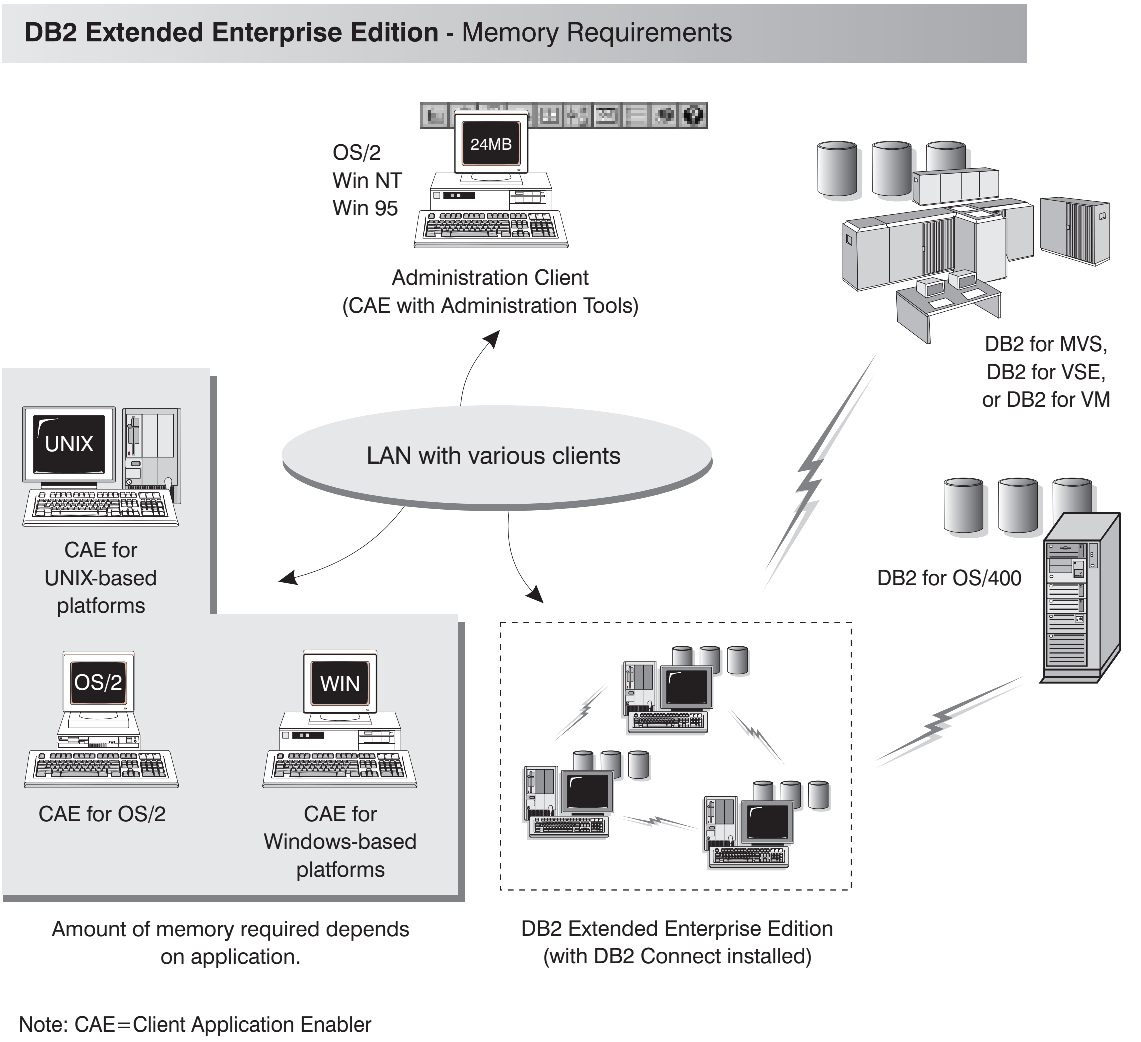
An administration client is shown in the diagram. This client can be an OS/2, Windows 95, or Windows NT workstation and can be used to manage DB2 servers on your network. Typically, 30 MB of memory is required to run all the graphical tools that you will need for an administration client.

For the DB2 clients in your environment, the amount of memory required depends on the database applications you are running.

Configuration for Accessing LAN and Host Data

If you want to access databases on both your local LAN and on host systems such as OS/390, OS/400, MVS, VM, or VSE, use the configuration shown in Figure 11 on page 38. You will need to install the DB2 Connect Support option of DB2 Extended Enterprise Edition.

Connect clients to DB2 Extended Enterprise Edition using one of the supported protocols: APPC, IPX/SPX, or TCP/IP. DB2 then connects to the appropriate host database management system using DRDA and either APPC or TCP/IP. (TCP/IP is supported on DB2 for OS/390 Version 5.1 or later.)



DB2 Extended Enterprise Edition - Memory Requirements
24MB
OS/2
Win NT
Win 95
Administration Client
(CAE with Administration Tools)
DB2 for MVS,
DB2 for VSE,
or DB2 for VM
LAN with various clients
UNIX
CAE for
UNIX-based
platforms
DB2 for OS/400
OS/2
WIN
CAE for OS/2
CAE for
Windows-based
platforms
Amount of memory required depends
on application.
DB2 Extended Enterprise Edition
(with DB2 Connect installed)
Note: CAE=Client Application Enabler

Figure 11. Configuration for Accessing LAN and Host Data

The amount of virtual memory (RAM) you need depends on the configuration that you want. Typically, you will require at least 256 MB of RAM per machine. If you have more than 32 database partition servers in the instance, or more than 30 clients, you will require at least 512 MB of RAM per machine. If your system contains more database partition servers, more users, or you have specific performance requirements, you may require more memory. Use these guidelines to determine the amount of memory necessary in your environment.

An administration client is shown in the diagram. This client can be on an OS/2, Windows 95, or Windows NT workstation and can be used to manage DB2 servers on your network. Typically, 30 MB of memory is required to run all the graphical tools you will need for an administration client.

For the DB2 clients in your environment, the amount of memory required depends on the database applications you are running.

Configuration for Accessing from the Web

If you want Internet clients to use Web browsers to access data in databases on both LAN and host-based DB2 servers, use the configuration shown in Figure 5 on page 16 or Figure 6 on page 18.

Follow the guidelines in “Configuration for Accessing DB2 Servers on the LAN” on page 35 and “Configuration for Accessing LAN and Host Data” on page 37 to determine the memory required on your DB2 server and administration client.

Disk Requirements

This section shows the *minimum* amount of disk space that is required to install DB2. It does not include the disk requirements necessary for the operating system, application development tools, and communications products. Consult each product's documentation for these values. Estimates for disk space requirements are listed here; the actual amounts required depend on the functions you are using. Use this information to plan for systems with a large number of concurrent clients or databases. For information about space requirements for data, refer to the *Administration Guide*.

Estimating Fixed Disk Requirements

To estimate the disk requirements for a particular configuration, add the recommended minimum disk sizes for the products and components you want to install. Include an allowance for your application data.

Server Components

Use Table 1 to estimate the amount of disk space you need to install DB2 and associated components on your operating system.

Table 1. Estimating Disk Requirements

	Recommended Minimum Disk (MB)
DB2 Universal Database Extended Enterprise Edition for AIX	
DB2 Extended Enterprise Edition	55 MB
• Online documentation in HTML format (English)	72 MB
• Far-East Code Page Conversion Support	5 MB
DB2 Universal Database Extended Enterprise Edition for Solaris	
DB2 Extended Enterprise Edition	54 MB
• Online documentation in HTML format (English)	72 MB
• Far-East Code Page Conversion Support	5 MB

Note: The disk amounts listed are approximations.

Client Components

Use Table 2 to estimate the amount of disk space you need on each of your client workstations.

Table 2 (Page 1 of 2). Estimating Disk Requirements for Client Components

	Recommended Minimum Disk (MB)
DB2 Client Application Enabler for AIX	
DB2 Client Application Enabler for AIX — all components	22 MB
DB2 Client Application Enabler for HP-UX	
DB2 Client Application Enabler for HP-UX — all components	18 MB
DB2 Client Application Enabler for Macintosh	
DB2 Client Application Enabler for Macintosh	8 MB
• Documentation	7 MB
DB2 Client Application Enabler for OS/2	
DB2 Client Application Enabler for OS/2 — this amount includes the following components: • Client Configuration Assistant • Graphical Tools • Control Center	55 MB 5 MB 5 MB 30 MB

Table 2 (Page 2 of 2). Estimating Disk Requirements for Client Components

	Recommended Minimum Disk (MB)
• Performance Monitor • Visual Explain	10 MB 5 MB
Online documentation in HTML format	25 MB
DB2 Client Application Enabler for SCO OpenServer Version 2	
DB2 Client Application Enabler for SCO OpenServer	16 MB
• Online documentation in HTML format	25 MB
DB2 Client Application Enabler for Silicon Graphics IRIX Version 2	
DB2 Client Application Enabler for Silicon Graphics IRIX	14 MB
• Documentation	19 MB
DB2 Client Application Enabler for SINIX Version 2	
DB2 Client Application Enabler for SINIX	16 MB
• Documentation	21 MB
DB2 Client Application Enabler for Solaris	
DB2 Client Application Enabler for Solaris — all packages	16 MB
DB2 Client Application Enabler for Windows 3.1x	
DB2 Client Application Enabler for Windows	10 MB
• Online documentation in HTML format	10 MB
DB2 Client Application Enabler for Windows 95 & Windows NT	
DB2 Client Application Enabler for Windows 95 & Windows NT — this amount includes the following components: • Graphical Tools (Client Configuration Assistant, Control Center, Performance Monitor, and Visual Explain)	54 MB 35 MB
Online documentation in HTML format	25 MB

Software Requirements

This section outlines the software required to run DB2 Universal Database and its associated products or components.

Server Product Requirements

Table 3 lists the operating system and communications software required for the server products.

Table 3 (Page 1 of 2). Software Requirements for DB2 Servers

Product	Hardware/Operating System	Communications
• DB2 Universal Database Extended Enterprise Edition (including DB2 Connect Enterprise Edition)	RISC System/6000 and the following: • AIX Version 4.1.4 or later	APPC, IPX/SPX, or TCP/IP • For APPC connectivity with two-phase commit, you require IBM Communications Server Version 4.1 or later for AIX in order to use the LU6.2 Syncpoint Manager function. If you do not need to use two-phase commit, IBM SNA Server/6000 Version 1.x or later is sufficient. • The AIX base operating system provides TCP/IP connectivity. For two-phase commit, TCP/IP connectivity is supported only on the AIX Version 4.1.4 or later server. • IPX/SPX connectivity is provided by: – IBM NetBIOS and IPX/SPX for AIX Version 2.1 – IBM NetWare for AIX Version 3.11B – AIX Connections feature of AIX 4.1.4 – AIX base operating system 4.1.4 or later **Notes:** 1. DB2 Connect requires SNA Server for AIX Version 2.1 or higher, or Bull DPX/20 SNA/20. 2. If you want to install the DRDA Application Server function of the DB2, you must install the Syncpoint Manager (SPM) and its prerequisites in order to use two-phase commit. 3. If you want to use DCE (Distributed Computing Environment) with Version 5 of the DB2 Universal Database products, you require a DCE product that is at OSF DCE level 1.1, which is provided by the AIX Version 4.1.4 operating system with its latest DCE PTF. For DB2 Connect support, you require DB2/MVS Version 5.1 plus its prerequisite, OS/390 DCE Base Services Version 3 for DCE support. With DB2 Connect, you must install DCE Directory Services on the client and the DRDA server. You do not need DCE installed on a DB2 Connect Enterprise Edition server. 4. If you plan to use the ADSTAR Distributed Storage Manager (ADSM) facilities for backup and restore of your databases, you require the ADSM Client Version 2.1.x.7. 5. If you want to use the Simple Network Management Protocol (SNMP) subagent, you require DPI 2.0 provided by IBM SystemView Agent.

Table 3 (Page 2 of 2). Software Requirements for DB2 Servers

Product	Hardware/Operating System	Communications
• DB2 Universal Database Extended Enterprise Edition (including DB2 Connect Enterprise Edition)	Solaris SPARCstation and the following: • Solaris Version 2.5.1 or later The following patches are required: • 101242 Rev.11 or higher • 103566 Rev.08 or higher • 103600 Rev.13 or higher • 103640 Rev.08 or higher	APPC, IPX/SPX, or TCP/IP • For APPC connectivity, you require SunLink SNA 9.0 and the following communication products: – SunLink P2P LU6.2 9.0 – SunLink PU2.1 9.0 – SunLink P2P CPI–C 9.0 • TCP/IP is provided with the Solaris base operating system. • IPX/SPX connectivity is provided with SolarNet PC Protocol Services 1.1 with IPX/SPX for the Solaris 2.x Operating Environment. **Notes:** 1. DB2 Connect for Solaris requires Solaris Version 2.5.1 or later and the following communication products: • SunLink P2P LU6.2 9.0 • SunLink SNA PU2.1 9.0 Server • SunLink P2P CPI-C 9.0 2. If you want to use DCE (Distributed Computing Environment) with Version 5 of the DB2 Universal Database products, you require Transarc DCE Version 1.1 for Solaris 2.5 and 2.5.1, patch level 18 or higher. With DB2 Connect, you must install DCE Directory Services on the client and the DRDA server. You do not need DCE installed on a DB2 Connect Enterprise Edition server. 3. If you plan to use the ADSTAR Distributed Storage Manager (ADSM) facilities for backup and restore of your databases, you require the ADSM Client Version 2.1.x.8.

Client Product Requirements

Table 4 lists the software requirements needed for the DB2 client products.

Table 4 (Page 1 of 4). Software Requirements for Clients

Component	Hardware/Operating System	Communications
• DB2 Client Application Enabler for AIX • DB2 Software Developer's Kit for AIX • Net.Data for AIX	RISC System/6000 and the following: • AIX Version 4.1.4 or later	APPC or TCP/IP • For APPC connectivity, you require IBM Communications Server for Version 4.1.4 or later for AIX • The AIX base operating system provides TCP/IP connectivity, if selected during install. **Notes:** 1. If you want to use DCE (Distributed Computing Environment), you require a DCE product that is provided by the AIX Version 4.1.4 operating system with its latest DCE PTF. 2. DB2 for AIX also has a server IPX communications manager.

Table 4 (Page 2 of 4). Software Requirements for Clients

Component	Hardware/Operating System	Communications
- DB2 Client Application Enabler for HP-UX - DB2 Software Developer's Kit for HP-UX	HP 9000 Series 700 or 800 system and the following: - HP-UX Version 10.10 or later The following patches are required: - For HP-UX Version 10.20 – PHSS_10556 – PHSS_10436 - For HP-UX Version 10.10 – PHSS_10437 - For systems with the ANSI C or C++ compilers: – PHSS_10261 – PHSS_7505 – PHSS_9096 for C++	APPC or TCP/IP - For APPC connectivity, you require either of the following: – HP-UX Version 10.10 for SNAplusLink Version A.10.10 – HP-UX Version 10.10 for SNAplusAPI Version A.10.10 - The HP-UX base operating system provides TCP/IP connectivity. **Notes:** 1. To support database clients using IPX/SPX, the following IPX/SPX software must be installed: - Netware Version 3.12 for the HP 9000 2. If you want to use DCE (Distributed Computing Environment), you require a DCE product that is provided by the HP-UX Version 10 or later base operating system.
- DB2 Client Application Enabler for Macintosh - DB2 Software Developer's Kit for Macintosh	- System 7.5 or later	TCP/IP - For TCP/IP connectivity, you require MacTCP Version 2.0.6. **Note:** If you are running in a 68K environment, you need to install Apple Shared Library Manager (ASLM) Version 2.0 or later.
- DB2 Client Application Enabler for OS/2 - DB2 Software Developer's Kit for OS/2 - Net.Data for OS/2	- OS/2 Warp Version 3 or Version 4 - OS/2 Warp Connect Version 3 - OS/2 Warp Server Version 4 - OS/2 Warp Server Advanced V4 - OS/2 Warp Server Advanced V4 with SMP Feature	APPC, IPX/SPX, Named Pipes (Local), NetBIOS, or TCP/IP - For APPC connectivity, you require IBM Communications Manager Version 1.x or later. - For NetBIOS connectivity, you require IBM NTS/2 Version 1.0, IBM Communications Manager Version 1.1, or IBM OS/2 LAN Requester. - For IPX/SPX connectivity, you require the Novell NetWare client for OS/2 Version 2.10 or later. - For TCP/IP connectivity, you require IBM TCP/IP Version 2.0 or later. - The OS/2 base operating system provides Named Pipes (Local) connectivity. Named Pipes is supported in DOS and WIN-OS/2 sessions. **Notes:** 1. OS/2 Warp Server provides the SystemView LAN client that SystemView LAN needs for installing clients and servers. 2. Net.Data requires OS/2 Warp Version 3 or later and a Web server such as IBM's Internet Connection Server. 3. For DCE Cell Directory Services Support (CDS) for DB2 Client Application Enabler for OS/2, you must install IBM Distributed Computing Environment Cell Directory Service client, Version 2.10, on each client workstation.

Table 4 (Page 3 of 4). Software Requirements for Clients

Component	Hardware/Operating System	Communications
• DB2 Client Application Enabler for SCO OpenServer • DB2 Software Developer's Kit for SCO OpenServer • Net.Data for SCO OpenServer	Intel-based personal computer supported by SCO OpenServer and the following: • SCO OpenServer Version 5.0.2 or later	APPC, IPX/SPX, or TCP/IP • For APPC connectivity, you require Interface Systems CLEO SNA Version 4.1.1.1. • The SCO OpenServer base operating system provides IPX/SPX and TCP/IP connectivity.
• DB2 Client Application Enabler for Silicon Graphics IRIX • DB2 Software Developer's Kit for Silicon Graphics IRIX	• Silicon Graphics IRIX, Version 5.3 or later.	TCP/IP **Note:** The Silicon Graphics IRIX base operating system provides TCP/IP connectivity.
• DB2 Client Application Enabler for SINIX • DB2 Software Developer's Kit for SINIX	Siemens Nixdorf RM200 or RM400 or RM600 system and the following: • SINIX-N or SINIX-Y Version 5.42 or later or Reliant UNIX	APPC or TCP/IP • For APPC connectivity, you require the following: – SINIX TRANSIT-SERVER V3.3, SNA Communication Server – SINIX TRANSIT-CLIENT V3.3 SNA Communication Server / Local Function – SINIX TRANSIT-CPIC V3.3 SNA LU6.2 Communication and CPI-C. • The SINIX base operating system provides TCP/IP connectivity.
• DB2 Client Application Enabler for Solaris • DB2 Software Developer's Kit for Solaris • Net.Data for Solaris	Solaris SPARCstation and the following: • Solaris Version 2.5.1 or later The following patches are required: • 101242 Rev.11 or higher • 103566 Rev.08 or higher • 103600 Rev.13 or higher • 103640 Rev.08 or higher	APPC or TCP/IP • For APPC connectivity, you require SunLink SNA 9.0 and the following: – SunLink P2P LU6.2 9.0 – SunLink PU2.1 9.0 – SunLink P2P CPI–C 9.0 • The Solaris base operating system provides TCP/IP connectivity. **Notes:** 1. If you want to use DCE (Distributed Computing Environment), you require a DCE product with Transarc DCE Version 1.1 for Solaris 2.5 and 2.5.1, patch level 18 or higher. 2. DB2 for Solaris also has a server IPX communications manager.
• DB2 Client Application Enabler for Windows, (Windows 3.1 and Windows for Workgroups) • DB2 Software Developer's Kit for Windows	• Microsoft Windows 3.1 • Windows for Workgroups 3.11	IPX/SPX, NetBIOS, or TCP/IP • For IPX/SPX connectivity, you require Novell NetWare client for Windows, which comes with Novell Netware Server Version 3.x or Version 4.x; or Novell NetWare Workstation for Windows Version 1.0 or later with the latest update kit installed. • For NetBIOS connectivity, you require the LAN Adapter and Protocol Support in any of these products: – IBM LAN Support Program Version 1.2.1 or later – DOS LAN Services (LAN Server 4.0) – Banyan Vines NetBIOS Version 5.52 – NetBEUI (available with Microsoft Windows for Workgroups 3.11) • For TCP/IP connectivity, you require a TCP/IP stack that is compliant with the Windows Sockets Version 1.1 specification.

Table 4 (Page 4 of 4). Software Requirements for Clients

Component	Hardware/Operating System	Communications
• DB2 Client Application Enabler for Windows (WIN-OS/2) • DB2 Software Developer's Kit for Windows (WIN-OS/2)	• OS/2 Warp Version 3 or Version 4 • OS/2 Warp Connect Version 3 • OS/2 Warp Server Version 4 • OS/2 Warp Server Advanced V4 • OS/2 Warp Server Advanced V4 with SMP Feature	NetBIOS, IPX/SPX, TCP/IP, or Named Pipes (Local) • For NetBIOS connectivity, you require IBM NTS/2 Version 1.0, IBM Communications Manager Version 1.1, or IBM OS/2 LAN Requester. • For IPX/SPX connectivity, you require Novell NetWare client for OS/2 Version 2.10 or later with virtual session support. • For TCP/IP connectivity, you require IBM TCP/IP for OS/2 with the Windows Access kit.
• DB2 Client Application Enabler for Windows 95 • DB2 Software Developer's Kit for Windows 95	• Windows 95 4.00.950 or later	IPX/SPX, Named Pipes, NetBIOS, or TCP/IP • The Windows 95 base operating system provides NetBIOS, IPX/SPX, TCP/IP, and Named Pipes connectivity.
• DB2 Client Application Enabler for Windows NT • DB2 Software Developer's Kit for Windows NT • Net.Data for Windows NT	• Windows NT Version 3.51 • Windows NT Version 4.0 or later	APPC, IPX/SPX, Named Pipes, NetBIOS, or TCP/IP • The Windows NT base operating system provides NetBIOS, IPX/SPX, TCP/IP, and Named Pipes connectivity. • For APPC connectivity, you require one of the following products: – IBM Communications Server for Windows NT Version 5.0. – Microsoft SNA Server Version 2.11 or later on the LAN. You do not have to install the SNA Server on the same workstation as the DB2 Client Application Enabler for Windows NT. – PCOMM AS/400 and 3270 V4.1 or later – Wall Data Rumba **Note:** If you have the IBM Antivirus program installed on your operating system, it must be Version 3.0 or later.

Possible Client-to-DB2 Server Connectivity Scenarios

The following table shows the communication protocols that can be used when connecting a specific DB2 Client to a specific DB2 Server.

Table 5. Possible Client-to-Server Connectivity Scenarios

Clients	DB2 Servers	
	AIX	**Solaris**
AIX	APPC TCP/IP	APPC TCP/IP
HP-UX	APPC TCP/IP	APPC TCP/IP
Macintosh	TCP/IP	TCP/IP
OS/2	APPC IPX/SPX[1,2] TCP/IP	APPC IPX/SPX[1] TCP/IP
SCO OpenServer	APPC IPX/SPX[1] TCP/IP	APPC IPX/SPX[1] TCP/IP
Silicon Graphics IRIX	APPC TCP/IP	APPC TCP/IP
SINIX	APPC TCP/IP	APPC TCP/IP
Solaris	APPC TCP/IP	APPC TCP/IP
Windows 3.1x	IPX/SPX[1,2] TCP/IP	IPX/SPX[1] TCP/IP
Windows 95	IPX/SPX[1] TCP/IP	IPX/SPX[1] TCP/IP
Windows NT	APPC IPX/SPX[1] TCP/IP	APPC IPX/SPX[1] TCP/IP

[1] Direct Addressing

[2] File Server Addressing

Understanding System Administrative Authority

System Administrative (SYSADM) authority is required in order to perform administration tasks such as cataloging, starting the database manager, or creating the sample database. By default, any user belonging to the primary group of the instance owner has System Administrative authority on the instance created during the installation. Throughout the book, IDs that have this authority are referred to as having SYSADM authority. To change the group that has SYSADM authority on the default instance, see "Working with the System Administrative Group" on page 170.

Preparing to Install the DB2 Products

Before you install DB2 Universal Database products, it is important to know which products and components you must install and which are optional. "Selecting Products and Components" lists these DB2 products and their components.

Selecting Products and Components

Table 6 lists the DB2 products components that you can select to install on your system.

Table 6 (Page 1 of 2). Components for DB2 Products

Product / Component Description	**DB2 Universal Database Extended Enterprise Edition**	**DB2 CAE**
DB2 Client Application Enabler	√√	√√
Open Database Connectivity (ODBC) support	√	√
Java Database Connectivity (JDBC) support	√	√
Code Page Conversion Tables - Unicode Support	√√	√
Administration Server	√√	
DB2 Run-time Environment	√√	
DB2 Engine	√√	
DB2 Parallel Extensions	√√	

Table 6 (Page 2 of 2). Components for DB2 Products

Product / Component Description	**DB2 Universal Database Extended Enterprise Edition**	**DB2 CAE**
DB2 Connect	√√	
DB2 Sample Database Source	√√	
Communication Support for DRDA Application Server	√√	
Communication Support for IPX/SPX	√√	
Communication Support for SNA	√√	
Communication Support for TCP/IP	√√	
Code Page Conversion Tables - Japanese	√	
Code Page Conversion Tables - Korean	√	
Code Page Conversion Tables - Simplified Chinese	√	
Code Page Conversion Tables - Traditional Chinese	√	
License Support for DB2 Universal Database Extended Enterprise Edition	√√	
DB2 Replication	√	
DB2 Product Messages (non-English)[1]	o	o
DB2 Product Library (HTML)[1]	o	o

√√ This is a required component that must be installed.

√ This is a recommended component that is installed by default. You can choose not to install this component.

o This is an optional component that is not installed by default. If you want to install it, you must select it.

Note: There is a separate component of the DB2 Product Messages and the DB2 Product Library for each locale.

Chapter 5. Installing and Configuring DB2 Universal Database Extended Enterprise Edition for AIX

This section describes how to install DB2 Universal Database Extended Enterprise Edition for AIX.

To install DB2 Universal Database Extended Enterprise Edition for Solaris, go to page Chapter 6.

To install DB2 clients, go to Chapter 8, "Installing DB2 Clients" on page 111.

Installation Steps

To install DB2, perform the following steps:

1. Prepare for installation.
2. Install DB2. As part of this installation step, the following installation activities are optional:
   - **a** Install the DB2 product online documentation.
   - **b** Install the DB2 product translated messages.
3. Enable the Fast Communication Manager (FCM).
4. Create a DB2 instance.
5. Create the node configuration file.
6. Make DB2 Extended Enterprise Edition available.
7. Create the Administration Server.
8. Set up Discovery.
9. Create links for DB2 files (optional).
10. Set up the instance environment.

You can install DB2 using any of the following methods:

Using dsh
Allows you to simultaneously install DB2 on multiple RS/6000 SP nodes.

Using SMIT
Allows you to install a subset of products and filesets on a single machine. If you want to install only a selected set of DB2 filesets or components, see “Install Products or Filesets Using SMIT” on page 57.

Installing software bundles using SMIT
A software bundle contains a list of filesets that are suited for a particular use. Installation, although easier with software bundles, only occurs on a single machine.

If you prefer to use the **installp** command, you must first mount the DB2 CD-ROM. You also need to mount the CD-ROM if you want to view the *Installation Notes* or print the postscript manuals prior to installation.

Step 1. Prepare for Installation

To prepare your system for installation, perform the following steps:

1 Prepare a list of the machine names on which your database is to reside. You can use the **uname -n** command to find out the name of a machine. (In the discussion that follows, the term *machine* also means an RS/6000 SP node.)

2 Generate a list of each machine's IP address. You can use the **host** *hostname* command to resolve the IP address of a machine; for example, **host machine1**.

3 As a user with *root* authority, create or assign groups and user IDs.

If you want to use an existing user or group ID, you do not need to create new ones at this time. Instead, you can proceed to the next step.

Create a group that will be the primary group for the user ID that will be the instance owner. When the instance is created, the instance owner and any other user who is a member of this group will have system administrative (SYSADM) authority. The group must be created on every machine that is to be used as a database partition server. The group ID (GID) of the group must be the same on each machine.

Next, create a user ID that will be the instance owner. This user ID will be the name of the instance. The user ID must be created on every machine that is to be used in the partitioned database system. The user ID of the user must be the same on every machine. Make this user's primary group the group that you just created. The instance owner user ID, and any other user who is a member of this group, will be automatically given SYSADM control for the instance.

In the following example, the instance user ID is `db2inst1` and the UID `1004`. The SYSADM group ID is `dbadmin1` and the GID `999`.

> Dedicate the instance owner user ID to that instance's use only. This allows for easier error recovery if a system error occurs.

a Use the **mkgroup** command to create groups, and the **mkuser** command to create users. For example, to create a user called `db2inst1` with user ID *1004*, which will use `dbadmin1` with group ID *999* as its primary group and use */home/db2inst1* as its home directory, type:

```
mkgroup id=999 dbadmin1
mkuser id=1004 pgrp=dbadmin1 groups=dbadmin1 home=/home/db2inst1 db2inst1
passwd db2inst1
```

The group must be created on every machine that is to be used as a database partition server. The group ID (GID) of the group must be the same on each machine. You must define the same user and user ID number on each machine.

b Create a group and user ID for fenced user defined functions and stored procedures.

For example:

```
mkgroup id=991 db2fadm1
mkuser id=1001 pgrp=db2fadm1 groups=db2fadm1 home=/home/db2fenc1 db2fenc1
passwd db2fenc1
```

Fenced user defined functions (UDFs) and stored procedures will execute under this user and group. The group must be the primary group of the user. The user for fenced UDFs and stored procedures is specified as a parameter of the instance creation script. The group for fenced UDFs and stored procedures is implicitly set to the primary group of this specified user (`db2fenc1`).

The group must be created on every machine that is to be used as a database partition server. The group ID (GID) of the group must be the same on each machine. You must define the same user and user ID number on each machine.

4 Export the home directory of the instance (*INSTHOME*) for the `db2inst1` user ID from one machine and use NFS to mount it on every other machine that will be part of the partitioned database system. Ensure that every machine has *write* access to this directory.

For better performance, if you are installing on an RS/6000 SP system, you should have the *INSTHOME* directory on one of the SP nodes (and not on the control workstation). Then NFS mount the *INSTHOME* directory on the other SP nodes over the high speed switch.

The *INSTHOME* directory must *not* be mounted with the *nosuid* NFS mount option.

> You should not mount the *INSTHOME* directory by using the **AMD** or **automounter** utilities. These utilities can cause mounting or locking problems.

5 Check whether NFS is running on the system by issuing the **lssrc -g nfs** command.

For additional information about NFS, refer to the NFS documentation.

> If you are installing a small to intermediate size system, the number of network file system daemons (NFSDs) on the machine that has the instance owner's home directory should be close to:
>
> ```
> # of biod on a machine * # of machines in the instance
> ```
>
> According to the above formula, on a four machine system with 10 biod processes, you would use 40 NFSDs.
>
> If you are installing a larger system, you can have up to 120 NFSDs on the machine.
>
> **Note:** It is recommended to run 10 biod processes on every machine.

6 Ensure that the `rpc.statd` and `rpc.lockd` processes are active on each machine. Issue the **ps -ef | grep rpc** command, and check for the names of the two processes in the output.

7 Ensure that */etc/rc.nfs* entry is included in the */etc/inittab* file using the **lsitab rcnfs** command.

8 Set the AIX `maxuproc` (maximum number of processes per user) device attribute to *500* on all the machines that will be in the partitioned database system by issuing the following command:

```
chdev -l sys0 -a maxuproc='500'
```

9 Create a local file system that is locally mounted on each machine; for example, */database*. This file system will function as a mount point for other files in the file system.

> Databases must be created on locally attached disks, and *not* on NFS-mounted file systems. Creating databases on an NFS-mounted file system is not supported.

10 On all machines, create a subdirectory in the */database* directory owned by the user `db2inst1` in the group `dbadmin1`. For this example, the subdirectory is called */database/db2inst1*.

When installing DB2 on an RS/6000 SP that is running AIX, you should ensure that the following network-related parameters are set on all of the SP nodes:

```
thewall       = 16384
sb_max        = 1310720
rfc1323       = 1
tcp_sendspace = 655360
tcp_recvspace = 655360
udp_sendspace = 65536
udp_recvspace = 655360
```

Note: You can use the **no -a** command to view the current setting for network parameters. The above values are minimum values for the parameters. If you are already setting any of the network-related parameters to a higher value, do not set the parameter to the lower listed value.

In addition, if you have the high speed switch, consider increasing the *spoolsize* and *rpoolsize* for *css0*. For example, to set the *spoolsize* and *rpoolsize* parameters to *8388608*, issue the following commands:

```
/usr/lpp/ssp/css/chgcss -l css0 -a spoolsize=8388608
/usr/lpp/ssp/css/chgcss -l css0 -a rpoolsize=8388608
```

To view the configuration for *css0*, issue the **lsattr -l css0 -E** command.

You should only use the following method to set the network-related parameters if you are not using the */tftpboot/tuning.cst* file to tune the system.

A sample file is provided for updating the network-related parameters. The file name is:

```
DB2DIR/misc/rc.local.sample
```

where:

DB2DIR = /usr/lpp/db2_05_00

1 Copy the sample file to the */etc* directory and make it executable by `root`. For example:

```
cp DB2DIR/misc/rc.local.sample /etc/rc.local
chown root:sys /etc/rc.local
chmod 744 /etc/rc.local
```

2 Review the */etc/rc.local* file and update it if necessary.

3 Add an entry to the */etc/inittab* file so that the */etc/rc.local* script is executed whenever the machine is rebooted. You can use the **mkitab** command to add an entry to the */etc/inittab* file as in the following example:

```
mkitab "rclocal:2:wait:/etc/rc.local > /dev/console 2>&1"
```

4 To update the network parameters without rebooting the machine, issue the **/etc/rc.local** command.

For more information about these and other AIX operating system parameters, refer to the *Performance and Tuning Guide*.

Step 2. Install DB2

The DB2 CD-ROM contains several products. Select the products you are licensed to install. *Your Proof of Entitlement* and *License Information* booklet identify the products for which you are licensed.

Go to the installation method that you would like to use to install DB2.

- “Install the Product or Filesets Using the dsh Command.”
- “Install Products or Filesets Using SMIT” on page 57.
- “Install Software Bundles Using SMIT” on page 58.

Install the Product or Filesets Using the dsh Command

You can use the **dsh** command to install DB2 Extended Enterprise Edition on an RS/6000 SP machine that is running AIX. To use the **dsh** command:

1. On the SP control workstation, create a file that lists the host names of the machines on which you want to install DB2 Extended Enterprise Edition. For example, assume that you want to install this product on three machines, `machine1`, `machine2`, and `machine3`, and you are using a file called *machine.txt*, the file contents would be:

   ```
   machine1
   machine2
   machine3
   ```

2. On the control workstation, enter the following command:

   ```
   export WCOLL=/machine.txt
   ```

3. Issue the **hostlist** or the **dsh -q** command to ensure that you have the correct set of machines that you want to run the **dsh** command on.

4. Mount the CD-ROM drive on the control workstation; for example, */cdrom*.

5. Export the */cdrom* file system to all the machines as follows:

   ```
   exportfs -o ro /cdrom
   ```

6. Create a directory on each machine as follows (this directory will be used to mount the */cdrom* file system):

   ```
    dsh mkdir /cdrom
   ```

7 Mount the */cdrom* file system from the control workstation as follows:

```
dsh mount control_workstation:/cdrom /cdrom
```

where *control_workstation* is the name of the control workstation.

After this command completes, all machines will be able to access the CD-ROM.

8 Issue the **dsh** command to install the DB2 package that you want on all machines; for example:

```
dsh installp -qagXd /cdrom pkgname
```

where *pkgname* is the name of the package that you want to install.

For example, to install DB2 Extended Enterprise Edition on every machine, you can use the following command:

```
dsh installp -qagXd /cdrom db2_05_00.xsrv
```

For package names, see Appendix D, “Contents of the DB2 Products” on page 309.

Now that you have finished installing DB2, go to “Install the DB2 Product Library (Optional)” on page 59 and “Install the DB2 Product Messages (Optional)” on page 60 to optionally install the product documentation and messages. If you do not want to install product documentation or messages, proceed to the next step.

Install Products or Filesets Using SMIT

To install DB2 Extended Enterprise Edition using SMIT, perform the following steps:

1 Log in as root.

2 Mount the DB2 CD-ROM.

3 Enter the **smit install_latest** command. The Software Installation and Maintenance menu opens.

4 Specify the *INPUT device/directory* for DB2 and press Enter.

For example, if */cdrom* is the CD-ROM mount directory, enter the following as the INPUT device:

```
/cdrom/db2
```

5 Press F4 to display a list of software to install. For a detailed description of the software packages, refer to Appendix D, “Contents of the DB2 Products” on page 309 for more information.

6 Press F7 to mark one or more packages and filesets for installation.

Note: You must select the **License Support for DB2 Extended Enterprise Edition** fileset to install DB2 Extended Enterprise Edition.

7 Press Enter to install DB2.

8 Press F10 to exit when the command displays **OK**.

After the installation is complete, DB2 will be installed in the */usr/lpp/db2_05_00* directory on that machine; now you must repeat the above steps on all machines.

Now that you have finished installing DB2, go to “Install the DB2 Product Library (Optional)” on page 59 and “Install the DB2 Product Messages (Optional)” on page 60 to optionally install the product documentation and messages. If you do not want to install product documentation or messages, proceed to the next step.

Install Software Bundles Using SMIT

The following procedure shows how to install DB2 Extended Enterprise Edition using the *Software Bundles* feature. A software bundle is a collection of filesets. When you install a software bundle, every fileset in that bundle is installed.

1 Log in as root.

2 Insert and mount the DB2 CD-ROM in the drive.

3 To proceed directly to the automatic bundle processing application, issue the **smit easy_install_bundle** command.

4 Specify the installation device or directory for the installation media by pressing F4 to display a list.

5 Select the CD-ROM drive in which you placed the DB2 CD-ROM, then press Enter.

6 If you did not install the DB2 Media-Defined software bundle on this system, use the following procedure to install it:

   a In the **Bundle to Install** menu, select the **Media-Defined** option, then press Enter.

   b In the **Install Bundle Contents** menu, press Enter to continue with the installation. Messages may appear to indicate the status of the installation. Press Enter to continue.

   c Press F3 to return to the **Install Bundle Contents** menu.

   d Press F4 to re-display the list of software bundles.

7 From the displayed menu, select one of the following software bundles:

- DB2V5-UDB-Ext-Enterprise
- DB2V5-Client

Notes:

a. Every required and recommended fileset for a product is installed with its bundle.

b. For a list of filesets included in a product, see Table 6 on page 48.

8. Press Enter to start the installation.

   Messages will appear to indicate the status of the installation. Press Enter to continue, or F3 if you want to cancel.

9. Repeat the steps in this section if you want to install software bundles on other machines.

10. Repeat the steps on every machine that you want to participate in the partitioned database system.

Now that you have finished installing DB2, go to “Install the DB2 Product Library (Optional)” and “Install the DB2 Product Messages (Optional)” on page 60 to optionally install the product documentation and messages. If you do not want to install product documentation or messages, proceed to the next step.

Install the DB2 Product Library (Optional)

When you install DB2 using SMIT, the DB2 Product Library (HTML) is not automatically installed.

To install the DB2 Product Library, do the following:

1. Log in as root.

2. Mount the DB2 CD-ROM.

3. Enter the **smit install_latest** command. The Software Installation and Maintenance menu opens.

4. Enter the directory for the **INPUT device/directory for software**. For example, if */cdrom* is the CD-ROM mount directory, enter the following as the INPUT device:

   ```
   /cdrom/db2
   ```

5. Press F4 to display a list of software to install. For a detailed description of the software packages, refer to Appendix D, “Contents of the DB2 Products” on page 309 for more information.

6. Select the fileset for the DB2 Product Library (HTML) that you want to install. A separate fileset exists for every language in which DB2 documentation is translated.

Note: Not every manual is translated into multiple languages. The *DB2 Product Library (HTML) - English* fileset contains a complete set of books in English.

The DB2 Product Library (HTML) fileset is listed as follows:

```
DB2 Product Library (HTML) - language
```

7. Press ENTER to start the installation.

8. Press F10 to exit SMIT when the command completes.

9. Run the following command to uncompress and untar the HTML files:

   ```
   /usr/lpp/db2_05_00/doc/db2insthtml locale
   ```

   where *locale* is the desired locale; for example, locale=en_US for English.

10. From your Web browser, open the following file URL to view DB2 online manuals:

    ```
    file:/usr/lpp/db2_05_00/doc/en_US/html/index.htm
    ```

To recover some disk space, you may want to remove the compressed tar files. It is recommended that you do not just remove the compressed tar HTML files; instead, you should remove the fileset for the DB2 Product Library.

After the fileset is removed, only compressed tar files will be removed. You will still be able to view the online DB2 books.

Install the DB2 Product Messages (Optional)

When you install DB2 using SMIT, DB2 product messages for locales, other than en_US (English - ISO 88591), are not automatically installed.

You should install messages on every machine that will function as a database partition server.

To install DB2 product messages in other locales, do the following:

1. Log in as root.

2. Mount the DB2 CD-ROM.

3. Enter the **smit install_latest** command. The Software Installation and Maintenance menu opens.

4. Enter the directory for the **INPUT device/directory for software**. For example, if */cdrom* is the CD-ROM mount directory, enter the following as the INPUT device:

   ```
   /cdrom/db2
   ```

5 Press F4 to display a list of software to install. For a detailed description of the software packages, refer to Appendix D, “Contents of the DB2 Products” on page 309 for more information.

6 List the software to be installed using the **SOFTWARE to install** option.

7 Select the messages options for the locales you want to install.

Step 3. Enable the Fast Communication Manager

> If you are using Network Information Services (NIS), NIS+, or any other tools to manage system configuration files, you must perform this step on the master server. After you have completed the following, ensure that the changes have taken effect on all DB2 servers.

To enable FCM communications between database partition servers, you must create a service entry for each logical node in the */etc/services* file. The service name maps the database partition server at the machine to the actual TCP/IP port that is used. The syntax of a service entry is as follows:

```
DB2_instance port/tcp #comment
```

where:

DB2_*instance* Is the name of the DB2 Extended Enterprise Edition instance. All characters in the name must be lowercase. Assuming an instance name of `db2inst1`, you would specify `DB2_db2inst1`.

port/tcp Is the TCP/IP port that you want to reserve for the database partition server (node).

comment Any comment that you want to associate with the entry. The comment must be preceded by a pound sign (#).

You must ensure that the entry defined above is the same in all */etc/services* files on all machines.

As an example, suppose you wanted to make an entry in */etc/services* for the instance `db2inst1`, and have it assigned to the TCP/IP port *9000* (indicating that there is only one node on every machine in the instance). The service name is:

```
 DB2_db2inst1     9000/tcp
```

If you want a single machine to have multiple logical nodes for an instance, you will need to allocate multiple ports for FCM to use; one port is needed for each *logical* node. Enter two lines in the */etc/services* file to indicate the number of ports that you are allocating. The first line specifies the first port, while the second line indicates the end of the block of ports. In the following example, five ports are allocated for the instance. This means that no machine in the instance `db2inst1` is running more than five logical nodes.

```
DB2_db2inst1       9000/tcp
DB2_db2inst1_END   9004/tcp
```

You must ensure that the number of ports allocated in the file is either greater than or equal to the largest number of multiple logical nodes on any machine in the instance. When allocating ports, ensure that you account for any machines that can be used as a backup (for example, if you use High Availability (HA)).

Note: Specify the `END` keyword in uppercase only. Also ensure that you include both underscore (_) characters.

Step 4. Create a DB2 Instance

A DB2 instance is an environment where you store data and run applications. Use the **db2icrt** command to create an instance. You must run this command on the machine where you have DB2 physically installed and have *root* authority. For more information about database instances, see Chapter 17, “Working with Instances” on page 173.

It is possible to add an instance to any machine. If you use different machines to create instances, then the list of instances that is stored on any machine may not be a complete list. We recommend creating every instance on one machine, so that the list of instances is complete on that machine. We also recommend that this node be the Administration Server node.

The **db2icrt** command is in the *DB2DIR/instance* directory, where `DB2DIR` is `/usr/lpp/db2_05_00`.

Run the **db2icrt** command from the machine where the home directory of the instance (*INSTHOME*) physically resides. Because the *INSTHOME* directory is NFS-mounted on all machines, you only have to run the **db2icrt** command on one machine.

The syntax of the **db2icrt** command is:

```
►►──db2icrt──[-h| -?]──[-d]──[-a AuthType]────[-u FencedID] ───────►

►─────InstName─────────────────────────────────────────────────►◄
```

where:

-h | -? Display a help menu for this command.

-d Sets the debug mode that you can use for problem determination.

-a *AuthType* Is an optional parameter that specifies the authentication type for the instance. Valid authentication types are SERVER, CLIENT, and DCS. If the *-a* parameter is not specified, the authentication type defaults to SERVER, if a DB2 server is installed. Otherwise, the *AuthType* is set to CLIENT.

Notes:

1. All databases in the instance have the same authentication type.
2. DCE authentication is not valid for this command; however, you can enable DCE authentication for an instance. For more information, refer to the *Administration Guide*.

-u *FencedID* Is the user under which the fenced UDFs and stored procedures will execute. This is not required if you install the DB2 Client Application Enabler or the DB2 Software Developer's Kit. For other products, this is a required parameter.

Note: *FencedID* may not be *root* or *bin*.

InstName Is the login name of the instance owner.

When you create an instance, ensure that its primary and secondary groups are different from the Administration Server's primary (SYSADM) group. When you create an instance on the same machine as the Administration Server, its SYSADM group is automatically added to the secondary group list of the Administration Server so that you can use the Control Center to perform administration tasks on that instance.

Examples:

- To create an instance for a DB2 server, you can use the following command:

```
db2icrt -u db2fenc1 db2inst1
```

- To create an instance for a DB2 client, you can use the following command:

```
db2icrt db2inst1
```

When an instance is created, its name is also added to the list of instances on the system.

The **db2icrt** command creates the *INSTHOME/sqllib* directory, where *INSTHOME* is the home directory of the instance owner.

> To avoid a potential loss of data if an instance is deleted, you should not create user files or directories under the *INSTHOME/sqllib* directory, other than those created by DB2. Exceptions are:
>
> 1. If your system supports fenced user defined functions and fenced stored procedures, put the fenced applications in the *INSTHOME/sqllib/function* directory.
> 2. The location of the *db2nodes.cfg* file. This file should be put in the *INSTHOME/sqllib* directory, where *INSTHOME* is the home directory of the instance owner.

Step 5. Create the Node Configuration File

You must create the node configuration file, *INSTHOME/sqllib/db2nodes.cfg*, before you can start the partitioned database system. This file contains configuration information for all database partition servers in an instance, and is shared by all database partition servers for that instance.

The file contains one line for each database partition server that belongs to an instance. Each line has the following format:

```
nodenum hostname [logical_port [netname]]
```

Tokens are delimited by blanks. The variables are:

nodenum
The unique node number to identify the database partition server. The number can be from 0 to 999. Node numbers must be in ascending sequence. You can have gaps in this sequence.

When a node number is assigned, it cannot be changed. Otherwise, the information in the partitioning map (which details how data is partitioned) would be compromised.

If you drop a database partition server, its node number can be used again on a new database partition server that you add.

The node number is used to generate a node name in the database directory during the **create database** command. It has the format:

```
NODEnnnn
```

where *nnnn* is the node number, which is left-padded with zeros.

hostname
The TCP/IP hostname of the IP address which is used by the FCM for internal communications, unless *netname* is specified. If *netname* is specified, the FCM uses *netname*, and *hostname* is only used for the **db2start**, **db2stop**, and **db2_all** commands.

logical_port
This parameter is optional, and specifies the logical port number for the database partition server. This number is used with the database manager instance name to identify a TCP/IP service name entry in the */etc/services* file.

The combination of the IP address and port number is used as a well-known address, and must be unique among all applications to support communication connections between database partition servers.

For each *hostname*, one *logical_port* must be either 0 or blank (this defaults to 0). The database partition server associated with the *logical-port 0* is the default database partition server on the machine to which clients connect. This default database partition server is known as the *coordinator node*. You can override this with the *DB2NODE* environment variable, or with the **set client** command.

If you have multiple logical nodes, (that is, more than one *nodenum* for a machine), you must assign *logical_port* numbers to the logical nodes.

netname
This parameter is optional, and is used to support a machine that has more than one active TCP/IP interface, each with its own hostname.

This parameter only applies to those machines that have a primary *hostname* that is either an Ethernet or a token-ring name, and you want to have DB2 Extended Enterprise Edition use an alternative name (for example, a switch name). If you do not specify a value for this parameter, its value is that of the *hostname* parameter.

Notes:

1. If you specify a *netname*, you must specify a value for *logical_port*.
2. The *netname* is commonly referred to as a switch name.
3. If you want to use the **autoloader** command, the *netname* must be specified.

The following is an example of a possible node configuration file on which SYSEN1 has multiple TCP/IP interfaces, two logical nodes, and uses SYSSW1 as the DB2 Extended Enterprise Edition interface. It also shows the node numbers starting at 1, and a gap in the *nodenum* sequence:

```
nodenum   hostname   logical-port   netname

1         SYSEN1     0              SYSSW1
2         SYSEN1     1              SYSSW1
4         SYSEN2     0
5         SYSEN3
```

The logical port for node 5 is assumed to be 0.

If you are using an RS/6000 configuration that is running AIX, and all the machines of the database system are in the switch domain, you can specify the switch name as the *hostname* value. If you do this, the switch name is recognized as the host name. If a switch is not active, the **db2start** command will issue a communications error. An example is as follows:

```
nodenum  hostname

1        SYSSW1
2        SYSSW2
```

If you have machines that are not in the switch domain, and you want to issue the **db2start** and **db2stop** commands from them, do *not* use this method.

You can update the *db2nodes.cfg* file using an editor of your choice. You must be careful, however, to protect the integrity of the information in the file, as data partitioning requires that the node number not be changed. The node configuration file is locked when you issue the **db2start** command and unlocked after the **db2stop** command stops the database manager. When the file is locked, the **db2start** command updates the file, if necessary (for example if you issue the **db2start nodenum restart** or **db2start addnode** commands).

If the **db2stop** command is not successful and does not unlock the node configuration file, issue the **db2stop force** command to unlock it.

Step 6. Make DB2 Extended Enterprise Edition Available

To make DB2 Extended Enterprise Edition accessible to users, you must ensure that the instance owner ID can execute any command using the **rsh** command on all nodes defined in the *db2nodes.cfg* file. To do this, you can either create an *INSTHOME/.rhosts* file or an */etc/hosts.equiv* file.

If you want to use the **autoloader** command, you must also include the *netname* values for each machine. These values are specified in the fourth column of the *db2nodes.cfg* file.

For example, the *INSTHOME/.rhosts* file may look like:

```
machine1.torolab.ibm.com db2inst1
machine2.torolab.ibm.com db2inst1
machine3.torolab.ibm.com db2inst1
switch01.torolab.ibm.com db2inst1
switch02.torolab.ibm.com db2inst1
switch03.torolab.ibm.com db2inst1
```

For more information about the *INSTHOME/.rhosts* or */etc/hosts.equiv* files, refer to your operating system's reference manual.

You should also ensure that the IP address of every machine can be resolved for the names in the */etc/hosts* and the */etc/resolv.conf* files. These files are also described in your operating system's reference manual.

The system database directory contains information about all defined databases. You must ensure that all nodes always access the same *system database directory* file, *INSTHOME/sqllib/sqldbdir/sqldbdir*. Unpredictable errors can occur if either the system database directory or the *system intention file* (*INSTHOME/sqllib/sqldbdir/sqldbins*) is a symbolic link to another file. All database partition servers need to access the same copy of these files.

Step 7. Create the Administration Server

The Administration Server provides services to support client tools that:

- Automate the configuration of connections to DB2 databases.
- Administer DB2 from your server system or a remote client using the Control Center.

Use the **dasicrt** command to create an Administration Server. You must run this command on the machine where you have DB2 physically installed and have root authority.

The syntax of the **dasicrt** command is as follows:

```
DB2DIR/instance/dasicrt ASName
```

where DB2DIR	= /usr/lpp/db2_05_00	on AIX
	= /opt/IBMdb2/V5.0	on Solaris

and where *ASName* is the name of the Administration Server, which is composed of a string of up to eight alphanumeric characters long. See Appendix F, “Naming Rules” on page 315 for more information.

Note: If you are running NIS or NIS+, you need to set up the user and group names in such a way that:

- The primary group of the Administration Server must be in the secondary group of all the instances.
- The secondary group of the Administration Server must contain the primary group of all the instances

Secondary group lists will be modified automatically only if NIS or NIS+ is running on your system.

Because a user ID can only own one instance, you must have a separate user ID to own each Administration Server that you create.

If you are running file collections on an RS/6000 SP system that is running AIX, ensure that the instance user ID is created from the file collections server. When file collections is run, the user ID and group will be deleted if added to an RS/6000 SP node that is *not* the file collections server.

To see if file collections is running, issue the **crontab -l** command from one of the RS/6000 SP nodes and look for an entry such as:

```
10 * * * * /var/sysman/supper update sup.admin
user.admin node.root
```

It is possible to add an instance to any machine. If you use different machines to create instances, then the list of instances that is stored on any machine may not be a complete list. We recommend creating every instance on one machine, so that the list of instances is complete on that machine. We also recommend that this node be the Administration Server node.

You use the name of the Administration Server to set up the directory structure and access permissions.

Use the **db2admin start** command to start the Administration Server (refer to “Starting the Administration Server” on page 186 for more information). When you reboot the system, it is automatically started.

By adding a system that has an Administration Server to the Control Center, you are designating the corresponding database partition server (that resides on the same machine as the Administration Server) as the Control Center's connection point to the entire instance. The system specified becomes the coordinator node for requests that are issued from the Control Center.

You may want to consider distributing the coordinator function among other database partition servers in the instance. To do this:

1 Create a new Administration Server on the other machines that are in the partitioned database system.

2 Catalog each Administration Server as a separate system in the Control Center.

3 Catalog the same instance under each new system, and each time specify the same machine that you used to catalog the Administration Server.

Step 8. Set up Discovery

If you are not planning on using the CCA or the Control Center, you can skip this step and proceed to the next step.

You must configure Discovery, a feature that is used by the CCA and Control Center, to retrieve information about systems on your network. Specifically, you must update instance lists and the Administration Server configuration to ensure that Discovery retrieves the correct information.

Update Instance Lists

If you created more than one Administration Server in your partitioned database system and then created instances on one of the machines that has an Administration Server on it, you should verify that the other Administration Servers are aware of all the instances. They may not be aware of all of them because the instance lists on some machines will be incomplete until you run the **db2iset** command or the **db2start** command on the instance.

To ensure that the Administration Servers are aware of all the instances, perform the following steps:

1 For each Administration Server, run the **db2set -l** command to display the list of instances known to it.

If this list of instances includes all the instances that you want the Administration Server to recognize, skip the following steps and go to "Update the Administration Server Configuration" below. If this list does not include all the instances, go to the next step.

2 For each instance that was not included in the list generated by the **db2set -l** command, check that the *db2nodes.cfg* file contains a line for the machine on which the Administration Server resides. If the machine on which the Administration Server resides is not in the *db2nodes.cfg* file, it must be added. For more information on how to add this entry, see to“Step 5. Create the Node Configuration File” on page 64 (for AIX) or “Step 8. Create the Node Configuration File” on page 84 (for Solaris).

3 Update the instance lists manually by performing the following steps:

a Log in as root.

b Run the **db2iset -a** *instance_name* command to add the instance.

If you don't update the instance lists manually, they will be updated automatically by DB2 when you run the **db2start** command on an instance. DB2 uses the node configuration file (*db2nodes.cfg*) to determine the machines on which the instance is defined and updates the instance lists on those machines. Therefore, if you didn't create an instance on the same machine as the Administration Server, the server will not recognize that the instance exists until you run the **db2start** command for that instance. To ensure that the Administration Server can return information about all instances in the partitioned database system, you should use the **db2iset** command to update the list of instances manually.

Update the Administration Server Configuration
By default, the install program sets the *DB2SYSTEM* registry value to the TCP/IP *hostname*. The system names that are retrieved by Discovery are the systems on which an Administration Server resides. Discovery uses the database partition server that is on the same machine as the Administration Server as the coordinator node when connections are established.

There are two ways of updating an Administration Server's configuration:

- If you want to be able to select a coordinator node from a list of DB2 systems, set DISCOVER=*SEARCH* in each Administration Server's configuration file.

  When there are multiple Administration Servers, the same instance may appear in more than one system on the CCA or Control Center's interface; however, each system will have a different communications access path to the instances. Users can select different DB2 systems as coordinator nodes for communications and thereby redistribute the workload.

- If you do not want users to be able to select the coordinator node, set DISCOVER=*KNOWN* on all Administration Servers, except set DISCOVER=*SEARCH* on just one Administration Server in the Administration Server configuration. Discovery uses the database partition server where the Administration Server resides as a coordinator node when connections are established.

Step 9. Create Links for DB2 Files (Optional)

This step is optional and can be used to create links for the DB2 files to the */usr/lib* directory, and for the include files to the */usr/include* directory for a particular version and release level of the product.

You may want to create these links if you are developing or running applications and want to avoid having to specify the full path to the product libraries and include files.

Invoke the **db2ln** command as follows: *dsh /usr/lpp/db2_05_00/cfg/db2ln*

If you do not issue the **db2ln** command using the **dsh** command, **db2ln** only sets up the links on the machine where the command is issued.

If there are existing links to the */usr/lib* and */usr/include* directories from previous versions of DB2, they will automatically be removed by executing the **db2ln** command to create links for this version of DB2. If you want to re-establish the links to the libraries or previous versions, then you must execute the **db2rmln** command from Version 5 before you execute the **db2ln** command from the previous versions of DB2.

Links can be established for only one version of DB2 on a given system.

Step 10. Set up the Instance Environment

By default, the scripts affect the user environment for the duration of the current session only. You can change the *.profile* file to enable it to run the *db2profile* script automatically when the user logs on using the Bourne or Korn shell. For users of the C shell, you can change the *.login* file to enable it to run the *db2cshrc* script file.

Add one of the following statements to the *.profile* or *.login* script files.

- For users who share one version of the script, add:

  ```
     . INSTHOME/sqllib/db2profile    (for Bourne or Korn shell)
     source INSTHOME/sqllib/db2cshrc (for C shell)
  ```

  where INSTHOME is the home directory of the instance that you want to use.

- For users that have a customized version of the script in their home directory, add:

  ```
  . USERHOME/db2profile              (in Bourne or Korn Shell)
  source USERHOME/db2cshrc           (in C Shell)
  ```

  where USERHOME is the home directory of the user.

For more information, see "Setting the Operating Environment for Each Instance User" on page 178.

Now that you have installed DB2 Extended Enterprise Edition, see Chapter 23, "Setting up Communications on the Server Using the Command Line Processor" on page 231 for information on how to set up your server to accept in-bound client connections.

Chapter 6. Installing and Configuring DB2 Universal Database Extended Enterprise Edition for Solaris

This section describes how to install DB2 Universal Database Extended Enterprise Edition for Solaris.

To install DB2 Universal Database Extended Enterprise Edition for AIX, go to page Chapter 5.

To install a DB2 client, go to Chapter 8, “Installing DB2 Clients” on page 111.

Installation Steps

To install DB2 Extended Enterprise Edition for Solaris, perform the following steps:

1 Install DB2. As part of this installation step, the following installation activities are optional:

  a Install the DB2 Product Library (documents).

  b Install the DB2 Product Messages.

2 Update the kernel configuration parameters.

3 Create or assign groups and user IDs.

4 Enable the Fast Communication Manager (FCM).

5 Share the home directory of the instance and the Administration Server.

6 Create a DB2 instance.

7 Create an Administration Server.

8 Create the node configuration file.

9 Make DB2 Extended Enterprise Edition available.

10 Set up Discovery.

11 Create links for DB2 files (optional).

12 Set up the DB2 environment.

Step 1. Install DB2

Use the **db2_install** command, which is available on the DB2 CD-ROM, to install DB2. If your system does not have a local CD-ROM drive, you can mount the CD-ROM on a different system and NFS mount on the current system.

To install DB2 from the DB2 CD-ROM, perform the following steps:

1 Log in as root.

2 Insert the DB2 CD-ROM into the CD-ROM drive.

3 If the Volume Manager is not running on your system, enter the following commands to mount the CD-ROM:

```
mkdir -p /cdrom/unnamed_cdrom
mount -F hsfs -o ro /dev/dsk/c0t6d0s2 /cdrom/unnamed_cdrom
```

where */cdrom/unnamed_cdrom* is the CD-ROM mount directory.

> If you are mounting the CD-ROM drive from a remote system using NFS, the CD-ROM file system on the remote machine must be exported with *root* access. You must also mount that file system with *root* access on the local machine.

If the Volume Manager (vold) is running on your system, the CD-ROM is automatically mounted as:

```
/cdrom/unnamed_cdrom
```

4 Run the **db2_install** command as follows:

```
/cdrom/unnamed_cdrom/db2_install
```

The **db2_install** command prompts for one or more of the following to be installed, and for the base directory where the product files are to be installed. The products are listed by keyword and product description.

DB2.EENT	DB2 Universal Database Extended Enterprise Edition for Solaris
DB2.CAE	DB2 Universal Database Client Application Enabler

The command displays the following prompt: `Specify one or more of the keywords separated by spaces.`

5 Type the keyword of the products to be installed when prompted.

6 Type the name of the base directory when prompted. The default base directory is */opt*.

If the default base directory is used, all files will be installed in the */opt/IBMdb2/V5.0* directory.

7 Enter `Yes` to start the DB2 product installation.

Installing the DB2 Product Library (Optional)

To view DB2 product documents online, HTML versions of these documents are provided. These documents are translated into several languages; however, not every manual in the DB2 product library is translated into multiple languages. The English version of every manual is available in HTML format.

A separate package exists for every language in which DB2 documents are translated. Packages for DB2 documentation are given names such as `db2ht%L50`, where `%L` represents two letters for the language and locale identifier.

For example, the package name for the DB2 Product Library in English is `db2hten50`. For a complete list of packages for the DB2 Product Library, see Appendix D, "Contents of the DB2 Products" on page 309.

To install the DB2 Product Library (HTML) package, do the following:

1 Log in as root and mount the DB2 product CD-ROM as described in "Step 1. Install DB2" on page 74.

2 Run the **pkgadd** command as follows:

```
pkgadd -d cdrom-dir/db2 pkgname
```

where *cdrom-dir* is the CD-ROM mount directory and *pkgname* is the package name. For example, to install DB2 documentation in English, you can enter the following command:

```
pkgadd -d /cdrom/unnamed_cdrom/db2 db2hten50
```

This command installs the compressed-tar files for the DB2 documents in the */opt/IBMdb2/V5.0*/doc/*lang*/html directory, where *lang* is the language and locale identifier.

3 Run the following command to uncompress and untar the HTML files:

```
/opt/IBMdb2/V5.0/doc/db2insthtml lang
```

where *lang* is the language and locale identifier.

For example, *lang*=en_US for English.

4 From your Web browser, open the following URL to view the DB2 manuals:

```
file:/opt/IBMdb2/V5.0/doc/en_US/html/index.htm
```

To recover some disk space, you may want to remove the compressed tar files. It is recommended that you do not just remove the compressed tar HTML files; instead, you should remove the fileset for the DB2 Product Library.

After the fileset is removed, only compressed tar files will be removed. You will still be able to view the online DB2 books.

Installing the DB2 Product Messages (Optional)

A separate package exists for every locale in which DB2 Product Messages are translated. Packages for DB2 Product Messages are given names such as `db2ms%L50`, where `%L` represents two letters for the language name. For example, the package name for the DB2 Product Messages in the `de` (German) locale is `db2msde50`. For a complete list of DB2 Product Messages, see Appendix D, “Contents of the DB2 Products” on page 309.

To install DB2 Product Message packages, do the following:

1 Log in as root and mount the DB2 CD-ROM as described in “Step 1. Install DB2” on page 74.

2 Run the **pkgadd** command as follows:

```
pkgadd -d cdrom-dir/db2 pkgname
```

where *cdrom-dir* is the CD-ROM mount directory and *pkgname* is the package name.

For example, to install the DB2 Product Messages in the `de` (German) locale, enter the following command:

```
pkgadd -d /cdrom/unnamed_cdrom/db2 db2msde50
```

Step 2. Update the Kernel Configuration Parameters

To run DB2 Extended Enterprise Edition for Solaris, you may have to update some kernel configuration parameters; the following minimum values are recommended:

Table 7. Solaris Kernel Configuration Parameters (Recommended Values)

Kernel Parameter	**Virtual Memory**			
	64MB - 128MB	**128MB - 256MB**	**256MB - 512MB**	**512MB+**
msgsys:msginfo_msgmax	65535 (1)	65535 (1)	65535 (1)	65535 (1)
msgsys:msginfo_msgmnb	65535 (1)	65535 (1)	65535 (1)	65535 (1)
msgsys:msginfo_msgmap	130	258	258	258
msgsys:msginfo_msgmni	128	256	256	256
msgsys:msginfo_msgssz	16	16	16	16
msgsys:msginfo_msgtql	128	256	256	512
msgsys:msginfo_msgseg	8192	16384	32768	32768
shmsys:shminfo_shmmax	67108864	134217728 (2)	268435456 (2)	268435456 (2)
shmsys:shminfo_shmseg	16	16	16	16
shmsys:shminfo_shmmni	300	300	300	300
semsys:seminfo_semmni	128	256	512	1024
semsys:seminfo_semmap	130	258	514	1026
semsys:seminfo_semmns	256	512	1024	2048
semsys:seminfo_semmnu	256	512	1024	2048

Notes:

1. The *msgsys:msginfo_msgmnb* and *msgsys:msginfo_msgmax* parameters must be set to 65535.
2. The *shmsys:shminfo_shmmax* parameters should be set to the suggested value in the above table, or 90% of the virtual memory (in bytes), whichever is higher. For example, if you have 196 MB of virtual memory in your system, set the *shmsys:shminfo_shmmax* parameter to 184968806 (176*1024*1024).

To set a kernel parameter, add a line at the end of the */etc/system* file as follows:

```
set parameter-name = value
```

For example, to set the value of the *msgsys:msginfo_msgmax* parameter, add the following line to the end of the */etc/system* file:

```
set msgsys:msginfo_msgmax = 65535
```

Sample files for updating the kernel configuration parameters are provided in the */opt/IBMdb2/V5.0/cfg* directory. The names for these files are as follows:

- *kernel.param.64MB* for systems with 64MB–124MB of virtual memory
- *kernel.param.128MB* for systems with 128MB–256MB of virtual memory
- *kernel.param.256MB* for systems with more than 256MB–512MB of virtual memory
- *kernel.param.512MB* for systems with more than 512MB of virtual memory

Depending upon the amount of virtual memory in your system, append the appropriate kernel configuration parameter file to the */etc/system* file. If necessary, change the value of the *shmsys:shminfo_shmmax* parameter as described in Note 2 above, then reboot the system.

Step 3. Create or Assign Groups and User IDs

If you want to use an existing user or group ID, you do not need to create new ones at this time. Instead, you can proceed to the next step.

If you are using Network Information Services (NIS), NIS+, or any other tools to manage system configuration files, you must perform this step on the master server. After you have completed the following, ensure that the changes have taken effect on all DB2 servers.

You must perform the following as a user with **root** authority.

Create a group that will be the primary group for the user ID that will be the instance owner. When the instance is created, the instance owner and any other user who is a member of this group will have system administrative (SYSADM) authority. The group must be created on every machine that is to be used as a database partition server. The group ID (GID) of the group must be the same on each machine.

Next, create a user ID that will be the instance owner. This user ID will be the name of the instance. The user ID must be created on every machine that is to be used in the partitioned database system. The user ID of the user must be the same on every machine. Make this user's primary group the group that you just created. The instance owner user ID, and any other user who is a member of this group, will be automatically given SYSADM control for the instance.

In the following example, the instance user ID is `db2inst1` and the UID `1004`. The SYSADM group ID is `dbadmin1` and the GID `999`.

Dedicate the instance owner user ID to that instance's use only. This allows for easier error recovery if a system error occurs.

1 Use **admintool** or **groupadd** to create groups, and **admintool** or **useradd** to create users. For example:

```
groupadd -g 999 dbadmin1
useradd -g dbadmin1 -u 1004 -d /export/home/db2inst1 -m db2inst1
passwd db2inst1
```

2 Similarly, create a group and user for fenced user defined functions and stored procedures.

For example:

```
groupadd -g 991 db2fadm1
useradd -g db2fadm1 -u 1001 -d /export/home/db2fenc1 -m db2fenc1
passwd db2fenc1
```

Fenced user defined functions (UDFs) and stored procedures will execute under this user and group. The group must be the primary group of the user. The user for fenced UDFs and stored procedures is specified as a parameter of the instance creation script. The group for fenced UDFs and stored procedures is implicitly set to the primary group of this specified user (db2fenc1).

For security reasons, we recommend that you do not use the instance name as the FencedID. However, if you are not planning to use fenced user defined functions or stored procedures, you can set the FencedID to the instance name instead of creating another user for the FencedID.

Notes:

1. You have to repeat Steps 1 and 2 above to create a user name and group name for the Administration Server. You must use different user IDs for the Administration Server and a DB2 instance (for example, db2as and db2inst). For security reasons, you should not use the primary group of a DB2 instance as the primary group for the user ID for the Administration Server. It is recommended that you create a different group ID (for example, db2asgrp).
2. In addition to the rules imposed by the operating system for login names and groups, you must also adhere to the rules described in Appendix F, “Naming Rules” on page 315.

Step 4. Share the Instance Home Directory

You must share the home directory of the instance across all database partition servers.

Using NFS, you can mount the home directory of the instance from one machine to all other machines that will participate in the partitioned database system. On the NFS server, the home directory of the instance must be exported with *root* access. On the NFS clients, this directory *must not* be mounted with the *nosuid* option.

Instead of exporting just the home directory of the instance (*/export/home/INSTNAME*, where *INSTNAME* is the name of the instance), it is recommended that you export the home filesystem */export/home*. This eliminates the need to export the home directory of each instance individually.

If you are mounting the */export/home/INSTNAME* directory from the NFS server, make sure that the home directory of the instance on the local machine is empty *before* you mount the directory.

Step 5. Enable the Fast Communication Manager

If you are using Network Information Services (NIS), NIS+, or any other tools to manage system configuration files, you must perform this step on the master server. After you have completed the following, ensure that the changes have taken effect on all DB2 servers.

To enable FCM communications between database partition servers, you must create a service entry for each logical node in the */etc/services* file. The service name maps the database partition server at the machine to the actual TCP/IP port that is used. The syntax of a service entry is as follows:

```
DB2_instance port/tcp #comment
```

where:

DB2_*instance* Is the name of the DB2 Extended Enterprise Edition instance. All characters in the name must be lowercase. Assuming an instance name of `db2inst1`, you would specify `DB2_db2inst1`.

port/tcp Is the TCP/IP port that you want to reserve for the database partition server (node).

comment Any comment that you want to associate with the entry. The comment must be preceded by a pound sign (#).

You must ensure that the entry defined above is the same in all */etc/services* files on all machines.

As an example, suppose you wanted to make an entry in */etc/services* for the instance `db2inst1`, and have it assigned to the TCP/IP port *9000* (indicating that there is only one node on every machine in the instance). The service name is:

```
DB2_db2inst1     9000/tcp
```

If you want a single machine to have multiple logical nodes for an instance, you will need to allocate multiple ports for FCM to use; one port is needed for each *logical* node. Enter two lines in the */etc/services* file to indicate the number of ports that you are allocating. The first line specifies the first port, while the second line indicates the end of the block of ports. In the following example, five ports are allocated for the instance. This means that no machine in the instance db2inst1 is running more than five logical nodes.

```
DB2_db2inst1        9000/tcp
DB2_db2inst1_END    9004/tcp
```

You must ensure that the number of ports allocated in the file is either greater than or equal to the largest number of multiple logical nodes on any machine in the instance. When allocating ports, ensure that you account for any machines that can be used as a backup (for example, if you use High Availability (HA)).

Note: Specify the END keyword in uppercase only. Also ensure that you include both underscore (_) characters.

Step 6. Create a DB2 Instance

If you are planning on setting up High Availability (HA) for DB2 using Solstice HA 1.3, skip the following step and see Appendix C, “Implementing Failover Support in the Solaris Operating Environment” on page 299 for instructions.

A DB2 instance is an environment where you store data and run applications. Use the **db2icrt** command to create an instance. You must run this command on the machine where you have DB2 physically installed and have *root* authority. For more information about database instances, see Chapter 17, “Working with Instances” on page 173.

It is possible to add an instance to any machine. If you use different machines to create instances, then the list of instances that is stored on any machine may not be a complete list. We recommend creating every instance on one machine, so that the list of instances is complete on that machine. We also recommend that this node be the Administration Server node.

The **db2icrt** command is located in the *DB2DIR/instance* directory, where DB2DIR is /opt/IBMdb2/V5.0.

Run the **db2icrt** command from the machine where the home directory of the instance (*INSTHOME*) physically resides. Because the *INSTHOME* directory is NFS-mounted on all machines, you only have to run the **db2icrt** command on one machine.

The syntax of the **db2icrt** command is:

```
►►──db2icrt──[-h| -?]──[-d]───[-a AuthType]─────[-u FencedID] ──────────►

►────── InstName ──────────────────────────────────────────────────────►◄
```

where:

-h | -? Display a help menu for this command.

-d Sets the debug mode that you can use for problem determination.

-a *AuthType* Is an optional parameter that specifies the authentication type for the instance. Valid authentication types are SERVER, CLIENT, and DCS. If the *-a* parameter is not specified, the authentication type defaults to SERVER, if a DB2 server is installed. Otherwise, the *AuthType* is set to CLIENT.

Notes:

1. All databases in the instance have the same authentication type.
2. DCE authentication is not valid for this command; however, you can enable DCE authentication for an instance. For more information, refer to the *Administration Guide*.

-u *FencedID* Is the user under which the fenced UDFs and stored procedures will execute. This is not required if you install the DB2 Client Application Enabler or the DB2 Software Developer's Kit. For other products, this is a required parameter.

Note: *FencedID* may not be *root* or *bin*.

InstName Is the login name of the instance owner.

When you create an instance, ensure that its primary and secondary groups are different from the Administration Server's primary (SYSADM) group. When you create an instance on the same machine as the Administration Server, its SYSADM group is automatically added to the secondary group list of the Administration Server so that you can use the Control Center to perform administration tasks on that instance.

Examples:

- To create an instance for a DB2 server, you can use the following command:

  ```
  db2icrt -u db2fenc1 db2inst1
  ```

- To create an instance for a DB2 client, you can use the following command:

  ```
  db2icrt db2inst1
  ```

When an instance is created, its name is also added to the list of instances on the system.

The **db2icrt** command creates the *INSTHOME/sqllib* directory, where *INSTHOME* is the home directory of the instance owner.

> To avoid a potential loss of data if an instance is deleted, you should not create user files or directories under the *INSTHOME/sqllib* directory, other than those created by DB2. Exceptions are:
>
> 1. If your system supports fenced user defined functions and fenced stored procedures, put the fenced applications in the *INSTHOME/sqllib/function* directory.
> 2. The location of the *db2nodes.cfg* file. This file should be put in the *INSTHOME/sqllib* directory, where *INSTHOME* is the home directory of the instance owner.

Step 7. Create the Administration Server

The Administration Server provides services to support client tools that:

- Automate the configuration of connections to DB2 databases.
- Administer DB2 from your server system or a remote client using the Control Center.

Use the **dasicrt** command to create an Administration Server. You must run this command on the machine where you have DB2 physically installed and have root authority.

The syntax of the **dasicrt** command is as follows:

```
DB2DIR/instance/dasicrt ASName
```

where DB2DIR	= /usr/lpp/db2_05_00	on AIX
	= /opt/IBMdb2/V5.0	on Solaris

and where *ASName* is the name of the Administration Server, which is composed of a string of up to eight alphanumeric characters long. See Appendix F, “Naming Rules” on page 315 for more information.

Note: If you are running NIS or NIS+, you need to set up the user and group names in such a way that:

- The primary group of the Administration Server must be in the secondary group of all the instances.
- The secondary group of the Administration Server must contain the primary group of all the instances

Secondary group lists will be modified automatically only if NIS or NIS+ is running on your system.

Because a user ID can only own one instance, you must have a separate user ID to own each Administration Server that you create.

It is possible to add an instance to any machine. If you use different machines to create instances, then the list of instances that is stored on any machine may not be a complete list. We recommend creating every instance on one machine, so that the list of instances is complete on that machine. We also recommend that this node be the Administration Server node.

You use the name of the Administration Server to set up the directory structure and access permissions.

Use the **db2admin start** command to start the Administration Server (refer to “Starting the Administration Server” on page 186 for more information). When you reboot the system, it is automatically started.

By adding a system that has an Administration Server to the Control Center, you are designating the corresponding database partition server (that resides on the same machine as the Administration Server) as the Control Center's connection point to the entire instance. The system specified becomes the coordinator node for requests that are issued from the Control Center.

You may want to consider distributing the coordinator function among other database partition servers in the instance. To do this:

1 Create a new Administration Server on the other machines that are in the partitioned database system.

2 Catalog each Administration Server as a separate system in the Control Center.

3 Catalog the same instance under each new system, and each time specify the same machine that you used to catalog the Administration Server.

Step 8. Create the Node Configuration File

If you are setting up High Availability (HA) for DB2, the **hadb2eee.setup** program completed this step for you. See Appendix C, “Implementing Failover Support in the Solaris Operating Environment” on page 299 for more information.

You must create the node configuration file, *INSTHOME/sqllib/db2nodes.cfg*, before you can start the partitioned database system. This file contains configuration information for all database partition servers in an instance, and is shared by all database partition servers for that instance.

The file contains one line for each database partition server that belongs to an instance. Each line has the following format:

```
nodenum hostname [logical_port [netname]]
```

Tokens are delimited by blanks. The variables are:

nodenum
The unique node number to identify the database partition server. The number can be from 0 to 999. Node numbers must be in ascending sequence. You can have gaps in this sequence.

When a node number is assigned, it cannot be changed. Otherwise, the information in the partitioning map (which details how data is partitioned) would be compromised.

If you drop a database partition server, its node number can be used again on a new database partition server that you add.

The node number is used to generate a node name in the database directory during the **create database** command. It has the format:

```
NODEnnnn
```

where *nnnn* is the node number, which is left-padded with zeros.

hostname
The TCP/IP hostname of the IP address which is used by the FCM for internal communications, unless *netname* is specified. If *netname* is specified, the FCM uses *netname*, and *hostname* is only used for the **db2start**, **db2stop**, and **db2_all** commands.

logical_port
This parameter is optional, and specifies the logical port number for the database partition server. This number is used with the database manager instance name to identify a TCP/IP service name entry in the */etc/services* file.

The combination of the IP address and port number is used as a well-known address, and must be unique among all applications to support communication connections between database partition servers.

For each *hostname*, one *logical_port* must be either 0 or blank (this defaults to 0). The database partition server associated with the *logical-port 0* is the default database partition server on the machine to which clients connect. This default database partition server is known as the *coordinator node*. You can override this with the *DB2NODE* environment variable, or with the **set client** command.

If you have multiple logical nodes, (that is, more than one *nodenum* for a machine), you must assign *logical_port* numbers to the logical nodes.

netname
This parameter is optional, and is used to support a machine that has more than one active TCP/IP interface, each with its own hostname.

This parameter only applies to those machines that have a primary *hostname* that is either an Ethernet or a token-ring name, and you want to have DB2 Extended

Enterprise Edition use an alternative name (for example, a switch name). If you do not specify a value for this parameter, its value is that of the *hostname* parameter.

Notes:

1. If you specify a *netname*, you must specify a value for *logical_port*.
2. The *netname* is commonly referred to as a switch name.
3. If you want to use the **autoloader** command, the *netname* must be specified.

The following is an example of a possible node configuration file on which SYSEN1 has multiple TCP/IP interfaces, two logical nodes, and uses SYSSW1 as the DB2 Extended Enterprise Edition interface. It also shows the node numbers starting at 1, and a gap in the *nodenum* sequence:

```
nodenum   hostname   logical-port   netname

1         SYSEN1     0              SYSSW1
2         SYSEN1     1              SYSSW1
4         SYSEN2     0
5         SYSEN3
```

The logical port for node 5 is assumed to be 0.

You can update the *db2nodes.cfg* file using an editor of your choice. You must be careful, however, to protect the integrity of the information in the file, as data partitioning requires that the node number not be changed. The node configuration file is locked when you issue the **db2start** command and unlocked after the **db2stop** command stops the database manager. When the file is locked, the **db2start** command updates the file, if necessary (for example if you issue the **db2start nodenum restart** or **db2start addnode** commands).

If the **db2stop** command is not successful and does not unlock the node configuration file, issue the **db2stop force** command to unlock it.

Step 9. Make DB2 Extended Enterprise Edition Available

If you are setting up High Availability (HA) for DB2, the **hadb2eee.setup** program completed this step for you. See Appendix C, “Implementing Failover Support in the Solaris Operating Environment” on page 299 for more information.

To make DB2 Extended Enterprise Edition accessible to users, you must ensure that the instance owner ID can execute any command using the **rsh** command on all nodes defined in the *db2nodes.cfg* file. To do this, you can either create an *INSTHOME/.rhosts* file or an */etc/hosts.equiv* file.

If you want to use the **autoloader** command, you must also include the *netname* values for each machine. These values are specified in the fourth column of the *db2nodes.cfg* file.

For example, the *INSTHOME/.rhosts* file may look like:

```
machine1.torolab.ibm.com db2inst1
machine2.torolab.ibm.com db2inst1
machine3.torolab.ibm.com db2inst1
switch01.torolab.ibm.com db2inst1
switch02.torolab.ibm.com db2inst1
switch03.torolab.ibm.com db2inst1
```

For more information about the *INSTHOME/.rhosts* or */etc/hosts.equiv* files, refer to your operating system's reference manual.

You should also ensure that the IP address of every machine can be resolved for the names in the */etc/hosts* and the */etc/resolv.conf* files. These files are also described in your operating system's reference manual.

The system database directory contains information about all defined databases. You must ensure that all nodes always access the same *system database directory* file, *INSTHOME/sqllib/sqldbdir/sqldbdir*. Unpredictable errors can occur if either the system database directory or the *system intention file* (*INSTHOME/sqllib/sqldbdir/sqldbins*) is a symbolic link to another file. All database partition servers need to access the same copy of these files.

Step 10. Set up Discovery

If you are not planning on using the CCA or the Control Center, you can skip this step and proceed to the next step.

You must configure Discovery, a feature that is used by the CCA and Control Center, to retrieve information about systems on your network. Specifically, you must update instance lists and the Administration Server configuration to ensure that Discovery retrieves the correct information.

Update Instance Lists

If you created more than one Administration Server in your partitioned database system and then created instances on one of the machines that has an Administration Server on it, you should verify that the other Administration Servers are aware of all the instances. They may not be aware of all of them because the instance lists on some machines will be incomplete until you run the **db2iset** command or the **db2start** command on the instance.

To ensure that the Administration Servers are aware of all the instances, perform the following steps:

1. For each Administration Server, run the **db2set -l** command to display the list of instances known to it.

   If this list of instances includes all the instances that you want the Administration Server to recognize, skip the following steps and go to "Update the Administration Server Configuration" below. If this list does not include all the instances, go to the next step.

2. For each instance that was not included in the list generated by the **db2set -l** command, check that the *db2nodes.cfg* file contains a line for the machine on which the Administration Server resides. If the machine on which the Administration Server resides is not in the *db2nodes.cfg* file, it must be added. For more information on how to add this entry, see to“Step 5. Create the Node Configuration File” on page 64 (for AIX) or “Step 8. Create the Node Configuration File” on page 84 (for Solaris).

3. Update the instance lists manually by performing the following steps:

   a. Log in as root.

   b. Run the **db2iset -a** *instance_name* command to add the instance.

   If you don't update the instance lists manually, they will be updated automatically by DB2 when you run the **db2start** command on an instance. DB2 uses the node configuration file (*db2nodes.cfg*) to determine the machines on which the instance is defined and updates the instance lists on those machines. Therefore, if you didn't create an instance on the same machine as the Administration Server, the server will not recognize that the instance exists until you run the **db2start** command for that instance. To ensure that the Administration Server can return information about all instances in the partitioned database system, you should use the **db2iset** command to update the list of instances manually.

Update the Administration Server Configuration

By default, the install program sets the *DB2SYSTEM* registry value to the TCP/IP *hostname*. The system names that are retrieved by Discovery are the systems on which an Administration Server resides. Discovery uses the database partition server that is on the same machine as the Administration Server as the coordinator node when connections are established.

There are two ways of updating an Administration Server's configuration:

- If you want to be able to select a coordinator node from a list of DB2 systems, set DISCOVER=*SEARCH* in each Administration Server's configuration file.

  When there are multiple Administration Servers, the same instance may appear in more than one system on the CCA or Control Center's interface; however, each system will have a different communications access path to the instances. Users can select different DB2 systems as coordinator nodes for communications and thereby redistribute the workload.

- If you do not want users to be able to select the coordinator node, set DISCOVER=*KNOWN* on all Administration Servers, except set DISCOVER=*SEARCH* on just one Administration Server in the Administration Server configuration. Discovery uses the database partition server where the Administration Server resides as a coordinator node when connections are established.

Step 11. Create Links for DB2 Files (Optional)

This step is optional and can be used to create links for the DB2 files to the */usr/lib* directory, and for the include files to the */usr/include* directory for a particular version and release level of the product.

You may want to create these links if you are developing or running applications and want to avoid having to specify the full path to the product libraries and include files.

You invoke the **db2ln** command as follows:

```
/opt/IBMdb2/V5.0/cfg/db2ln
```

If there are existing links to the */usr/lib* and */usr/include* directories from previous versions of DB2, they will automatically be removed by executing the **db2ln** command to create links for this version of DB2. If you want to re-establish the links to the libraries or previous versions, then you must execute the **db2rmln** command from Version 5 before you execute the **db2ln** command from the previous versions of DB2.

> Links can be established for only one version of DB2 on a given system.

Step 12. Set up the Instance Environment

By default, the scripts affect the user environment for the duration of the current session only. You can change the *.profile* file to enable it to run the *db2profile* script automatically when the user logs on using the Bourne or Korn shell. For users of the C shell, you can change the *.login* file to enable it to run the *db2cshrc* script file.

Add one of the following statements to the *.profile* or *.login* script files.

- For users who share one version of the script, add:

  ```
     . INSTHOME/sqllib/db2profile    (for Bourne or Korn shell)
     source INSTHOME/sqllib/db2cshrc (for C shell)
  ```

  where INSTHOME is the home directory of the instance that you want to use.

- For users that have a customized version of the script in their home directory, add:

  ```
  . USERHOME/db2profile                (in Bourne or Korn Shell)
  source USERHOME/db2cshrc             (in C Shell)
  ```

  where USERHOME is the home directory of the user.

For more information, see “Setting the Operating Environment for Each Instance User” on page 178.

Now that you have installed DB2 Extended Enterprise Edition, see Chapter 23, “Setting up Communications on the Server Using the Command Line Processor” on page 231 for information on how to set up your server to accept in-bound client connections.

Chapter 7. Migrating from Previous Versions

This section describes the steps required to migrate to DB2 Extended Enterprise Edition Version 5 database system from:

- Version 1.x or Version 2.x single-partition UNIX database system
- DB2 Parallel Edition for AIX Version 1.x database system
- Database Server Version 4.x for UNIX-based database systems. (In this section, any references to Version 2.x single-partition database systems also apply to Database Server Version 4 database systems.

This section contains the following information:

- “Overview of the Migration Process”
- “Pre-Migration” on page 94.
- “Migrating Instances” on page 94.
- “Migrating Databases” on page 101.
- “Post Migration” on page 104.

To avoid difficulties during database migration, read this entire section before starting the migration process. It is strongly recommended that you follow all the steps listed in this section.

Overview of the Migration Process

After you install DB2 Extended Enterprise Edition, any databases that you created with earlier versions of DB2 will not automatically work with the new version of DB2. You must follow the steps in this section to migrate your instances and then to move data from the previous database system to the new one. Although your data is not changed during the process, the database is different from the original one. The differences depend on the type of database system from which you are migrating.

Migrating Single-Partition Databases

After you migrate a single-partition database from an earlier version of DB2, the following changes are made to your database:

- The database is relocated from the *INSTHOME/INSTNAME* directory to the *INSTHOME/INSTNAME/NODE0000* directory. A node is created with a default name of NODE0000.
- The database is migrated to SMS table space containers. If you are migrating a Version 1.x database, a table space container is created for each segment directory. For example, if you had 16 segment directories before migration, you would have 16 containers in each table space after migration. The following table depicts the segment directories prior to migration and shows two of the resulting table spaces after migration.

Version 1.x **Segment Directories**	**Version 5** **Table Space ID 0**	**Version 5** **Table Space ID 5**
segment_dir	*segment_dir* /SQLT*tablespace_id* *.segment_id*	*segment_dir* /SQLT*tablespace_id* *.segment_id*
SQLS0000 SQLS0001 SQLS0002 . . . SQLS0016	SQLS0000/SQLT0000.0 SQLS0001/SQLT0000.1 SQLS0002/SQLT0000.2 . . . SQLS0016/SQLT0000.16	SQLS0000/SQLT0005.0 SQLS0001/SQLT0005.1 SQLS0002/SQLT0005.2 . . . SQLS0016/SQLT0005.16

If you are migrating a Version 2.x database, there are no changes to the table spaces (SMS table space remain as SMS; DMS table spaces remain as DMS).

- The following node groups are created: IBMCATGROUP, IBMDEFAULTGROUP, and IBMTEMPGROUP. For more information, see “Nodegroups and Data Partitioning” on page 26.
- The following table spaces are created for Version 1.x databases that are migrated:

  SYSCATSPACE Belongs to the IBMCATGROUP nodegroup on the catalog node and contains system catalog tables.

  TEMPSPACE1 Belongs to the IBMTEMPGROUP nodegroup and is empty.

  USERSPACE1 Belongs to the IBMDEFAULTGROUP nodegroup and contains all user tables.

Migrating Partitioned Databases

After you migrate a partitioned database from a previous version of DB2, the following changes are made to your database:

- The database is relocated from the *INSTHOME/INSTNAME/NODEnnnnn* directory to the *INSTHOME/INSTNAME/NODEnnnn* directory. Note that node number is shorter by one digit.
- The database is migrated to SMS table space containers. If you are migrating a database you created using DB2 Parallel Edition Version 1.x, a table space container is created for each segment directory. For example, if you had 16 segment directories before migration, you would have 16 containers in each table space after migration. The following table depicts the segment directories prior to migration and shows two of the resulting table spaces after migration:

Version 1.x **Segment Directories**	**Version 5** **Table Space ID 0**	**Version 5** **Table Space ID 5**
segment_dir	*segment_dir* /SQLT*tablespace_id* *.segment_id*	*segment_dir* /SQLT*tablespace_id* *.segment_id*
SQLS0000 SQLS0001 SQLS0002 . . . SQLS0016	SQLS0000/SQLT0000.0 SQLS0001/SQLT0000.1 SQLS0002/SQLT0000.2 . . . SQLS0016/SQLT0000.16	SQLS0000/SQLT0005.0 SQLS0001/SQLT0005.1 SQLS0002/SQLT0005.2 . . . SQLS0016/SQLT0005.16

- One table space is created for each nodegroup. The name of the table space is the same as the name of the nodegroup. In addition to the table spaces that are created for your nodegroups, the following table spaces are created for the three IBM nodegroups (IBMDEFAULTGROUP, IBMCATGROUP, and IBMTEMPGROUP):

  IBMCATGROUP Contains user tables that were in the IBMCATGROUP nodegroup.

  IBMDEFAULTGROUP Contains user tables that were in the IBMDEFAULTGROUP nodegroup.

  SYSCATSPACE Belongs to the IBMCATGROUP nodegroup on the catalog node and contains system catalog tables.

  TEMPSPACE1 Belongs to the IBMTEMPGROUP nodegroup and is empty after the migration.

  USERSPACE1 Belongs to the IBMDEFAULTGROUP nodegroup and contains all user tables that were in that nodegroup in the originating database.

Pre-Migration

To ensure that all databases on your system can be migrated to DB2 Version 5, perform the following steps:

1 Complete all database transactions.

2 Ensure all applications disconnect from each database.

3 Ensure all databases are cataloged.

> The **db2ckmig** database migration verification command does not verify uncataloged databases. See “Verify that Databases Can Be Migrated” on page 95 for a description of how **db2ckmig** works.

4 Make a backup copy of all databases. Refer to the *Administration Guide* for your DB2 product for information on making a backup copy of a database and to the *Command Reference* for the syntax of the backup command.

Note: Make sure that this is the most recent backup copy of the database before you start the next procedure.

5 Stop the database manager by issuing the **db2stop** command.

6 If you are migrating from DB2 Parallel Edition Version 1.x, make a backup copy of the *INSTHOME/sqllib/db2nodes.cfg* file and store it in a different directory.

Migrating Instances

This procedure describes how to migrate DB2 instances that were created using a previous version of DB2.

> To update a Version 5 single-partition database system to a Version 5 partitioned database system, you must update the instance using the **db2iupdt** command. For more information, see “Updating Instances” on page 180.

Each DB2 instance must be migrated separately. To successfully migrate a DB2 instance, you need to perform the following steps:

1 Prepare the DB2 instance for migration.

2 Verify that databases can be migrated. This step is optional; however, we recommend that you perform this step before migrating an instance.

3. Ensure that the user exit program can be migrated.
4. Migrate the DB2 instance.

Note: If you want to migrate several instances, you must repeat these steps for each instance.

Prepare the DB2 Instance for Migration

Before you can migrate a DB2 instance, all applications using any databases owned by this instance must be completed. To prepare a DB2 instance for migration, you need to perform the following steps:

1. Log in as the DB2 instance.
2. Ensure that there are no applications using any databases owned by this DB2 instance. You can get the list of all applications using databases owned by the instance by issuing the **db2 list applications** command.
3. When all applications are complete, stop all database server processes owned by the DB2 instance by issuing the **db2stop** command.
4. Stop all command line processor sessions by issuing the **db2 terminate** command in each session that was running the command line processor.
5. Issue the **db2_kill** command to clean up any remaining DB2 resources.
6. Log off.

The DB2 instance is now ready for migration.

Verify that Databases Can Be Migrated

DB2 provides the **db2ckmig** migration command. This command verifies whether all cataloged databases can be migrated.

> To ensure that you can migrate all databases to DB2 Extended Enterprise Edition, you should run the **db2ckmig** command before you migrate the instance.

You must correct errors reported by this command before continuing with instance migration. For detailed information about the **db2ckmig** command, refer to the Version 5 *Command Reference*.

For example, to verify that all cataloged databases can be migrated, log in as the instance owner and issue the following command:

```
DB2DIR/instance/db2ckmig -h -a 0 -l INSTHOME/migration.log
```

where DB2DIR	= /usr/lpp/db2_05_00	on AIX
	= /opt/IBMdb2/V5.0	on Solaris

and *INSTHOME* is the home directory of the instance and *migration.log* is the name for the output file.

Check the log file; for example, *INSTHOME/migration.log*. If it shows any errors, see Table 8 for suggested corrective actions. In the partitioned database system, errors may be returned at the database partition level.

Note: The log file displays the errors that occur when you run the **db2ckmig** command. Check that the log is empty before continuing with the instance migration. Backup the database after making corrections.

Table 8 (Page 1 of 3). Correcting Error Messages

Error	**Action**
A database is in backup pending state	Perform a backup of the database.
A database is in roll-forward pending state	Recover the database as required; perform or resume a roll-forward database to end of log and stop.
Table space ID is not in normal state	Recover the database and table space as required; perform or resume a roll-forward database to end of logs and stop. **Note:** This error does not apply to DB2 Parallel Edition or Version 1.x single-partition database.
A database is in an inconsistent state	Restart the database to return it to a consistent state.

Table 8 (Page 2 of 3). Correcting Error Messages

Error	Action
The database contains database objects that have a schema name of SYSCAT, SYSSTAT, or SYSFUN	These schema names are reserved for the Version 5 database manager. To correct this error, do the following: **1** Back up the database. **2** Export the data from the database object (catalogs or tables). **3** Drop the object. **4** Recreate the object with the corrected schema name. **5** Import/Load the data into the object. **6** Run the **db2ckmig** command against the database again, ensuring that the database passes the **db2ckmig** check. **7** Make a backup copy of the database.
The database contains database objects that have a dependency on the SYSFUN.DIFFERENCE function. Possible violated database objects are: • Constraint • Function • Trigger • View	The SYSFUN.DIFFERENCE function must be dropped and recreated during database migration. However, if there is a database object that is dependent on this function, migration will fail. To correct this error, do the following: **Constraint** Issue the **alter table** command to drop the constraint. **Function** Issue the **drop function** command to drop the function dependent on SYSFUN.DIFFERENCE. **Trigger** Issue the **drop trigger** command to drop the trigger. **View** Issue the **drop view** command to drop the view. **Notes:** 1. Any package dependent on the SYSFUN.DIFFERENCE function will be marked inoperative after migration. Therefore, the **db2ckmig** command will not report any package that is dependent on the SYSFUN.DIFFERENCE function. 2. This error does not apply to DB2 Parallel Edition or Version 1.x. single-partition database.

Table 8 (Page 3 of 3). Correcting Error Messages

Error	**Action**
The database contains user-defined distinct types that use the type name of DATALINK or REFERENCE.	The data type names, DATALINK and REFERENCE are reserved for the Version 5 database manager. To correct this error, do the following: **1** Back up the database. **2** Export the data from any tables that are dependent on the data types. **3** Drop any tables dependent on the data types, and then drop the data types. These drops may drop other objects such as views, indexes, triggers, or functions. **4** Create data types with different type names and recreate the tables using the new data type names. Recreate any dropped views, indexes, triggers, or functions. **5** Import/Load the data into the object. **6** Run the **db2ckmig** command against the database again, ensuring that the database passes the **db2ckmig** check. **7** Make a backup copy of the database. **Note:** This error does not apply to DB2 Parallel Edition or Version 1.x single-partition database.

All local databases now have the same authentication type as the instance where they reside; the authentication type in the database directory is ignored by DB2 Version 5 servers. If a warning is logged due to a conflicting authentication type, and you want a database to retain its previous authentication type, then you can do one of the following:

- Change the authentication type of the instance to the previous one.
- Move the database to another instance that has the required authentication type.

Before changing the authentication type of the instance, you should make sure that the new authentication type will be appropriate for all databases residing there. Be certain to consider the security implications of the different authentication types.

If there are databases that you do not want to migrate, you can uncatalog them (along with all aliases); the **db2ckmig** command does not perform any verification of uncataloged databases.

Refer to the *Administration Guide* for more information about the actions required to correct these conditions.

Migration Considerations for the User Exit Program

Follow these instructions if you are using the **db2uexit** user exit program with previous versions of DB2.

DB2 Version 5 has changed the interface it uses to invoke the user exit program to archive and retrieve log files. For more information on these new interfaces, refer to the *Administration Guide*. The name for the user exit program has changed to **db2uext2** in Version 5. (In previous versions, it was called **db2uexit**.)

The following should be considered before migrating instances:

- If the pre-Version 5 **db2uexit** program is installed in the *INSTHOME/sqllib/adm* directory before migration, it will remain in this directory after migration. The DB2 Version 5 **db2uext2** program will be also installed in this directory. Its function is to invoke **db2uexit** using the pre-Version 5 interface. This allows the old user exit program to be used on DB2 Version 5.
- If **db2uexit** is installed in a directory other than *INSTHOME/sqllib/adm*, it will not be installed after migration. For example, if **db2uexit** was in the *INSTHOME/sqllib/bin* directory, after migration the **db2uexit** file will not be in the *INSTHOME/sqllib/bin* directory. If you want to continue using the old user exit after migration, you must copy **db2uexit** to the *INSTHOME/sqllib/adm* directory. Then, you can do one of the following:
  - If you are migrating from DB2 Version 1.x or Version 2.x, copy **db2uext2.v2** from the *DB2DIR/misc* directory to the *INSTHOME/sqllib/adm* directory and rename it to **db2uext2**. You can use the following command to copy the file:

    ```
    cp DB2DIR/misc/db2uext2.v2 INSTHOME/sqllib/adm/db2uext2
    ```

where DB2DIR	= /usr/lpp/db2_05_00	on AIX
	= /opt/IBMdb2/V5.0	on Solaris

  - If you are migrating from DB2 Parallel Edition Version 1.x, copy **db2uext2.pe** from the *DB2DIR/misc* directory to the *INSTHOME/sqllib/adm* directory and rename it to **db2uext2**. You can use the following command to copy the file:

    ```
    cp DB2DIR/misc/db2uext2.pe INSTHOME/sqllib/adm/db2uext2
    ```

    where `DB2DIR` = */usr/lpp/db2_05_00* on AIX.

  Note: You must ensure that **db2uext2** is owned by the instance owner and is executable by the owner.

At a convenient time, you should modify your user exit program to use the new DB2 Version 5 interfaces. The new user exit program should replace **db2uext2** in the *INSTHOME/sqllib/bin* directory, used to support the pre-Version 5 user exit program, **db2uexit**, which should be removed.

Migrate the DB2 Instance

Only local cataloged databases that are owned by the DB2 instance are checked for migration. Uncataloged databases may be unusable after the instance has been migrated. Refer to the *Administration Guide* for further information.

After an instance is ready for migration, use the **db2imigr** command to migrate the instance as follows:

1 Log in as root.

2 Run the **db2imigr** command as follows:

```
DB2DIR/instance/db2imigr [-d] [-a AuthType] [-u fencedID] InstName
```

where DB2DIR	= /usr/lpp/db2_05_00	on AIX
	= /opt/IBMdb2/V5.0	on Solaris

and where:

-d
Sets the debug mode that you can use for problem determination

-a *AuthType*
Is an optional parameter that specifies the authentication for the instance. Valid authentication types are SERVER, CLIENT, and DCS. If the *-a* parameter is not specified, the authentication type defaults to SERVER, if a server product is installed. Otherwise, the *AuthType* is set to CLIENT.

Notes:

a. The authentication type of the instance applies to all databases owned by the instance.

b. Authentication type DCE is valid. However, it is not valid to choose DCE as an *AuthType* for this command. To enable DCE authentication for a DB2 instance, refer to the *Administration Guide*.

-u *fencedID*
Is the user under which the fenced user defined functions (UDFs) and stored procedures will execute.

InstName
Is the login name of the instance owner.

Because the *INSTHOME* directory is NFS mounted on all machines, you only have to run the **db2imigr** command on one machine to migrate the entire instance.

3 If there are any errors in verifying that all databases can be migrated, see Table 8 on page 96 and take the suggested corrective actions. Then, reissue the **db2imigr** command.

If you are migrating a DB2 Version 2.1 instance, created on AIX, and the instance uses the environment variable*DB2SORT* set to a keyword *SMARTSORT*, you must set the registry value *db2sort* after the instance is migrated to Version 5. Set the *db2sort* registry value to the runtime library for the sort command as follows:

```
db2set DB2SORT="/usr/lib/libsort.a"
```

Migrating Databases

To migrate databases owned by an instance, you need to perform the following steps:

1 Log on with a user ID that has SYSADM authority.

2 Ensure that the databases you want to migrate are cataloged.

3 Migrate the database. Refer to the *Command Reference* for the database migration syntax command.

Specifying Table Spaces when Migrating DB2 Parallel Edition Version 1.x

When you migrate DB2 Parallel Edition Version 1.x to DB2 Extended Enterprise Edition Version 5, the migration command will automatically create a default set of SMS table spaces and migrate your tables to one of these table spaces. This default behavior will work well for many of your tables. You may, however, want to migrate some tables to specific table spaces that you identify ahead of time.

For example, one very important consideration is that with DB2 Extended Enterprise Edition Version 5, when you load a table, its table space is placed in a quiesce state for the duration of the load. This means that other tables in the table space are not accessible during the load. (This is not the case with DB2 Parallel Edition Version 1.x, as it does not support table spaces.) So, if you regularly perform large loads and depend on other tables being available during the loads, you should identify the target tables of these loads, and using the information described below, migrate each to its own dedicated table space.

Other reasons you may want to customize table spaces at migration time include:

- DB2 Extended Enterprise Edition Version 5's point-in-time table space roll-forward recovery can be simplified if related tables are in the same table space.
- To take advantage of DB2 Extended Enterprise Edition Version 5's ability to restore a specified subset of table spaces from any backup.

By default, when you migrate DB2 Parallel Edition to DB2 Extended Enterprise Edition Version 5, the following will occur:

- An SMS table space will be created for every nodegroup (the table space name will be the same as the nodegroup's name).
- Every table will be migrated to the table space created for the nodegroup.

For example, if you have:

```
nodegroup NG1 consisting of nodes 1, 2 and 3
nodegroup NG2 consisting of nodes 3 and 4
```

and

```
table TA in NG1
table TB in NG2
table TC in NG1
table TD in NG1
```

Migration will create two SMS table spaces, named NG1 and NG2 and migrate tables TA, TC and TD to NG1, and table TB to NG2. Table space NG1 will exist in nodegroup NG1 (nodes 1, 2, and 3). Table space NG2 will exist in nodegroup NG2 (nodes 3 and 4). This is the default behavior

You can override the default behavior as follows:

1 Create a file that lists specific tables and which table space each table should be migrated to. The format of the file is:

```
line 1: <d>
line 2: <d><schema><d><table><d><tablespace><d>
line 3: <d><schema><d><table><d><tablespace><d>
        .
        .
        .
line n: <d><schema><d><table><d><tablespace><d>
```

where <d> is a single character delimiter. An example is as follows:

```
path  :  /a/b/c
---------------
!"              !
!"S1"TC"TS1     !
!"S1"TD"TS2"    !
----------------
```

This file indicates to create SMS table spaces TS1 and TS2, and to migrate table TC (under schema S1) and table TD (under schema S1) to TS2. The delimiter used is this file is the " character.

> If you use a delimiter of `' '` (a blank), ensure that the first line actually contains a blank in the first column (a new line character in the first column is not treated as a blank).

2. Ensure that the DB2 instance owner has permission the read the file.

3. Run the **db2set** command as follows:

   ```
   db2set DB2_MIGRATE_TS_INFO=/a/b/c
   ```

   where `a/b/c` is the fully qualified path to the above file. (Alternatively, you can specify `DB2_MIGRATE_TS_INFO` in your *db2profile* file. Using the **db2set** command is the recommended method.)

4. Run the **db2stop** command.

5. Run the **db2start** command.

6. Perform the migration.

   This results in two additional SMS table spaces being created during migration: TS1 and TS2. Table TC will be migrated to TS1 and table TD will be migrated to TS2. Any table not listed in the file will be migrated according to the default behavior. Continuing with the previous example, table TA will be migrated to table space NG and table TB will be migrated to table space NG2.

You should be aware of the following:

- All identifiers in the file are case sensitive.
- You can specify the same table space more than once in the file. If you do, ensure that all tables to be migrated to the table space exist in the same nodegroup.
- You cannot specify the same table more than once.
- Any errors that occur when the file is processed are reported in the *db2diag.log* file. An SQLCODE of -901 will be reported and the migration will be rolled back. If this occurs, correct the problems reported in the *db2diag.log* file and try the migration again.
- This procedure has been tested to create up to 100 additional table spaces. You should be able to create more, but this has not been tested. (Note that the maximum number of table spaces per database is 4096.)

Post Migration

There are optional activities you may want to undertake following database migration.

> You can also apply these optional activities to a down-level database backup which is restored to Version 5, because at the end of the restore, the database is migrated to Version 5.

- **Unique index conversion to DB2 Universal Database Version 5 semantics**

  Version 5 of DB2 supports deferred unique constraint checking until the end of a statement. This can result in correct processing of multiple row updates, that in previous releases of DB2, returned an error because the updates temporarily created duplicate values in the transient state. Deferred unique constraint checking will guarantee that updates that result in a table with only unique keys (for example, key = key + 1) will succeed, regardless of the order of the data.

  Note: This change only applies for unique indexes that are created in Version 5 of DB2.

  All unique indexes in a migrated database do not automatically migrate to Version 5 semantics during database migration because of the following reasons:

  – Converting unique indexes is a very time-consuming operation.
  – You may have applications that depend on the previous version's unique index semantics.
  – You may want to manage the staged conversion of unique indexes on your own schedule, when needed.

> All existing applications will continue to work even if the unique indexes are not converted to Version 5 semantics. You have to convert unique indexes to Version 5 semantics only if support for deferred uniqueness checking is required.

To convert unique indexes, you need to perform the following steps:

1 Log in with a user ID that has SYSADM authority.

2 Run the **db2start** command.

3 Run the **db2uiddl** command against your migrated database. Refer to the Version 5 *Command Reference* for the syntax of this command.

The **db2uiddl** command searches the database catalog tables and generates all the CREATE UNIQUE INDEX statements for user tables in an output file.

4. Review the output generated from the **db2uiddl** command, and make changes, if needed. Comments in the output will flag any situations that require your attention.

5. Connect to the database by issuing the **db2 connect to** *database_alias* command.

6. Execute the file as a DB2 Command Line Processor command file, using a command similar to the following:

   ```
   db2 -tvf filename
   ```

   where `filename` is an output file from **db2uiddl**.

7. Disconnect from the database by issuing the **db2 connect reset** command.

DB2 interprets the re-creation of an existing unique index to signal that the index is ready to be converted to Version 5 semantics.

- **Update Statistics**

  When database migration is completed, the old statistics that are used to optimize query performance are retained in the catalogs. However, Version 5 of DB2 has statistics that are modified or do not exist in the previous version. To take advantage of these, you may want to issue the **runstats** command on tables, particularly those tables that are critical to the performance of your SQL queries.

  Refer to the Version 5 *Command Reference* for the syntax of the **runstats** command. For details on the statistics, refer to the Version 5 *Administration Guide*.

- **Rebind Packages**

  During database migration, all existing packages are invalidated. After the migration, each package is rebuilt when it is used for the first time by the Version 5 database manager. However, for better performance, we recommend that you run the **db2rbind** command to rebuild all packages stored in the database, after database migration is complete. Refer to the Version 5 *Command Reference* for the syntax of this command.

- **Update database and database manager configuration**

  Some of the database configuration parameters are changed to Version 5 defaults or to other values during database migration. The same is true for database manager configuration parameters which may have changed, during instance migration, to Version 5 defaults or to other values. Refer to the *Administration Guide* for more information about configuration parameters.

  The following database manager configuration parameters are changed to Version 5 defaults:

  – *Sort heap threshold (sheapthres)*

    If the migrating database configuration file has this parameter at a value which is less than the Version 5 default, the parameter is reset to its Version 5 default value.

  – *Backup buffer (backbufsz)*

If the migrating database configuration file has this parameter at a value which is less than the Version 5 minimum, the parameter is reset to its Version 5 minimum value.

- *Restore buffer (restbufsz)*

  If the migrating database configuration file has this parameter at a value which is less than the Version 5 minimum, the parameter is reset to its Version 5 minimum value.

- *Maximum number of concurrent active databases (numdb)*

  If the migrating database configuration file has this parameter at a value which is greater than the Version 5 maximum, the parameter is reset to its Version 5 maximum value.

- *TM-database (tm_database)*

  If the migrating database configuration file has this parameter set to NULL, the parameter is reset to 1ST_CONNECT.

- *Query heap size (query_heap_sz)*

  If the migrating database configuration file has this parameter at a value less than the Application Support Layer heap size (aslheapsz) of the same file, the parameter is reset to aslheapsz + 1.

- *Agent pool size (num_poolagents)*

  If the migrating database configuration file has the value of this parameter greater than the value of the Maximum Number of agents (maxagents) of the same file, the parameter is reset to the value of maxagents - 1.

The following database configuration parameters are changed to Version 5 defaults:

- *Application Control heap size (app_ctl_heap_sz)*

  If the migrating database configuration file has this parameter at a value which is less than the Version 5 default, the parameter is reset to its Version 5 default value.

  Note: This only applies to DB2 Parallel Edition database migration to DB2 Version 5, because this parameter exists in DB2 Parallel Edition Version 1.x databases only. For other downlevel databases, this parameter is set to its default value after migration completes.

- *Lock List (locklist)*

  For Version 1 DB2 databases, the lock list parameter will be first adjusted to locklist * 32 / 25. If the migrating database configuration file has this parameter at a value less than the Version 5 default value, the parameter is reset to its Version 5 default value.

- *Database Heap (dbheap)*

  For Version 1 DB2 databases, the database heap size will first be adjusted to dbheap * 16. If the migrating database configuration file has this parameter at a value that is less than the Version 5 default, the parameter is reset to its Version 5 default value.

– *Default log space (logfilsiz)*

  The database migration process will attempt to increase the *logfilsiz* value, if the log file related parameters have a total logfilsiz that is less than the default logfilsiz value.

– *Application Heap size (applheapsz)*

  For Version 1 DB2 databases, application heap size will be first adjusted to applheapsz * 16. If the migrating database configuration file has this parameter at a value which is less than the Version 5 default, the parameter is reset to its Version 5 default value.

– *Sort heap (sortheap)*

  For Version 1 DB2 databases, the value of the sort heap parameter will be adjusted to sortheap * 16.

For better performance, you may want to tune your database and database manager configuration parameters to take advantage of Version 5 enhancements. Refer to the *Command Reference* for the syntax of updating database and database manager configuration.

- **Migrate Explain Tables**

If you are not using explain tables in the version of DB2 that you are migrating, skip this task.

Version 5 of DB2 has added several new columns to the explain tables. These columns provide for the capture of:

– Data for new SQL features added in Version 5
– More detailed access plan information.

While the explain function in Version 5 will continue to work with explain tables created for Version 2, the new Version 5 data will not be captured in them.

For better performance of SQL statements, we recommend that the Version 2 explain tables be dropped and new explain tables be created; refer to the *SQL Reference* and the *Administration Guide* for details on creating new explain tables. If, however, there are Version 2 explain tables that you need for ongoing comparison, you can use the EXPLMIG.DDL script to migrate them.

To migrate the explain tables in a database that has been migrated to Version 5, connect to the database and run the following command from the *sqllib/misc* directory:

```
db2 -tf EXPLMIG.DDL
```

The explain tables belonging to the user ID that is used to connect to the database will be migrated. To migrate explain tables belonging to another user, connect to the database with that user ID and run the command.

Part 3. Installing DB2 Clients

Chapter 8. Installing DB2 Clients

The DB2 Client Application Enabler is installed with any DB2 Version 5 product. If a DB2 Version 5 product currently exists on this workstation, the DB2 Client Application Enabler component has already been installed, and there is no need to install a DB2 Client Application Enabler. To configure your client to access remote servers, go to Chapter 14, "Configuring Client-to-Server Communications Using the CCA" on page 143.

Use the instructions in this section to install a DB2 client. You can also run unattended installations of DB2 to remote target machines running OS/2, Windows 3.x, Windows 95, or Windows NT.

Clients on a LAN that will connect to a DB2 Universal Database server must have the DB2 Client Application Enabler or the DB2 Software Developer's Kit installed.

The DB2 Client Application Enablers (except the DOS client) are on the DB2 Client Application Enablers CD-ROM.

DB2 Client Application Enablers and DB2 Software Developer's Kits are available for the following operating systems: AIX, DOS, HP-UX, Macintosh, OS/2, SCO OpenServer, Silicon Graphics IRIX, SINIX, Solaris, Windows 3.1x, Windows 95, and Windows NT.

You may install DB2 Client Application Enablers on any number of workstations; there are no licensing restrictions for installation. DB2 Extended Enterprise Edition enables an unlimited number of users to connect to the server.

Go to the section that gives instructions for the DB2 client that you want to install.

- Chapter 9, "Installing DB2 Clients on OS/2 Workstations" on page 113.
- Chapter 10, "Installing DB2 Clients on Windows 95 or Windows NT Workstations" on page 115.
- Chapter 11, "Installing DB2 Clients on Windows 3.x Workstations" on page 119.
- Chapter 12, "Installing DB2 Clients on UNIX Workstations" on page 123.
- Chapter 13, "Installing DB2 Clients on Macintosh Workstations" on page 139.

To obtain DB2 Client Application Enablers for all clients, including DOS, connect to the IBM Web site (http://www.software.ibm.com/data/db2/db2tech/clientpak.html), search for the `Client Application Enablers`, and follow the instructions provided.

The DB2 Software Developer's Kit on the DB2 Extended Enterprise Edition CD-ROM has been upgraded so that it can communicate with a DB2 Extended Enterprise Edition server. If you install a DB2 Software Developer's Kit from the DB2 Software Developer's Kit CD-ROM, you must apply a patch to enable them to communicate with a DB2 Extended Enterprise Edition server. To obtain the patch, referred to as a maintenance package, connect to the IBM Web site (http://www.software.ibm.com/data/db2/db2tech).

Chapter 9. Installing DB2 Clients on OS/2 Workstations

This section contains the information you need to install the DB2 Client Application Enabler on OS/2 workstations.

To install the DB2 Client Application Enabler for OS/2:

1 Insert the DB2 Client Application Enablers CD-ROM that contains the OS/2 client into the drive.

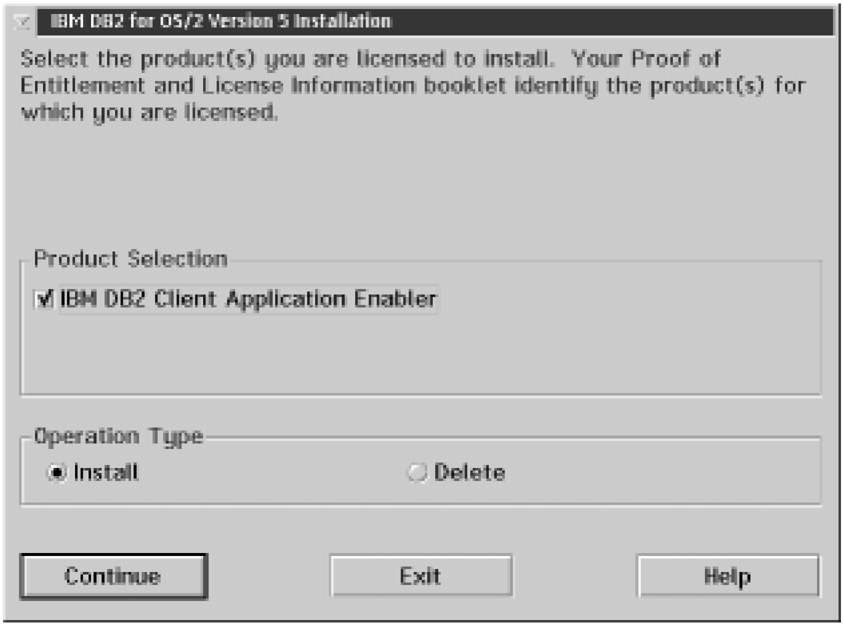
2 Begin the installation program as follows:

a Open an OS/2 window and set the drive to *x*:, where *x* is the letter that represents your CD-ROM drive.

b Set the current directory to \db2\os2\language\install, where *language* is the two-character country code that represents your language (for example, *EN* for English). Table 28 on page 314 lists the code for each available language.

c Issue the **install** command.

The IBM DB2 for OS/2 Version 5 Installation window opens.



IBM DB2 for OS/2 Version 5 Installation

Select the product(s) you are licensed to install. Your Proof of Entitlement and License Information booklet identify the product(s) for which you are licensed.

Product Selection

☑ IBM DB2 Client Application Enabler

Operation Type

● Install ○ Delete

To select the product that you want to install on your system, click on the **Continue** push button. The Update CONFIG.SYS File window opens.

3. Indicate whether or not you want to update the *config.sys* file as part of the installation.

   - If you want the *config.sys* file updated, click on **OK**. A backup copy of your existing file is kept as *config.0xx*.
   - If you do not want the *config.sys* file updated, clear the **Update CONFIG.SYS** check box and click on **OK**. You will need to manually update your *config.sys* file with the values that are stored in the *config.add* file.

4. The Install - Directories window provides a list of the product's components. You can choose which of the components you want to install on your system.

   For the Install - Directories window, complete the entries as follows:

   a. Select each of the components you want to install.

      You have the option of installing the Client Configuration Assistant component. The Client Configuration Assistant makes it easy for you to configure and administer connections to databases and to bind applications and utilities to a database. If the Client Configuration Assistant is not installed, you will need to manually configure connections to DB2 servers. Refer to “Managing Connections to Databases Using the Client Configuration Assistant” on page 12 for a complete description of the Client Configuration Assistant's features.

   b. In the **File directory** field, type the directory where you want the product installed. The default value is *c:\sqllib*.

      Note: If a DB2 Version 5 product is already installed on the workstation, you must install on the same drive and directory.

   c. Click on the **Disk Space** push button to see how much space you have on each of the drives on your machine and to change disks if necessary.

   d. Click on the **Install** push button to continue.

A progress window appears until the installation is complete. You can click on the **Stop** push button at any time to end the installation.

5. At the end of the installation, shut down and reboot your workstation before you can use the client.

Note: If you want to run Windows 3.1 applications on your OS/2 client, you must also install the DB2 Client Application Enabler for Windows 3.1 component on your system. You should install this client component in the same directory as DB2 Client Application Enabler for OS/2. Having both DB2 Client Application Enablers in the same directory merges the DB2 directories and allows the CCA to be used to configure the access to remote DB2 databases for both OS/2 and Windows 3.x applications.

To configure your client to access remote servers, go to Chapter 14, “Configuring Client-to-Server Communications Using the CCA” on page 143.

Chapter 10. Installing DB2 Clients on Windows 95 or Windows NT Workstations

This section contains the information you need to install the DB2 Client Application Enabler on Windows 95 or Windows NT workstations.

Before You Begin

Read and perform the steps in this section before you begin the installation to ensure that you have the items and information that you will need.

Performing the Installation

To install a DB2 Client Application Enabler for Windows 95 or Windows NT:

1 Log on as a local Administrator.

2 Shut down any other programs so that the setup program can update files as required.

3 Insert the DB2 Client Application Enablers CD-ROM that contains the Windows 95 or Windows NT client into the drive. The auto-run feature automatically starts the setup program. The setup program will determine the system language, and launch the setup program for that language. If you want to run the setup program in a different language, see the tip that follows.

To manually invoke the setup program, do the following:

1 Click on **Start** and select **Run**

2 Type the following in the **Open** field:

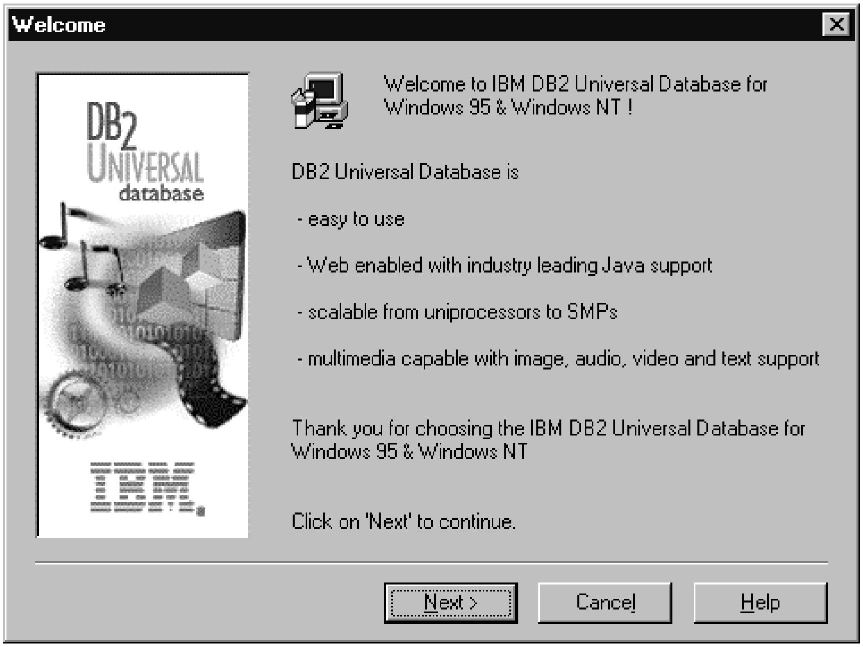
```
x:\setup /i language
```

where:

- *x*: represents your CD-ROM drive
- *language* represents the two-character country code for your language (for example, EN for English). Table 28 on page 314 lists the code for each available language.

3 Select **OK**.

4 The Welcome window opens.



Welcome to IBM DB2 Universal Database for Windows 95 & Windows NT !

DB2 Universal Database is

- easy to use
- Web enabled with industry leading Java support
- scalable from uniprocessors to SMPs
- multimedia capable with image, audio, video and text support

Thank you for choosing the IBM DB2 Universal Database for Windows 95 & Windows NT

Click on 'Next' to continue.

Click on the **Next** push button to continue.

5 The Enable Remote Administration window opens. Select the **Install components required to administer remote servers** check box if you would like to administer remote servers from this client, then click on the **Next** push button.

6 Select the installation type you prefer:

Typical Install: Installs those DB2 components that are used most often, including all required components, ODBC support, documentation, and commonly used DB2 tools such as the Client Configuration Assistant and the Information Center. The DB2 instance is created.

Compact Install: Installs only the required DB2 components and ODBC support. The DB2 instance is created.

Custom Install: Installs only those components that you select. The DB2 instance is created.

The Client Configuration Assistant (CCA) component makes it easy for you to configure and administer connections to databases and bind applications and utilities to a database. See “Managing Connections to Databases Using the Client Configuration Assistant” on page 12 for a complete description of the CCA's features.

If the CCA is not installed, you will need to use the command line processor to configure connections to a DB2 server.

See Chapter 15, “Configuring Client-to-Server Communications Using the Command Line Processor” on page 147 to configure clients using the command line processor.

7 Respond to the setup program's prompts. Online help is available to walk you through the remaining steps. Invoke online help by clicking on the **Help** push button at any time.

You can click on the **Cancel** push button at any time to end the installation.

8 After you install the product, you must reboot your workstation before you can begin to use it. Select a reboot option and click on the **Finish** push button. This completes the installation.

For information on errors encountered during installation, see the *db2.log* file. The *db2.log* file stores general information and error messages resulting from installation and uninstall activities. By default, the *db2.log* file is located in the *x:\db2log* directory; where x: is the drive on which your operating system is installed.

The installation program has:

- Created DB2 program groups and items (or shortcuts).
- Updated the Windows registry.
- Created a default client instance called DB2.
- Registered a security service.

To configure your client to access remote servers, go to Chapter 14, “Configuring Client-to-Server Communications Using the CCA” on page 143.

Chapter 11. Installing DB2 Clients on Windows 3.x Workstations

If you have a previous version of the DB2 Client Application Enabler for Windows 3.x installed on your system, it is recommended that you back it up before beginning this installation procedure.

To install the DB2 Client Application Enabler on Windows 3.x workstations:

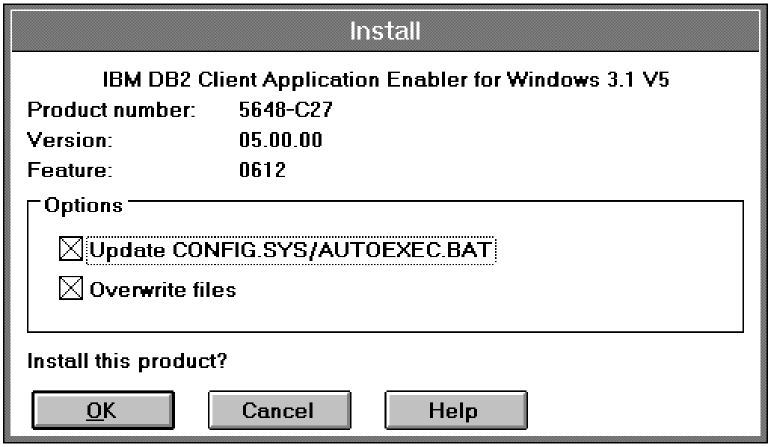
1 Insert the DB2 Client Application Enablers CD-ROM that contains the Windows 3.x client into the drive.

2 To begin the installation program enter:

```
x:\db2\windows\language\install\install.exe
```

where *x:* is the letter representing your CD-ROM drive and where *language* is the two-character country code that represents your language (for example, *EN* for English). Table 28 on page 314 lists the code for each available language.

The Install window opens.



Install

IBM DB2 Client Application Enabler for Windows 3.1 V5

Product number: 5648-C27

Version: 05.00.00

Feature: 0612

Options

☒ Update CONFIG.SYS/AUTOEXEC.BAT

☒ Overwrite files

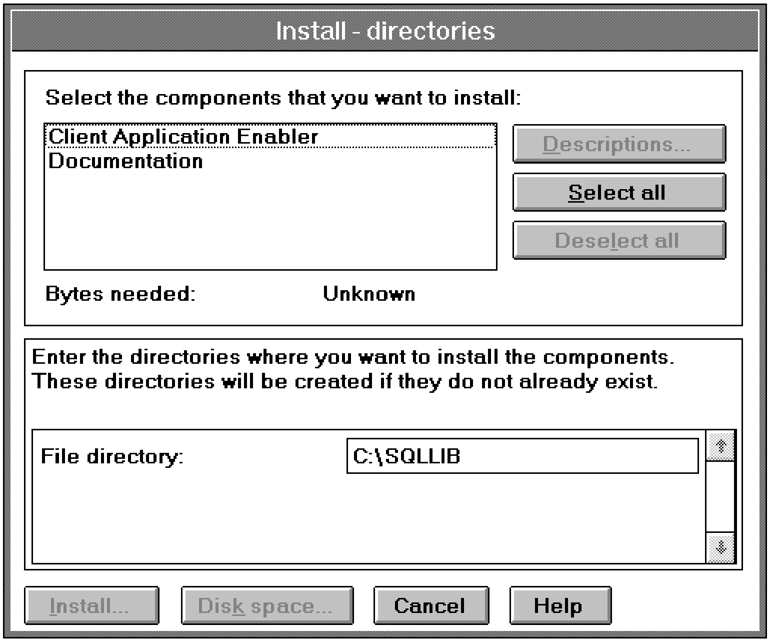
Install this product?

3 In the Install window, indicate whether or not you want to update the *autoexec.bat* file as part of the installation.

- If you want the *autoexec.bat* file updated, check that the **Update CONFIG.SYS/AUTOEXEC.BAT** check box is selected, then click on **OK**. A backup copy, called *autoexec.bak*, is created in the same directory as your *autoexec.bat* file.
- If you do not want the *autoexec.bat* file updated, clear the **Update CONFIG.SYS/AUTOEXEC.BAT** check box. You will need to manually update your *autoexec.bat* file with the values that are stored in the *autoexec.add* file.

The **Overwrite files** check box is selected by default, it is recommend that you do not change this setting.

4 Click on **OK** to continue. The Install-directories window opens.



Install - directories

Select the components that you want to install:

Client Application Enabler
Documentation

Bytes needed: Unknown

Enter the directories where you want to install the components. These directories will be created if they do not already exist.

File directory: C:\SQLLIB

5 In the **File directory** field, type the directory where you want the product installed. The default value is *c:\sqllib*.

If you are installing the DB2 Client Application Enabler for Windows 3.x on an OS/2 system that already has the DB2 Client Application Enabler for OS/2 Version 5 product installed, enter the same directory in the **File directory** field as the DB2 Client Application Enabler for OS/2 product. This will merge the two DB2 directories together, allowing tools like the Client Configuration Assistant to be used to configure clients to access a remote DB2 database for both Windows 3.x and OS/2 applications.

6 Select the components that you want to install.

7 Click on the **Install** push button to start the installation. A progress window appears until the installation is complete.

You can click on the **Stop** push button at any time to end the installation.

8 Once you have completed the installation, shut down and reboot your system.

To configure your client to access remote servers, go to Chapter 14, “Configuring Client-to-Server Communications Using the CCA” on page 143.

Chapter 12. Installing DB2 Clients on UNIX Workstations

This section contains the information you need to install the DB2 Client Application Enabler on UNIX systems. Refer to the following sections as appropriate:

- "Installing on AIX, HP-UX and Solaris Systems" on page 124.
- "Installing on Other UNIX Systems" on page 132.

Before You Begin

Before you begin installing DB2 products using the DB2 Installer program, you need to gather the following information:

Where is the CD-ROM mount point?
You need to mount the CD-ROM before you can install the DB2 product. To mount the CD-ROM, you must know where the mount point is. For example, the CD-ROM mount point could be `/cdrom`.

How much disk space does my operating system require for all selected products?
See "Estimating Fixed Disk Requirements" on page 39 and "Server Components" on page 40 to find out how much disk space is required for selected products and how to estimate disk space requirements.

How do I assign a group name and user name for a DB2 instance?
A separate user name must be used for each DB2 Instance. We recommend that a new group be created which should be used as a primary group for the DB2 instance user.

Installing on AIX, HP-UX and Solaris Systems

To install the DB2 Client Application Enabler on AIX, HP-UX and Solaris systems, perform the following steps:

Step 1. Update Kernel Configuration Parameters

To run DB2 Client Application Enabler for UNIX systems, you may have to update some kernel configuration parameters.

This step is not required on AIX. If you are installing an AIX client, skip this step and go to “Step 2. Mount the DB2 Client Application Enablers CD-ROM” on page 126

Recommended Values for HP-UX

The following values in Table 9 are the recommended HP-UX kernel configuration parameters.

Table 9. HP-UX Kernel Configuration Parameters (Recommended Values)

Kernel Parameter	Recommended Value
`msgseg`	`8192`
`msgmnb`	`65535 (1)`
`msgmax`	`65535 (1)`
`msgssz`	`16`

Notes:

1. Parameters *msgmnb* and *msgmax* must be set to 65535.
2. To maintain the interdependency among kernel parameters, change parameters in the same sequence in which they appear in the preceding table.

To change the values, do the following:

1 Use **SAM** and select **Kernel Configuration**.

2 Select **Configurable Parameters**.

3 Highlight the parameter to be changed.

4. Select **Modify Configurable Parameter** from the **Actions** menu and make the appropriate changes.
5. Repeat the previous two steps for every kernel parameter which needs to be updated.
6. Create a new kernel by selecting **Create a New Kernel** from the **Actions** menu.
7. Reboot the system so that the changes can take effect.

Go to "Step 2. Mount the DB2 Client Application Enablers CD-ROM" on page 126 to continue with the installation.

Recommended Values for Solaris

The following values in Table 10 are the recommend Solaris kernel configuration parameters.

Table 10. Solaris Kernel Configuration Parameters (Recommended Values)

Kernel Parameter	Recommended Value
msgsys:msginfo_msgmax	65535 (1)
msgsys:msginfo_msgmnb	65535 (1)
msgsys:msginfo_msgseg	8192
msgsys:msginfo_msgssz	16

Notes:

1. Parameters *msgsys:msginfo_msgmnb* and *msgsys:msginfo_msgmax* must be set to 65535.

To set a kernel parameter, add a line at the end of the */etc/system* file as follows:

```
set parameter-name = value
```

For example, to set the value of parameter *msgsys:msginfo_msgmax*, add the following line to the end of the */etc/system* file:

```
set msgsys:msginfo_msgmax = 65535
```

After changing the kernel parameters, reboot the system so that the changes can take effect.

Step 2. Mount the DB2 Client Application Enablers CD-ROM

To install DB2 Client Application Enabler using the **DB2 Installer**, you must first mount the CD-ROM. Use the DB2 Client Application Enablers CD-ROM to perform the installation. Once you have mounted the CD-ROM, you can start installing the DB2 products.

Mounting on AIX Systems

Perform the following steps to mount the CD-ROM on AIX operating systems:

1 Log in as root.

2 Insert the CD-ROM in the drive.

3 Create a root directory by entering the following command:

```
mkdir -p /cdrom
```

4 Allocate a CD-ROM file system by typing the following command:

```
smit storage
```

5 Select **FileSystems**.

6 Select **Add/Change/Show/Delete FileSystems**.

7 Select **CD-ROM FileSystems**.

8 Select **Add CDROM FileSystems**.

9 Select the **DEVICE Name**.

Device names for CD-ROM file systems must be unique. You may need to delete a previously-defined CD-ROM file system or use another name for your directory, if there is a duplicate device name.

10 Respond to the `mount point` prompt by typing the following:

```
/cdrom
```

11 Mount the CD-ROM file system by typing the following command:

```
smit mountfs
```

12 Select the **FileSystem** name. For example, the name could be `/dev/cd0`.

13 Select the **Directory** name, `/cdrom`.

14 Select the **Type of filesystem**, `cdrfs`.

15 Set the **Mount as READ-ONLY system** to `Yes`.

16 Log off.

Go to “Step 3. Perform the Installation” on page 128 to continue with the installation.

Mounting on HP-UX Systems

Perform the following steps to mount the CD-ROM on HP-UX operating systems:

1 Log in as root.

2 Insert the CD-ROM in the drive and mount it as in the following example:

```
mkdir /cdrom
/usr/sbin/mount /dev/dsk/c0t2d0 /cdrom
```

where `/cdrom` is the CD-ROM mount directory.

3 Log out.

Go to “Step 3. Perform the Installation” on page 128 to continue with the installation.

Mounting on Solaris Systems

Perform the following steps to mount the CD-ROM on Solaris operating systems:

1 Log in as root.

2 If the Volume Manager is not running on your system, enter the following commands to mount the CD-ROM:

```
mkdir -p /cdrom/unnamed_cdrom
mount -F hsfs -o ro /dev/dsk/c0t6d0s2 /cdrom/unnamed_cdrom
```

where */cdrom/unnamed_cdrom* is the CD-ROM mount directory.

> If you are mounting the CD-ROM drive from a remote system using NFS, the CD-ROM file system on the remote machine must be exported with *root* access. You must also mount that file system with *root* access on the local machine.

If the Volume Manager (vold) is running on your system, the CD-ROM is automatically mounted as:

```
/cdrom/unnamed_cdrom
```

3. Log out.

Step 3. Perform the Installation

After you mount the CD-ROM file system, use the DB2 Installer program to install the DB2 products.

If you are installing the DB2 Client Application Enabler from a remote server, it is better to use the **telnet** command to open a telnet session instead of using the **rlogin** command to connect to your remote server.

1. Log in as root.
2. Insert the DB2 Client Application Enablers CD-ROM in the drive.
3. If required, mount the CD-ROM. Refer to "Step 2. Mount the DB2 Client Application Enablers CD-ROM" on page 126 if required.
4. Change to the directory where the DB2 install images are available by typing the following command:
   - On AIX:

     ```
     cd /cdrom/db2/aix
     ```

   - On HP-UX:

     ```
     cd /cdrom/db2/hpux
     ```

   - On Solaris:

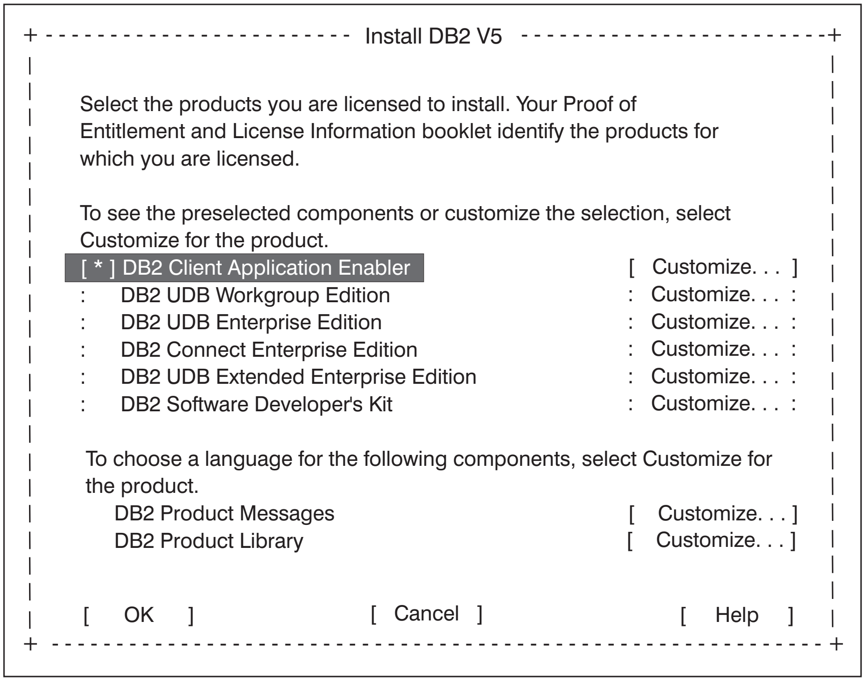
     ```
     cd /cdrom/unnamed_cdrom/db2/solaris
     ```

5. Type the following command to start the DB2 Installer program:

   ```
   ./db2setup
   ```

It will take some time for the DB2 Installer program to start up, as it is scanning your system for information.

Note: Your screen will be different if you have already installed DB2 Version 5 products on your system.



```
+ - - - - - - - - - - - - - - - - - - - - - - -  Install DB2 V5  - - - - - - - - - - - - - - - - - - - - - - -+
|                                                                                                            |
|    Select the products you are licensed to install. Your Proof of                                          |
|    Entitlement and License Information booklet identify the products for                                   |
|    which you are licensed.                                                                                 |
|                                                                                                            |
|    To see the preselected components or customize the selection, select                                    |
|    Customize for the product.                                                                              |
|   [ * ] DB2 Client Application Enabler                                 [  Customize. . .  ]                |
|    :     DB2 UDB Workgroup Edition                                     :  Customize. . .  :                |
|    :     DB2 UDB Enterprise Edition                                    :  Customize. . .  :                |
|    :     DB2 Connect Enterprise Edition                                :  Customize. . .  :                |
|    :     DB2 UDB Extended Enterprise Edition                           :  Customize. . .  :                |
|    :     DB2 Software Developer's Kit                                  :  Customize. . .  :                |
|                                                                                                            |
|    To choose a language for the following components, select Customize for                                |
|    the product.                                                                                            |
|        DB2 Product Messages                                            [   Customize. . . ]                |
|        DB2 Product Library                                             [   Customize. . . ]                |
|                                                                                                            |
|                                                                                                            |
|    [    OK    ]                     [  Cancel  ]                          [    Help    ]                   |
+ - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -+
```

6 From the product list on the Install DB2 V5 screen, select the DB2 products that you want to install.

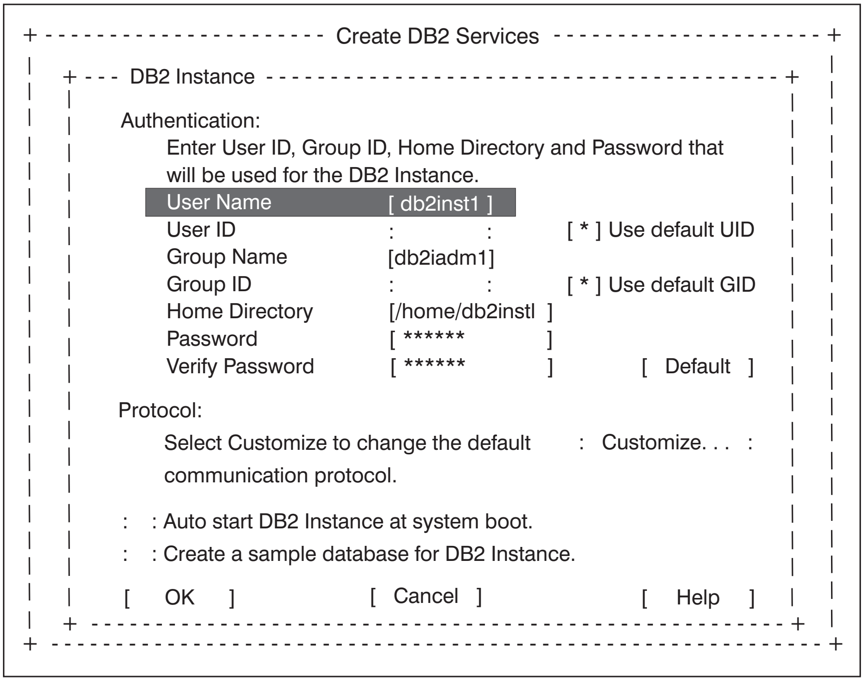
To display the required and optional components for the product you want to install, select **Customize**. The optional components that are most typically used are pre-selected for you.

To refresh the current screen, press the F5 key or Ctrl+L.

When you have finished selecting the product components to install, select **OK** to go to the Create DB2 Services screen. To undo any selections you made, select **Cancel**.

Go to step 9 on page 130 if you do not want to create a DB2 Instance at this time.

7 At the Create DB2 Services screen, select **Create a DB2 Instance** to display the DB2 Instance screen as follows:



```
+--------------------- Create DB2 Services ---------------------+
|                                                               |
|  +--- DB2 Instance ---------------------------------------+   |
|  |                                                        |   |
|  |  Authentication:                                       |   |
|  |     Enter User ID, Group ID, Home Directory and Password that |
|  |     will be used for the DB2 Instance.                 |   |
|  |     User Name           [ db2inst1 ]                   |   |
|  |     User ID             :          :   [ * ] Use default UID |
|  |     Group Name          [db2iadm1]                     |   |
|  |     Group ID            :          :   [ * ] Use default GID |
|  |     Home Directory      [/home/db2instl  ]             |   |
|  |     Password            [ ******         ]             |   |
|  |     Verify Password     [ ******         ]    [ Default ] |   |
|  |                                                        |   |
|  |  Protocol:                                             |   |
|  |     Select Customize to change the default  :  Customize. . .  : |
|  |     communication protocol.                            |   |
|  |                                                        |   |
|  |  :  : Auto start DB2 Instance at system boot.          |   |
|  |  :  : Create a sample database for DB2 Instance.       |   |
|  |                                                        |   |
|  |  [   OK   ]          [ Cancel ]          [   Help   ]  |   |
|  +--------------------------------------------------------+   |
+---------------------------------------------------------------+
```

You can use the default values displayed at the DB2 Instance screen or change them. To change the default **User ID** for the instance, deselect **Use Default UID** and enter a new value for the **User ID**.

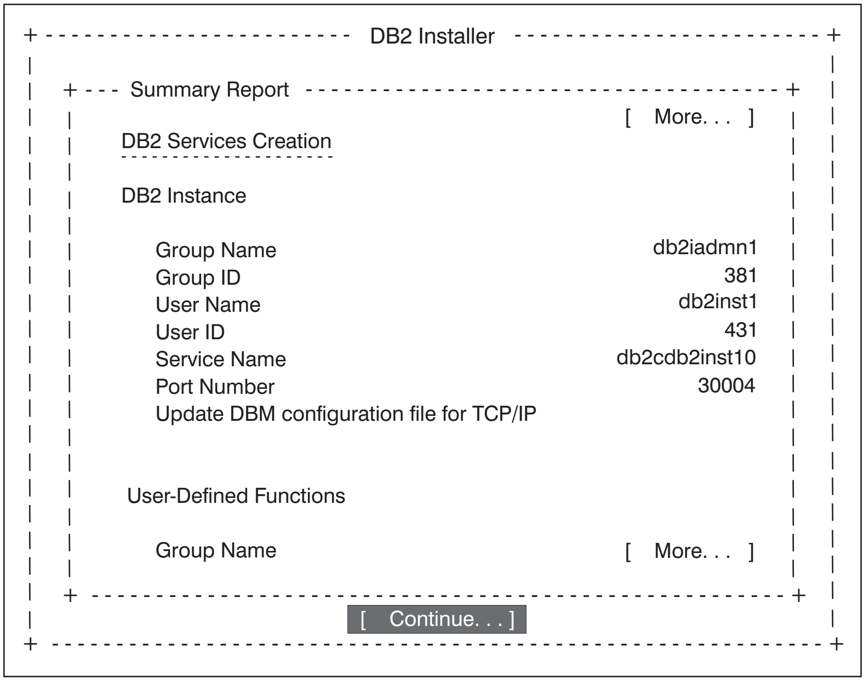
If you are running file collections on an RS/6000 SP system that is running AIX, ensure that the instance user ID is created from the file collections server. When file collections is run, the user ID and group will be deleted if added to an RS/6000 SP node that is *not* the file collections server.

To see if file collections is running, issue the **crontab -l** command from one of the RS/6000 SP nodes and look for an entry such as:

```
10 * * * * /var/sysman/supper update sup.admin
user.admin node.root
```

8 Complete the fields and select **OK** to return to the Create DB2 Services screen.

9 Select **OK** to display the Summary Report as shown in the following screen. The report lists the installable items as well as the location of the installation log file.



```
+------------------------ DB2 Installer ------------------------+
|                                                               |
|  +--- Summary Report ----------------------------------------+ |
|  |                                          [  More. . . ]  | |
|  |   DB2 Services Creation                                  | |
|  |   ---------------------                                  | |
|  |                                                          | |
|  |   DB2 Instance                                           | |
|  |                                                          | |
|  |      Group Name                             db2iadmn1    | |
|  |      Group ID                                     381    | |
|  |      User Name                               db2inst1    | |
|  |      User ID                                      431    | |
|  |      Service Name                       db2cdb2inst10    | |
|  |      Port Number                                30004    | |
|  |      Update DBM configuration file for TCP/IP            | |
|  |                                                          | |
|  |                                                          | |
|  |   User-Defined Functions                                 | |
|  |                                                          | |
|  |      Group Name                         [  More. . . ]   | |
|  |                                                          | |
|  +----------------------------------------------------------+ |
|                       [  Continue. . . ]                      |
+---------------------------------------------------------------+
```

10 Use the Up or Down arrow keys to review the Summary Report. If the selections are not correct, press the **F3** key, to return to the previous screen. Otherwise select **Continue** to start the installation.

11 When the installation is complete, use the Up or Down arrow keys to review the Status Report. Select **View Log** to view the installation log file. Select **OK** to close the Status Report screen.

12 Select **Close** from the DB2 Installer screen to terminate the DB2 Installer program.

After the installation is complete, the software is installed in the *DB2DIR* directory,

where DB2DIR	= /usr/lpp/db2_05_00	on AIX
	= /opt/IBMdb2/V5.0	on HP-UX or Solaris

Proceed either to Chapter 14, “Configuring Client-to-Server Communications Using the CCA” on page 143 or “Installing on Other UNIX Systems” on page 132.

Installing on Other UNIX Systems

To install the DB2 Client Application Enabler on SINIX, SCO OpenServer or Silicon Graphics IRIX, perform the following steps:

1. Install the DB2 products. This includes mounting the DB2 Client Application Enablers CD-ROM.
2. Update the kernel configuration parameters.
3. Create or assign groups and users.
4. Create an instance of the product.

You can also set the DB2 operating environment and optionally create links for DB2 files.

Installing on a SINIX System

On the CD-ROM, the installation image for the DB2 Client Application Enabler for SINIX is available in the */cdrom/db2/sinix/IBMdb2*directory. To install the DB2 products, you must use the **pkgadd** command.

Install the DB2 products as follows:

1. Log in as root.
2. Insert the DB2 Client Application Enablers CD-ROM into the drive.
3. Mount the CD-ROM using either the **sysadm** or the **mount** command as follows:

   ```
   mkdir /cdrom
   mount -F hs /dev/ios0/sdisk006s0 /cdrom
   ```

   where */cdrom* is the CD-ROM mount directory.

   The install images for DB2 for SINIX products are available in the */cdrom/db2/sinix/IBMdb2* directory.
4. Install the DB2 Client Application Enabler. Type the following command:

   ```
   pkgadd -d /cdrom/db2/sinix/IBMdb2 db2cliv21 db2msgDe db2conv
   ```

Some kernel configuration parameters must be changed to use DB2 Client Application Enabler for SINIX. Use the SINIX **idtune** command to update the values as given in Table 11 on page 133. After doing so, rebuild the SINIX kernel and reboot the system for the changes to take effect.

The following are the minimum recommended values:

Table 11. SINIX Kernel Configuration Parameters (Recommended Values)

SINIX RM600 Kernel Parameter	Recommended Value
MSGMAX	65535
MSGMNB	65535
MSGSEG	8192
MSGSSZ	16

Note: The kernel configuration parameters *MSGMNB* and *MSGMAX* must be changed to 65535.

Go to "Create or Assign Groups and Users" on page 135.

Installing on an SCO OpenServer System

Install the DB2 products as follows:

1 Log in as root.

2 Insert the DB2 Client Application Enablers CD-ROM into the drive.

3 Mount the CD-ROM. For example, to mount the CD-ROM as */cdrom*, type:

```
mkdir /cdrom
mount /dev/cd0 /cdrom
```

Install images for DB2 products are available in the */cdrom/db2/sco/IBMdb2* directory.

4 Issue the **custom** command. The Software Manager window opens.

5 From the Software Manager window, do the following:

a From the **Software** pulldown menu, select **Install New**.

b If you are doing a remote install, select the host from which you want to install. Otherwise, select **Continue**.

c From the **Media Device** menu, select **Media Images** and select **Continue**.

d In the **Image Directory** field, enter:

```
/cdrom/db2/sco/IBMdb2
```

e Choose the software you want to install:

- To install all components, select **Full**.

- To install selected components, choose **Partial**. On the Software Selection window, highlight the products you want to install:
  - DB2 Client Application Enabler for SCO OpenServer
  - DB2 Product Messages for SCO OpenServer.

  Press Enter.

6 Update the kernel configuration parameters.

Change the values as given in the table that follows by updating the */etc/conf/cf.d/stune* file. Then, rebuild the kernel and reboot the system for the changes to take effect.

Table 12. SCO OpenServer Kernel Configuration Parameters (Recommended Values)

Kernel Parameter	Recommended Value
MSGSSZ	64
SEMMAP	100
SEMMNI	100
SEMMNS	100
SEMMNU	100

Go to “Create or Assign Groups and Users” on page 135.

Installing on a Silicon Graphics IRIX System

On the CD-ROM, the installation image for Silicon Graphics IRIX products is contained in the */cdrom/db2/sgi/IBMdb2* directory.

Install the DB2 software as follows:

1 Log in as root.

2 Insert the DB2 Client Application Enablers CD-ROM into the drive.

3 Mount the CD-ROM using the **mount** command as follows:

```
mount -t iso9660 device mount_point
```

For example, to mount the CD-ROM as `/cdrom`, enter the following commands:

```
mkdir /cdrom
mount -t iso9660  /dev/scsi/sc0d7l0 /cdrom
```

4 Install the DB2 Client Application Enabler. Issue the following command:

```
pkgadd -d /cdrom/db2/sgi/IBMdb2 db2cliv21 db2conv
```

Go to “Create or Assign Groups and Users” on page 135.

Create or Assign Groups and Users

If you do not already have a user ID to own the instance and a system administration group ID that is the primary group of the instance owner, create these as follows:

1 Log in as `root`.

2 Create a system administration group ID. Use these commands on the following operating systems:

SINIX, SGI	Use **sysadm** or **groupadd**
SCO OpenServer	Use **scoadmin** or **groupadd**

For example, to create a group named dbadmin1 on SCO OpenServer, type:

```
groupadd dbadmin1
```

Use the appropriate command for the other operating systems when you create a group.

3 Create a user ID that will be the instance owner.

SINIX, SGI	Use **sysadm** or **useradd**
SCO OpenServer	Use **scoadmin** or **useradd**

For example, to create a user named inst1 on SCO OpenServer, type:

```
useradd -g dbadmin1 -G dbadmin1 -d /home/inst1 inst1
passwd inst1
```

Use the appropriate command for the other operating systems when you create a user.

- When choosing a name for a group or user, follow the rules described in Appendix F, “Naming Rules” on page 315.
- Dedicate the instance-owner user ID to that instance's use only. This allows for easier error recovery if a system error occurs.

Create an Instance of the Product

Use the **db2icrt** command to create an instance of the product. The location of the **db2icrt** command depends on your operating system:

- On SCO OpenServer, go to the */usr/IBMdb2/V2.1/instance* directory.
- On SINIX and Silicon Graphics IRIX operating systems, go to the */opt/IBMdb2/V2.1/instance* directory.

The syntax of the **db2icrt** command is:

```
►►──db2icrt──── InstName ──────[-a AuthType]────[-u FencedID] ──────►◄
```

where:

InstName Is the login name of the instance owner.

-a *AuthType* Is an optional parameter that specifies the authentication type for the instance. Valid authentication types are SERVER, CLIENT, and DCS. If the -a parameter is not specified, the authentication type defaults to SERVER, if a server product is installed. Otherwise, the *AuthType* is set to CLIENT.

Note: The authentication type of the instance applies to all databases under the instance.

-u *FencedID* Is the user under which the fenced UDFs and stored procedures will execute. This is not required if you install the DB2 Client Application Enabler or the DB2 Software Developer's Kit. For other products, this is an required parameter.

Note: *FencedID* may not be root or bin.

To create an instance for the DB2 client, you can use the following command:

```
db2icrt db2inst1
```

The **db2icrt** command creates the *INSTHOME/sqllib* directory, where *INSTHOME* is the home directory of the instance owner.

> To avoid a potential loss of data if an instance is deleted, you should not create user files or directories under the *INSTHOME/sqllib* directory, other than those created by DB2. The only exceptions is if your system supports fenced user defined functions and fenced stored procedures, put the fenced applications in the *INSTHOME/sqllib/function* directory.

Set Up the DB2 Operating Environment

You must execute a script to set the DB2 operating environment and select an instance before starting DB2. On AIX, HP-UX, and Solaris, see Chapter 21, “Controlling Your DB2 Environment” on page 213 for more information on how to do this. For other UNIX operating systems, the sample scripts, *db2profile* (for Bourne or Korn shell) and *db2cshrc* (for C shell) are provided to help you set up the operating environment for a DB2 instance.

The instance owner may customize these scripts for all users of an instance. These scripts are available in the *INSTHOME/sqllib* directory, where *INSTHOME* is the home directory of the instance.

Determine the most appropriate way to execute the commands in the sample *db2profile* script in your environment and instruct your users to do it; for example you could add the following to the startup script:

```
. INSTHOME/sqllib/db2profile    (for Bourne or Korn shell)
source INSTHOME/sqllib/db2cshrc (for C shell)
```

Create Links for DB2 Files

If you are developing or running applications, you may want to create links for DB2 libraries to avoid needing to specify the full path to the product libraries and the include files. To create these links, use the **db2ln** command.

- On SCO OpenServer systems, type:

  ```
  /usr/IBMdb2/V2.1/cfg/db2ln
  ```

- On SINIX and Silicon Graphics IRIX systems, type:

  ```
  /opt/IBMdb2/V2.1/cfg/db2ln
  ```

This creates links from the DB2 libraries to */usr/lib*, and from the DB2 include files to */usr/include*.

To configure your client to access remote servers, go to Chapter 15, “Configuring Client-to-Server Communications Using the Command Line Processor” on page 147.

Chapter 13. Installing DB2 Clients on Macintosh Workstations

To install the DB2 Client Application Enabler on a Macintosh workstation, perform the following steps:

1 Insert the DB2 Client Application Enablers CD-ROM into the drive.

2 Double-click on the **DB2 CLIENTS** icon.

3 Double-click on the **MAC** folder.

4 Double-click on the *language* folder, where *language* is the two-character country code that represents your language (for example, *EN* for English). Table 28 on page 314 lists the code for each available language.

5 Double-click on the **CLIENT** folder.

6 Double-click on the **CAEMAC** icon to start the installation. The Welcome window opens.

7 Follow the instructions provided to complete the installation.

8 When the install program is finished, you will need to restart your computer to complete the installation. Click on the **Restart** push button to restart your computer and complete the installation.

To configure your client to access a remote DB2 server, see Chapter 15, "Configuring Client-to-Server Communications Using the Command Line Processor" on page 147.

Part 4. Configuring DB2 Communications

Chapter 14. Configuring Client-to-Server Communications Using the CCA

The information in this section describes how to use the Client Configuration Assistant to configure OS/2, Windows 95, and Windows NT workstations to access remote DB2 Universal Database servers.

To configure all other clients, see Chapter 15, “Configuring Client-to-Server Communications Using the Command Line Processor” on page 147.

Use the Client Configuration Assistant to configure your OS/2, Windows 95, or Windows NT client workstation to access remote DB2 servers. The Client Configuration Assistant provides three configuration methods: one makes use of a server's access profile, one searches the network for databases, and one requires that you enter the database name and the communication protocol values of the DB2 server.

Perform the following steps to configure your workstation to access remote servers:

1 Start the Client Configuration Assistant (CCA).

For OS/2:
Double-click on the **Client Configuration Assistant** icon, located in the **DB2 for OS/2** folder.

For Windows 95 or Windows NT:
Click on **Start** and select **Programs**->**DB2 for Windows**->**Client Configuration Assistant**.

The Welcome panel opens each time you start the CCA, until you have added at least one database to your client.

2 Click on the **Add Database** or **Add** push button to configure connections using the Add Database SmartGuide. Choose one of the following configuration methods:

- If your administrator provided you with a file containing database access information, select the **Use an access profile** radio button and click on the **Next** push button.

  a Click on the **Browse** push button to select an access profile.

  b You are presented with a list of systems, instances, and databases. Select the database that you want to use and proceed to Step 3.

- If you want to search the network for databases, select the **Search the network** radio button and click on the **Next** push button.

  Note: Discovery cannot find information about pre-Version 5 DB2 systems.

The **Other Systems (Search the network)** icon will only appear if the client's *discover* parameter is set to SEARCH.

a Click on the **[+]** sign beside the **Known Systems** icon to list all the systems known to your client.

b Click on the **[+]** sign beside the system to get a list of the instances and databases on it. Select the database that you want to add and proceed to Step 3.

c If the system that contains the database that you want is not listed, click on the **[+]** sign beside the **Other Systems (Search the network)** icon to search the network for additional systems. Click on the **[+]** sign beside the system to get a list of the instances and databases on it. Select the database that you want to add and proceed to Step 3.

If an Administration Server that recognizes the remote instance is not running, or if the *db2comm* registry value is not set to support the protocol that you are using, the Clien t Configuration Assistant will be unable to detect the remote system.

d If the system you want is still not listed, it can be added to the list of systems by clicking on the **Add Systems** push button. Enter the required communication protocol parameters and click on the **Next** push button. For more information, click on the **Help** push button.

Select the database that you want to add and proceed to Step 3.

- If you have the protocol information for the database that you want to connect to, select the **Manually configure a connection to a DB2 database** radio button and click on the **Next** push button.

a Select the radio button that corresponds to the protocol that you want to use from the **Protocol** list. Click on the **Next** push button.

b Enter the required communication protocol parameters and click on the **Next** push button. For more information, click on the **Help** push button.

c Specify the name of the database that you want to connect to in the **Target database** field and proceed to Step 3.

3 To specify the database alias name or to add a description, click on the **Next** push button. If you do not specify a database alias name, the default will be the same as the database name.

4 If you want to run ODBC-enabled applications, click on the **Next** push button. If you do not plan to use ODBC, click on the **Done** push button to finish using the Add Database SmartGuide and proceed to Step 6.

5. If you are using ODBC applications, select the **Register this database for ODBC** check box, select the radio button that describes the type of data source that you want to register this database as, and select the application that you want to use from the **Application** drop down box. Click on the **Done** push button to finish using the Add Database SmartGuide.

6. The Confirmation window opens. Click on the **Test Connection** push button to test the connection to the specified database.

7. Enter your user ID and password to access the database and click on **OK**. If the connection is successful, a message confirming the connection appears. If the connection fails, click on the **Help** push button for more information. If you require additional information, refer to the *Troubleshooting Guide*.

8. You are now able to use the database. If you want to access another database, select the **Add Another** push button. To finish using the Add Database SmartGuide, click on the **Close** push button.

Verifying the Connection

When the configuration of the server and client is complete, use the following steps to verify that you can access data from a remote database:

> You will need to connect to a remote database to test the connection. If you do not have a database on the server, create the SAMPLE database on the server to test the connection. See “Creating the Sample Database” on page 168 for more information.

1. Start the database manager by issuing the **db2start** command on the server (if it was not automatically started at boot time).

2. Issue the following command in the client's Command Center or command line processor to connect the client to the remote database:

```
connect to database_alias user userid using password
```

The database alias (*database_alias*) is an arbitrary local nickname for the remote database on the client. If you do not provide one, the default is the same as the database name (*database_name*). This is the name that you use when connecting to a database from a client.

The values for *userid* and *password* must be valid for the system on which they are authenticated. By default, authentication takes place on the SERVER. If the database manager is configured for CLIENT authentication, the *userid* and *password* must be valid on the client.

If the connection is successful, you will get a message showing the name of the database to which you have connected. You are now able to retrieve data from that database. For example, to retrieve a list of all the table names listed in the system catalog table, enter the following SQL command in the Command Center or command line processor:

```
"select tabname from syscat.tables"
```

When you are finished using the database connection, enter the **connect reset** command to end the database connection.

You are ready to start using the DB2 client to communicate with a DB2 server. See Chapter 16, “Getting Started with DB2 Universal Database” on page 167 for details.

Chapter 15. Configuring Client-to-Server Communications Using the Command Line Processor

This section describes how to use commands to configure a client workstation to communicate with a DB2 server using the command line processor.

> If you are planning to use an OS/2, Windows 95, or Windows NT client for communications, the Client Configuration Assistant (CCA) makes it easy to automate the tasks of configuring and administering clients to communicate with servers. If you have installed the CCA, it is recommended that you use this utility to configure your clients for communications. See Chapter 14, "Configuring Client-to-Server Communications Using the CCA" on page 143 for more information.

For instructions on using the command line processor, see Chapter 20, "Entering DB2 Commands and SQL Statements" on page 207. For a description of database naming rules, see Appendix F, "Naming Rules" on page 315.

Go to the communication protocol that you want your client to use to access a remote server.

- TCP/IP - see "Configuring TCP/IP on the Client"
- IPX/SPX - see "Configuring IPX/SPX on the Client" on page 156
- APPC - refer to the *Installing and Configuring DB2 Clients* online manual for more information.

Configuring TCP/IP on the Client

This section assumes that TCP/IP is functional on the client and DB2 server workstations. See "Software Requirements" on page 41 for the communication protocol requirements for your platform. See "Possible Client-to-DB2 Server Connectivity Scenarios" on page 47 for the supported communication protocols for your particular client and server.

The following steps are required to set up a DB2 client to use TCP/IP communications:

1. Identify and record parameter values.
2. Configure the client:
   - **a** Resolve the server's host address.
   - **b** Update the *services* file.
   - **c** Catalog the TCP/IP node.
   - **d** Catalog the database.
3. Test the connection between the client and server.

> Due to the characteristics of the TCP/IP protocol, the TCP/IP subsystem may not be immediately notified of the failure of a partner on another host. As a result, a client application accessing a remote DB2 server using TCP/IP, or the corresponding agent at the server, may sometimes appear to be hung. DB2 uses the TCP/IP SO_KEEPALIVE socket option to detect when there has been a failure and the TCP/IP connection has been broken.
>
> If you are experiencing problems with your TCP/IP connection, refer to the *Troubleshooting Guide* for information on this parameter and other common TCP/IP problems.

Step 1. Identify and Record Parameter Values

As you proceed through the configuration steps, complete the *Your Value* column in the following table. You can fill in some of the values before you start configuring this protocol.

Table 13 (Page 1 of 2). TCP/IP Values Required at the Client

Parameter	Description	Sample Value	Your Value

Hostname (*hostname)*
or
IP address (*ip_address*)
	Use the *hostname* or *ip_address* of the remote server workstation.
	To resolve this parameter:
	• Issue the **hostname** command at the server to obtain the *hostname* parameter.
	• Issue the **ping** *server_hostname* command to obtain the *ip_address* parameter.
		`serverhost`
		or
		`9.21.15.235`	

Table 13 (Page 2 of 2). TCP/IP Values Required at the Client

Parameter	Description	Sample Value	Your Value

Connection Port
- Connection Service name (*svcename*)
- Port number/Protocol (port_number/tcp)
	Values required in the *services* file.
	The Connection Service name is an arbitrary name used to represent the port number (*port_number*) on the client.
	The port number for the client must be the same as the port number that the *svcename* parameter maps to in the *services* file at the server. (The *svcename* parameter is located in the database manager configuration file on the DB2 server.) This value must not be in use by any another applications, and must be unique within the *services* file. Ensure that you do not specify a port number that is being used by the Fast Communication Manager (FCM).
		`server1`
		`3700/tcp`	

Node name (*node_name*)	A local alias, or nickname, that describes the node where the database resides. You can choose any name you want, however, all node name values within your local node directory must be unique.	`db2node`	

Step 2. Configure the Client

The following steps describe how to configure this protocol on the client. Replace the sample values with your worksheet values.

A. Resolve the Server's Host Address

If your network has a name server, or you are planning to directly specify the IP address (*ip_address*) of the server, skip this step and proceed to “Step B. Update the Services File” on page 151.

The client workstation must know the address of the server to which it is attempting to establish communications. If a name server does not exist on your network, you may directly specify a hostname that maps to the IP address (*ip_address*) of the host in the local *hosts* file. See Table 14 on page 150 for the location of the *hosts* file for your particular platform.

If you are planning on supporting a UNIX client that is using Network Information Services (NIS), and you are not using a name server on your network, you must update the *hosts* file located on your NIS master server.

Table 14. Location of the Local Hosts and Services Files

Platform	Location
Macintosh	The *hosts* file is located in the folder called **System Folder**. **Note:** This operating system does not use a *services* file. You must catalog this node using the *Port_number* parameter. See "Step C. Catalog the TCP/IP Node" on page 151 for more information.
OS/2	Specified by the *etc* environment variable. Issue the **set etc** command to determine the location of your local *hosts* or *services* files. **Note:** For DOS and WIN-OS2 sessions, you might need to update the *hosts* and *services* files located in the *tcpip_product\dos\etc* directory.
Windows 3.x	Typically in the *tcpip_product\etc* directory, but it depends on the products that you have installed. Refer to your TCP/IP documentation for more information.
Windows 95	*windows* directory
Windows NT	*winnt\system32\drivers\etc* directory
UNIX	*/etc* directory

Using a local text editor, add an entry to the client's *hosts* file for the server's hostname. For example:

```
9.21.15.235     serverhost   # host address for serverhost
```

where:

9.21.15.235 is the *ip_address*

serverhost is the *hostname*

# is a comment describing the entry

Notes:

1. If the server is not in the same domain as the client, you must provide a fully qualified domain name such as `serverhost.vnet.ibm.com`, where `vnet.ibm.com` is the domain name.
2. For specific information on resolving host addresses, refer to your TCP/IP documentation.

Step B. Update the Services File

If you are planning to catalog the TCP/IP node using the port number (*port_number*), skip this step and go to “Step C. Catalog the TCP/IP Node.”

If you are configuring a DB2 for Macintosh client, you must catalog the TCP/IP node using the port number (*port_number*). Go to “Step C. Catalog the TCP/IP Node” to catalog the node.

Using a local text editor, add the Connection Service name and port number to the client's *services* file for TCP/IP support. For example:

```
server1  3700/tcp  # DB2 connection service port
```

where:

server1 is the Connection Service name

3700 is the port number for the Connection Port

`tcp` is the communication protocol that you are using

The port number used on the client must match the port number used on the server.

If you are planning on supporting a UNIX client that is using Network Information Services (NIS), you must update the *services* file located on your NIS master server.

The file called *services* is located in the same directory as the local *hosts* file that you may have edited in “A. Resolve the Server's Host Address” on page 149.

Ensure that you did not specify a port number that is being used by the Fast Communication Manager (FCM) or any other process.

See Table 14 on page 150 for the location of the *services* file for your particular platform.

Step C. Catalog the TCP/IP Node

You must add an entry to the client's node directory to describe the remote node.

This entry specifies the chosen alias (*node_name*), the *hostname* (or *ip_address*), and the *svcename* (or *port_number*) that the client will use to access the remote server.

To catalog the TCP/IP node, perform the following steps:

1 Log on to the system as a user with System Administrative (SYSADM) or System Controller (SYSCTRL) authority.

If you have trouble logging on to the system, see “Logging on to the System” on page 167.

If you are configuring communications for a Windows 3.x client, you can use the Client Setup tool to complete this task. Go to “Using the Client Setup Tool to Configure a Windows 3.x Client” on page 163 to catalog a node using the Client Setup tool.

2 If you are using a UNIX client, set up the instance environment and invoke the DB2 command line processor as follows:

a. Run the start up script as follows:

```
. INSTHOME/sqllib/db2profile    (for Bourne or Korn shell)
source INSTHOME/sqllib/db2cshrc (for C shell)
```

where INSTHOME is the home directory of the instance.

b. Start the DB2 command line processor by issuing the **db2** command.

3 Catalog the node by issuing the following commands in the command line processor:

```
catalog tcpip node node_name remote [  hostname | ip_address ] server [ svcename | port_number ]
terminate
```

For example, to catalog the remote server *serverhost* on the node called *db2node*, using the service name *server1*, use:

```
catalog tcpip node db2node remote serverhost server server1
terminate
```

To catalog a remote server with the IP address *9.21.15.235* on the node called *db2node*, using the Port number *3700*, use:

```
catalog tcpip node db2node remote 9.21.15.235 server 3700
terminate
```

If you need to change values that were set with the **catalog node** command, first run the **uncatalog node** command in the command line processor as follows:

```
uncatalog node node_name
```

Recatalog the node with the value that you want to use.

Step D. Catalog the Database

Before a client application can access a remote database, the database must be cataloged on the server node and on any client nodes that will connect to it. When you create a database, it is automatically cataloged on the server with the database alias (*database_alias*) the same as the database name (*database_name*). The information in the database directory, along with the information in the node directory, is used on the client to establish a connection to the remote database.

To catalog a database on the client, perform the following steps.

1 Log on to the system as a user with System Administrative (SYSADM) or System Controller (SYSCTRL) authority.

> If you have trouble logging on to the system, see “Logging on to the System” on page 167.

2 Fill in the *Your Value* column in the following worksheet.

Table 15. Parameter Values for Cataloging Databases

Parameter	**Description**	**Sample Value**	**Your Value**
Database name (*database_name*)	The database alias (*database_alias*) of the *remote* database. When you create a database, it is automatically cataloged on the server with the database alias name (*database_alias*) the same as the database name (*database_name*).	`sample`	
Database alias (*database_alias*)	An arbitrary local nickname for the remote database, on the client. If you do not provide one, the default is the same as the database name (*database_name*). This is the name that you use when connecting to a database from a client.	`tor1`	
Node name (node_name)	The name of the node directory entry that describes where the database resides. Use the same value for node name (*node_name*) that you used to catalog the node in the previous step.	`db2node`	

> If you are configuring communications for a Windows 3.x client, you can use the Client Setup tool to complete this task. Go to “Using the Client Setup Tool to Configure a Windows 3.x Client” on page 163 to catalog a database using the Client Setup tool.

3 If you are using a UNIX client, set up the instance environment and invoke the DB2 command line processor as follows:

a. Run the start up script as follows:

```
. INSTHOME/sqllib/db2profile    (for Bourne or Korn shell)
source INSTHOME/sqllib/db2cshrc (for C shell)
```

where INSTHOME is the home directory of the instance.

b. Start the DB2 command line processor by issuing the **db2** command.

4 Catalog the database by issuing the following commands in the command line processor:

```
catalog database database_name as database_alias at node node_name
terminate
```

For example, to catalog a remote database called *sample* so that it has the alias *tor1*, on the node *db2node*, use:

```
catalog database sample as tor1 at node db2node
terminate
```

If you need to change values that were set with the **catalog** *database* command, first run the **uncatalog** *database* command in the command line processor as follows:

```
uncatalog database database_alias
```

Recatalog the database with the value that you want to use.

Step 3. Test the Client-to-Server Connection

When the configuration of the server and client is complete, use the following steps to verify that you can access data from a remote database:

You will need to connect to a remote database to test the connection. If you do not have a database on the server, create the sample database on the server to test the connection. See “Creating the Sample Database” on page 168 for more information.

1 Start the database manager by issuing the **db2start** command on the server (if it was not automatically started at boot time).

2 Enter the following command in the client's Command Center or command line processor to connect the client to the remote database:

```
connect to database_alias user userid using password
```

The values for *userid* and *password* must be valid for the system on which they are authenticated. By default, authentication takes place on the SERVER. If the database manager is configured for CLIENT authentication, the *userid* and *password* must be valid on the client.

If the connection is successful, you will get a message showing the name of the database to which you have connected. You are now able to retrieve data from that database. For example, to retrieve a list of all the table names listed in the system catalog table, enter the following SQL command in the Command Center or command line processor:

```
"select tabname from syscat.tables"
```

When you are finished using the database connection, enter the **connect reset** command to end the database connection.

You are ready to start using the DB2 server. See Chapter 16, “Getting Started with DB2 Universal Database” on page 167 for details.

Troubleshooting the Client-to-Server Connection

If the connection fails, check the following items:

At the *server*:

1 The *db2comm* registry value includes the value `tcpip`. DB2COMM can be set in the *db2profile* or *db2cshrc* file.

Check the settings for the *db2comm* registry value by issuing the **db2set DB2COMM** command. For more information, see Chapter 21, “Controlling Your DB2 Environment” on page 213.

2 The *services* file was updated correctly.

3 The service name (*svcename*) parameter was updated correctly in the database manager configuration file.

4 The database was created and cataloged properly.

5 The database manager was stopped and started (enter the **db2stop** and **db2start** commands on the server).

6 Ensure that you did not specify a port number that is being used by the Fast Communication Manager (FCM) or any other process.

If there are problems starting a protocol's connection managers, a warning message is displayed and the error messages are logged in the *db2diag.log* file, located in the *instance* directory.

Additionally, any failure encountered while executing the **db2start** command is logged to a time-stamped file in the log subdirectory within the instance directory. This file contains the **db2start** results from each database partition server in the instance.

If there is no error when the **db2start** command is run, these files are removed.

Refer to the *Troubleshooting Guide* for information on the *db2diag.log* file.

At the *client*:

1. If used, the *services* and *hosts* files were updated correctly.
2. The node was cataloged with the correct hostname (*hostname*) or IP address (*ip_address)*.
3. The port number must match, or the service name must map to, the port number used on the server.
4. The node name (*node_name*), specified in the database directory, points to the correct entry in the node directory.
5. The database was cataloged properly, using the *server's* database alias (*database_alias*) that was cataloged when the database was created on the server, as the database name (*database_name*) on the client.

After you verify these items, refer to the *Troubleshooting Guide* if the connection still fails.

Configuring IPX/SPX on the Client

This section assumes that IPX/SPX is functional on the client and DB2 server workstations. See “Software Requirements” on page 41 for the communication protocol requirements for your platform. See “Possible Client-to-DB2 Server Connectivity Scenarios” on page 47 for the supported communication protocols for your particular client and server.

A client can access a DB2 Extended Enterprise Edition server via Direct Addressing only. For a description of Direct Addressing, see “Configuring IPX/SPX on the Server” on page 237.

The following steps are required to set up a DB2 client to use IPX/SPX communications:

1. Identify and record parameter values.
2. Configure the client:
   a. Catalog the IPX/SPX node.
   b. Catalog the database.
3. Test the connection between the client and server.

Step 1. Identify and Record Parameter Values

As you proceed through the configuration steps, complete the *Your Value* column in the following table. You can fill in some of the values before you start configuring this protocol.

Table 16. IPX/SPX Values Required at the Client

Parameter	Description	Sample Value	Your Value
File server name (*FILESERVER*)	**Direct Addressing** A *** value indicates that you are using Direct Addressing. Locate this parameter in the database manager configuration file on the server.	**Direct Addressing** *	
DB2 server object name (*OBJECTNAME*)	**Direct Addressing** The server's IPX/SPX internetwork address of the form: *netid (8 byte).nodeid (12 byte). socket # (4 byte)* To resolve this parameter, issue the **db2ipxad** command at the server. See "A. Catalog the IPX/SPX Node" on page 158 for more information. Locate this parameter in the database manager configuration file on the server.	**Direct Addressing** `09212700.400011527745.879E`	
Node name (*node_name*)	A local alias, or nickname, that describes the node where the database resides. You can choose any name you want, however, all node name values within your local node directory must be unique.	`db2node`	

Step 2. Configure the Client

The following steps describe how to configure this protocol on the client. Replace the sample values with your worksheet values.

A. Catalog the IPX/SPX Node

You must add an entry to the client's node directory to describe the remote node.

This entry specifies the chosen alias (*node_name*), the file server name (*FILESERVER*), and the DB2 server object name (*OBJECTNAME*) that the client will use to access the remote DB2 server.

To catalog the IPX/SPX node, perform the following steps:

1 Log on to the system as a user with System Administrative (SYSADM) or System Controller (SYSCTRL) authority.

If you have trouble logging on to the system, see “Logging on to the System” on page 167.

If you are configuring communications for a Windows 3.x client, you can use the Client Setup tool to complete this task. Go to “Using the Client Setup Tool to Configure a Windows 3.x Client” on page 163 to catalog a node using the Client Setup tool.

2 If you are using a UNIX client, set up the instance environment and invoke the DB2 command line processor as follows:

a. Run the start up script as follows:

```
. INSTHOME/sqllib/db2profile    (for Bourne or Korn shell)
source INSTHOME/sqllib/db2cshrc (for C shell)
```

where INSTHOME is the home directory of the instance.

b. Start the DB2 command line processor by issuing the **db2** command.

3 Catalog the node by issuing the following commands in the command line processor:

```
catalog ipxspx node node_name remote FILESERVER server OBJECTNAME
terminate
```

Direct Addressing Example

You must assign a *** to the *FILESERVER* parameter and specify the server's IPX/SPX internetwork address value as the *OBJECTNAME* parameter on the client.

To determine the value for the *OBJECTNAME* parameter, issue the **db2ipxad** command on the server. (This command is located in the *sqllib/misc/* directory.)

Make note of the output that is generated, and use that value in place of the sample value (`09212700.400011527745.879E`) in the example that follows.

To catalog a remote node called *db2node*, using the IPX/SPX internetwork address *09212700.400011527745.879E* as the *OBJECTNAME*, use:

```
catalog ipxspx node db2node remote * server 09212700.400011527745.879E
terminate
```

If you need to change values that were set with the **catalog node** command, first run the **uncatalog node** command in the command line processor as follows:

```
uncatalog node node_name
```

Recatalog the node with the value that you want to use.

B. Catalog the Database

Before a client application can access a remote database, the database must be cataloged on the server node and on any client nodes that will connect to it. When you create a database, it is automatically cataloged on the server with the database alias (*database_alias*) the same as the database name (*database_name*). The information in the database directory, along with the information in the node directory, is used on the client to establish a connection to the remote database.

To catalog a database on the client, perform the following steps.

1 Log on to the system as a user with System Administrative (SYSADM) or System Controller (SYSCTRL) authority.

If you have trouble logging on to the system, see "Logging on to the System" on page 167.

2 Fill in the *Your Value* column in the following worksheet.

Table 17. Parameter Values for Cataloging Databases

Parameter	Description	Sample Value	Your Value
Database name (*database_name*)	The database alias (*database_alias*) of the *remote* database. When you create a database, it is automatically cataloged on the server with the database alias name (*database_alias*) the same as the database name (*database_name*).	`sample`	
Database alias (*database_alias*)	An arbitrary local nickname for the remote database, on the client. If you do not provide one, the default is the same as the database name (*database_name*). This is the name that you use when connecting to a database from a client.	`tor1`	
Node name (node_name)	The name of the node directory entry that describes where the database resides. Use the same value for node name (*node_name*) that you used to catalog the node in the previous step.	`db2node`	

If you are configuring communications for a Windows 3.x client, you can use the Client Setup tool to complete this task. Go to "Using the Client Setup Tool to Configure a Windows 3.x Client" on page 163 to catalog a database using the Client Setup tool.

3 If you are using a UNIX client, set up the instance environment and invoke the DB2 command line processor as follows:

a. Run the start up script as follows:

```
. INSTHOME/sqllib/db2profile    (for Bourne or Korn shell)
source INSTHOME/sqllib/db2cshrc (for C shell)
```

where INSTHOME is the home directory of the instance.

b. Start the DB2 command line processor by issuing the **db2** command.

4 Catalog the database by issuing the following commands in the command line processor:

```
catalog database database_name as database_alias at node node_name
terminate
```

For example, to catalog a remote database called *sample* so that it has the alias *tor1*, on the node *db2node*, use:

```
catalog database sample as tor1 at node db2node
terminate
```

If you need to change values that were set with the **catalog** *database* command, first run the **uncatalog** *database* command in the command line processor as follows:

```
uncatalog database database_alias
```

Recatalog the database with the value that you want to use.

Step 3. Test the Client-to-Server Connection

When the configuration of the server and client is complete, use the following steps to verify that you can access data from a remote database:

You will need to connect to a remote database to test the connection. If you do not have a database on the server, create the sample database on the server to test the connection. See “Creating the Sample Database” on page 168 for more information.

1 Start the database manager by issuing the **db2start** command on the server (if it was not automatically started at boot time).

2 Enter the following command in the client's Command Center or command line processor to connect the client to the remote database:

```
connect to database_alias user userid using password
```

The values for *userid* and *password* must be valid for the system on which they are authenticated. By default, authentication takes place on the SERVER. If the database manager is configured for CLIENT authentication, the *userid* and *password* must be valid on the client.

If the connection is successful, you will get a message showing the name of the database to which you have connected. You are now able to retrieve data from that database. For example, to retrieve a list of all the table names listed in the system catalog table, enter the following SQL command in the Command Center or command line processor:

```
"select tabname from syscat.tables"
```

When you are finished using the database connection, enter the **connect reset** command to end the database connection.

You are ready to start using the DB2 server. See Chapter 16, “Getting Started with DB2 Universal Database” on page 167 for details.

Troubleshooting the Client-to-Server Connection

If the connection fails, check the following items:

At the *server*:

1. The *db2comm* registry value includes the value `ipxspx`. *db2comm* can be set in the *db2profile* or the *db2cshrc* file.

> Check the settings for the *db2comm* registry value by issuing the **db2set DB2COMM** command. For more information, see Chapter 21, “Controlling Your DB2 Environment” on page 213.

2. The *FILESERVER*, *OBJECTNAME*, and *IPX_SOCKET* parameters were updated correctly in the database manager configuration file.
3. The database was created and cataloged properly.
4. The database manager was stopped and started (enter the **db2stop** and **db2start** commands on the server).

> If there are problems starting a protocol's connection managers, a warning message is displayed and the error messages are logged in the *db2diag.log* file, located in the *instance* directory.
>
> Additionally, any failure encountered while executing the **db2start** command is logged to a time-stamped file in the log subdirectory within the instance directory. This file contains the **db2start** results from each database partition server in the instance.
>
> If there is no error when the **db2start** command is run, these files are removed.
>
> Refer to the *Troubleshooting Guide* for information on the *db2diag.log* file.

At the *client*:

1. Check that the node was cataloged with a value of * for *FILESERVER*, and the correct IPX/SPX internetwork address value for the *OBJECTNAME* parameter.
2. The node name (*node_name*), specified in the database directory, points to the correct entry in the node directory.
3. The database was cataloged properly, using the *server's* database alias (*database_alias*) that was cataloged when the database was created on the server, as the database name (*database_name*) on the client.

After you verify these items, refer to the *Troubleshooting Guide* if the connection still fails.

Configuring APPC on the Client

This section assumes that APPC is functional on the client and server workstations. See “Software Requirements” on page 41 for the communication protocol requirements for your platform. See “Possible Client-to-DB2 Server Connectivity Scenarios” on page 47 for the supported communication protocols for your particular client and server.

Refer to the online *Installing and Configuring DB2 Clients* manual for information on how to configure clients to communicate with a DB2 server using the APPC protocol.

Using the Client Setup Tool to Configure a Windows 3.x Client

The Client Setup tool provides you with a graphical interface that you can use to complete the catalog node and database instructions in Chapter 15, “Configuring Client-to-Server Communications Using the Command Line Processor” on page 147. This section describes how to use the Client Setup tool to catalog a node or database for a Windows 3.x client.

Use the instructions in this section, along with the protocol-specific instructions in Chapter 15, “Configuring Client-to-Server Communications Using the Command Line Processor” on page 147, to configure your Windows 3.x client for communications with a DB2 server.

To invoke the Client Setup tool, double-click on the **Client Setup** icon, located in the DB2 product folder for your particular platform.

Catalog the Node

1. Select **Node**->**NewNode**, the New Node window opens.
2. Fill in the node name (*node_name*) parameter in the **Node** field and optionally add a comment describing the node in the **Comment** field.
3. Select the **Protocol** radio button for the protocol that you want the Windows 3.x client to use to communicate with the server.
4. Fill in the required parameters in the **Protocol Settings** box to catalog the node for the protocol that you selected.

5. Click on **OK**.

Catalog the Database

1. Click on the **Databases** push button, the DB2 Client Setup - Databases window opens.

2. Select **Database**->**New**, the New Database window opens.

3. Fill in the **Name** (*database_name*) and **Alias** (*database_alias*) parameters and optionally add a comment in the **Comment** field describing the database.

4. Select the **Location** radio button that describes the location of the database that you are trying to connect to.

   - If the database resides on a remote DB2 server, select the **Other Node** radio button and select the name of the Node that represents the remote server from the **At Node** drop down box.

   - If the database resides locally, select the **This node** radio button and select the drive where the database is located from the **On** drop down box.

5. Select the **Authentication Type (DB2 1.x Database)** check box if you are configuring communications to a DB2 Version 1.x database.

> After you have cataloged the node and database for the Windows 3.x client, and completed the other specific steps for the protocol that you are using in Chapter 15, “Configuring Client-to-Server Communications Using the Command Line Processor” on page 147, you can test the database connection by clicking on the **Test Database Connection** push button.
>
> If the connection is not successful, see the "Troubleshooting the Client to DB2 Server Connection" section for the protocol that you trying to configure.

Part 5. Using DB2 Universal Database

Chapter 16. Getting Started with DB2 Universal Database

This section describes how to use DB2 Universal Database in a variety of ways.

Go to the section that gives instructions for the tasks that you want to perform.

- "Logging on to the System."
- "Creating the Sample Database" on page 168.
- "Connecting to a Database" on page 169.
- "Working with the System Administrative Group" on page 170.
- "Upgrading from a Try-and-Buy License" on page 170.

Logging on to the System

To use DB2, log on with a valid user ID that has the appropriate authority level for the commands you want to execute. If the system administrative group parameter (*sysadm_group*) is defined in the database manager configuration file, the user performing administration tasks (such as cataloging, starting and stopping the database manager, or creating the sample database) must belong to the System Administrative group. The *sysadm_group* parameter is not set when DB2 is first installed; therefore, the following criteria is used to determine SYSADM authority:

OS/2 A valid DB2 user ID which belongs to the UPM Administrator or Local Administrator group.

UNIX A valid DB2 username that belongs to the primary group of the instance owner's user ID.

Windows 95 Any Windows 95 user.

Windows NT A valid DB2 username which belongs to the local administrators group.

You can override this default case by explicitly providing a group name for the *sysadm_group* parameter. If you do this, the only way to assign SYSADM authority is by membership to this group.

See "Working with the System Administrative Group" on page 170 for more information

See Appendix F, "Naming Rules" on page 315 for a list of naming restrictions.

As well as using the local operating system security to authenticate users, there is also the option of using DCE security. Refer to the *Administration Guide* for information on DCE security.

Creating the Sample Database

If you did not create the sample database after installation, you might want to create the sample database, called `SAMPLE`, that is provided with DB2 Universal Database. It contains tables, views, and fictitious data that you can use to verify that DB2 is correctly installed. For a detailed description of the sample database, refer to the *Administration Guide*.

The sample database is automatically cataloged with the database alias `SAMPLE` when it is created. By default, the database is created in the path specified by the *DFTDBPATH* database manager configuration parameter. You must ensure that this path does *not* specify an NFS-mounted directory.

To create the sample database, perform the following steps:

1 Log in as the instance owner.

2 Run the start up script as follows:

```
. INSTHOME/sqllib/db2profile    (for Bourne or Korn shell)
source INSTHOME/sqllib/db2cshrc (for C shell)
```

where *INSTHOME* is the home directory of the instance

3 Create the sample database by issuing the following command:

```
db2sampl
```

Note: The database partition server from which you issue the **db2sampl** command becomes the catalog node for the database.

A database partition is created for the database on every database partition server in the instance.

This command may take a few minutes to process. There is no completion message; when the command prompt returns, the process is complete.

To remove the `SAMPLE` database, enter the following command as the instance owner in the command line processor:

```
drop database sample
```

Connecting to a Database

You need to connect to a database before you can use SQL statements to query or update it. The CONNECT statement associates a database connection with a user ID.

A database is created in an instance using one of three different authentication types: CLIENT, SERVER, or DCS. If an authentication type is not specified when the instance is created, the default SERVER authentication is used. Depending on the authentication type and where the database resides, you connect to remote databases differently.

Connecting to a local database

To connect to a local database called SAMPLE, type the following command in the Command Center or the command line processor:

```
connect to sample
```

Connecting to a remote database

- To connect a client to a remote database called SAMPLE, using SERVER authentication, type the following command in the command line processor:

  ```
  connect to sample user userid using password
  ```

  (Be sure to choose values for *userid* and *password* that are valid on the server system.)

- To connect a client to a remote database called SAMPLE, using CLIENT authentication, type the following command in the command line processor:

  ```
  connect to sample
  ```

> If you are connecting to the database from an *OS/2* client, and are not currently logged on, you will be prompted to log on using a user ID and password.
>
> If you are connecting to a database from a *Windows NT* client using CLIENT authentication, and you provide a user ID and password on the **connect** command, you must start the DB2 for Windows NT Security Service on the system. The Security Service is installed by DB2 and set up to run as a Windows NT service; however, it is not started automatically. To start the DB2 Security Service, double-click on the **Services** icon in the **Control Panel** folder. Select the **DB2 Security Server** option and click on the **Start** push button.

When a successful connection is made, a message similar to the following is displayed:

```
Database product       = DB2/6000 5.0.0
SQL authorization ID   = USERID
Local database alias   = SAMPLE
```

If you receive an error message, make sure that the SAMPLE database exists on the server and the database manager was started on the server.

For more information on authentication, refer to the *Administration Guide*.

Working with the System Administrative Group

By default, System Administrative (SYSADM) authority is granted to the following:

OS/2 Any valid DB2 user ID which belongs to the UPM Administrator or Local Administrator group.

UNIX Any valid DB2 username that belongs to the primary group of the instance owner's user ID.

Windows 95 Any Windows 95 user.

Windows NT Any valid DB2 username which belongs to the local Administrators group.

Upgrading from a Try-and-Buy License

Try-and-Buy versions are available for some of the DB2 products. If you install a product originally as Try-and-Buy, you do not need to reinstall the same product to upgrade the license. Simply add the license from the DB2 CD-ROM that comes in the product box.

Note: *Your Proof of Entitlement* and *License Information* booklets identify the products for which you are licensed.

The following steps describe how to upgrade the license:

1 Log in as root.

2 Mount the CD-ROM, if required.

3 Update your DB2 product license with the following command:

```
DB2DIR/cfg/db2licinst license_filename
```

where DB2DIR	= /usr/lpp/db2_05_00	on AIX
	= /opt/IBMdb2/V5.0	on Solaris

and `license_filename` is the full pathname for the license file that corresponds to the product you have purchased. The names of the license files for these products are:.

db2eent.lic DB2 Universal Database Extended Enterprise Edition for AIX

db2eent.nod DB2 Universal Database Extended Enterprise Edition for Solaris

For example, if the CD-ROM is mounted in the */cdrom* directory and the name of the license file is *db2eent.lic*, the command should be as follows:

- For AIX:

```
DB2DIR/cfg/db2licinst /cdrom/db2/license/db2eent.nod
```

- For Solaris:

```
DB2DIR/cfg/db2licinst /cdrom_unnamed/db2/license/db2eent.nod
```

Note: The DB2 product license key can be found in the *nodelock* file, located as follows:

- For AIX:

```
/usr/lib/netls/conf/nodelock
```

- For Solaris:

```
/var/netls/nodelock
```

Chapter 17. Working with Instances

The information in this section describes DB2 server instances and how to work with them.

An instance is a logical database manager environment where you catalog databases and set configuration parameters. Depending on your needs, you can create more than one instance. You can use multiple instances to do the following:

- Use one instance for a development environment and another instance for a production environment.
- Tune an instance for a particular environment.
- Restrict access to sensitive information.
- Control the assignment of SYSADM, SYSCTRL, and SYSMAINT authority for each instance.
- Optimize the database manager configuration for each instance.
- Limit the impact of an instance crash. In the unlikely event of an instance crash, only one instance is impacted. The other instance may continue to function normally.

Each additional instance requires additional system resources (virtual storage and disk space) and more administration.

Instance Directory

The instance directory stores all information that pertains to a database instance. You cannot change the location of the instance directory once it is created. The directory contains:

- The database manager configuration file
- The system database directory
- The node directories
- The DB2 diagnostic file (db2diag.log)
- The node configuration file
- Any other files that contain debugging information such as the exception/register dump or the call stack for the DB2 processes.

In a partitioned database system, the instance directory is shared between all database partition servers belonging to the instance. Therefore, the instance directory must be created on a mounted file system that all machines in the instance can access.

For information on adding nodes to participate in an existing instance, refer to the *Administration Guide*.

Creating Instances

If you are running file collections on an RS/6000 SP system that is running AIX, ensure that the instance user ID is created from the file collections server. When file collections is run, the user ID and group will be deleted if added to an RS/6000 SP node that is *not* the file collections server.

To see if file collections is running, issue the **crontab -l** command from one of the RS/6000 SP nodes and look for an entry such as:

```
10 * * * * /var/sysman/supper update sup.admin
user.admin node.root
```

The user ID of the instance owner, and the group that is the System Administrative (SYSADM) group, are associated with every instance. The instance owner an the SYSADM group are assigned during the process of creating the instance. One user ID can be used for only one instance. That user ID is also referred to as the *instance owner*.

Each instance owner must have a unique home directory. All of the files necessary to run the instance are created in the home directory of the instance owner's user ID. If it becomes necessary to remove the instance owner's user ID from the system, you could potentially lose files associated with the instance and lose access to data stored in this instance. For this reason, it is recommended that you dedicate an instance owner user ID to be used exclusively to run DB2.

The primary group of the instance owner user ID is also important. This primary group automatically becomes the system administrative group for the instance and gains SYSADM authority over the instance. (Administrative authority allows users to start, stop, or change the database instance.) Other user IDs that are members of the primary group of the instance user ID also gain this level of authority. For this reason, you may want to assign the instance owner ID to a primary group that is reserved for the administration of instances. (Also make sure that you assign a primary group to the instance owner user ID; otherwise, the system-default primary group is used.)

If you already have a group that you want to make the system administrative group for the instance, you can simply assign this group as the primary group when you create the instance owner user ID. To give other users administrative authority on the instance, add them to the group that is assigned as the system administrative group.

To separate SYSADM authority between instances, ensure that each instance owner user ID uses a different primary group. However, if you choose to have common

SYSADM authority over multiple instances, you can use the same primary group for multiple instances.

It is possible to add an instance to any machine. If you use different machines to create instances, then the list of instances that is stored on any machine may not be a complete list. We recommend creating every instance on one machine, so that the list of instances is complete on that machine. We also recommend that this node be the Administration Server node.

When you create an instance on a machine that has an Administration Server on it, its SYSADM group is automatically added to the secondary group list of the Administration Server so that you can use the CCA and the Control Center to perform administration tasks on that instance.

To create additional instances using the **db2icrt** command, enter:

```
DB2DIR/instance/db2icrt -u FencedID InstName
```

where *InstName* is an alphanumeric string up to eight characters long (see “Usernames, Group Names, and Instance Names” on page 317 for more information) and

where DB2DIR	= /usr/lpp/db2_05_00	on AIX
	= /opt/IBMdb2/V5.0	on Solaris

Run the **db2icrt** command from the machine where the home directory of the instance (*INSTHOME*) physically resides. Because the *INSTHOME* directory is NFS-mounted on all machines, you only have to run the **db2icrt** command on one machine.

The syntax of the **db2icrt** command is:

```
►►──db2icrt──[-h| -?]──[-d]──[-a AuthType]────[-u FencedID] ─────────►

►──── InstName ─────────────────────────────────────────────────────►◄
```

where:

-h | -? Display a help menu for this command.

-d Sets the debug mode that you can use for problem determination.

-a *AuthType* Is an optional parameter that specifies the authentication type for the instance. Valid authentication types are SERVER, CLIENT, and DCS. If the *-a* parameter is not specified, the authentication type defaults to SERVER, if a DB2 server is installed. Otherwise, the *AuthType* is set to CLIENT.

Notes:

1. All databases in the instance have the same authentication type.
2. DCE authentication is not valid for this command; however, you can enable DCE authentication for an instance. For more information, refer to the *Administration Guide*.

-u *FencedID* Is the user under which the fenced UDFs and stored procedures will execute. This is not required if you install the DB2 Client Application Enabler or the DB2 Software Developer's Kit. For other products, this is a required parameter.

Note: *FencedID* may not be *root* or *bin*.

InstName Is the login name of the instance owner.

When you create an instance, ensure that its primary and secondary groups are different from the Administration Server's primary (SYSADM) group. When you create an instance on the same machine as the Administration Server, its SYSADM group is automatically added to the secondary group list of the Administration Server so that you can use the Control Center to perform administration tasks on that instance.

Examples:

- To create an instance for a DB2 server, you can use the following command:

```
db2icrt -u db2fenc1 db2inst1
```

- To create an instance for a DB2 client, you can use the following command:

```
db2icrt db2inst1
```

When an instance is created, its name is also added to the list of instances on the system.

The **db2icrt** command creates the *INSTHOME/sqllib* directory, where *INSTHOME* is the home directory of the instance owner.

> To avoid a potential loss of data if an instance is deleted, you should not create user files or directories under the *INSTHOME/sqllib* directory, other than those created by DB2. Exceptions are:
>
> 1. If your system supports fenced user defined functions and fenced stored procedures, put the fenced applications in the *INSTHOME/sqllib/function* directory.
> 2. The location of the *db2nodes.cfg* file. This file should be put in the *INSTHOME/sqllib* directory, where *INSTHOME* is the home directory of the instance owner.

Listing Instances

To get a list of all the instances that are available on a system, run the **db2ilist** command. The **db2ilist** command is located in:

```
DB2DIR/bin/db2ilist
```

where DB2DIR	= /usr/lpp/db2_05_00	on AIX
	= /opt/IBMdb2/V5.0	on Solaris

To determine which instance applies in the current session, enter:

```
echo $DB2INSTANCE
```

Starting and Stopping an Instance

You must start an instance before you can perform the following tasks:

- Connect to a database on the instance.
- Precompile an application.
- Bind a package to a database.
- Access host databases.

To start an instance:

1. Log in as the instance owner.
2. Run the start up script as follows:

   ```
   . INSTHOME/sqllib/db2profile    (for Bourne or Korn shell)
   source INSTHOME/sqllib/db2cshrc (for C shell)
   ```

   where INSTHOME is the home directory of the instance that you want to use.
3. Start the DB2 database manager by entering the **db2start** command from a command line.

To stop a database instance:

1. Log in as the instance owner.
2. Run the start up script as follows:

   ```
   . INSTHOME/sqllib/db2profile    (for Bourne or Korn shell)
   source INSTHOME/sqllib/db2cshrc (for C shell)
   ```

   where INSTHOME is the home directory of the instance that you want to use.

3 Stop the DB2 database manager by entering the **db2stop** command from a command line.

Auto-Starting Instances

To enable an instance to auto-start after each system restart, issue the following command:

```
DB2DIR/bin/db2iauto -on InstName
```

where DB2DIR	= /usr/lpp/db2_05_00	on AIX
	= /opt/IBMdb2/V5.0	on Solaris

and where *InstName* is the login name of the instance.

To prevent an instance from auto-starting after each system restart, issue the following command:

```
DB2DIR/bin/db2iauto -off InstName
```

where DB2DIR	= /usr/lpp/db2_05_00	on AIX
	= /opt/IBMdb2/V5.0	on Solaris

and where *InstName* is the login name of the instance.

Setting the Operating Environment for Each Instance User

Before users can use DB2, the database environment for each user must be updated so that it can access an instance and run the DB2 programs. This applies to all users (including administrative users).

The sample scripts, **db2profile** (for Bourne or Korn shell) and **db2cshrc** (for C shell), are provided to help you set the database environment. These scripts are available in the *INSTHOME/sqllib* directory, where *INSTHOME* is the home directory of the instance owner.

These scripts are also required when the **db2start** and the **db2stop** commands are run to start or stop the DB2 Extended Enterprise Edition database manager.

The instance owner (or any user that belongs to the instance's SYSADM group) can customize the script for all users of an instance. Alternatively, the script can be copied and customized for each user.

The sample script contains statements to:

- Update a user's PATH by adding the following directories to the existing search path:

  ```
  INSTHOME/sqllib/bin
  INSTHOME/sqllib/adm
  INSTHOME/sqllib/misc
  ```

  where INSTHOME is the home directory of the instance.

- Set the *DB2INSTANCE* environment variable to the instance name.

Setting the DB2 Environment Automatically

By default, the scripts affect the user environment for the duration of the current session only. You can change the *.profile* file to enable it to run the *db2profile* script automatically when the user logs on using the Bourne or Korn shell. For users of the C shell, you can change the *.login* file to enable it to run the *db2cshrc* script file.

Add one of the following statements to the *.profile* or *.login* script files.

- For users who share one version of the script, add:

  ```
     . INSTHOME/sqllib/db2profile    (for Bourne or Korn shell)
     source INSTHOME/sqllib/db2cshrc (for C shell)
  ```

  where INSTHOME is the home directory of the instance that you want to use.

- For users that have a customized version of the script in their home directory, add:

  ```
  . USERHOME/db2profile                 (in Bourne or Korn Shell)
  source USERHOME/db2cshrc              (in C Shell)
  ```

  where USERHOME is the home directory of the user.

Setting the DB2 Environment Manually

To choose which instance that you want to use, enter one of the following statements at a command prompt. The period (.) and the space are required.

- For users who share one version of the script, add:

  ```
     . INSTHOME/sqllib/db2profile    (for Bourne or Korn shell)
     source INSTHOME/sqllib/db2cshrc (for C shell)
  ```

  where INSTHOME is the home directory of the instance that you want to use.

- For users that have a customized version of the script in their home directory, add:

  ```
  . USERHOME/db2profile               (in Bourne or Korn Shell)
  source USERHOME/db2cshrc            (in C Shell)
  ```

  where USERHOME is the home directory of the user.

If you want to work with more than one instance at the same time, run the script for each instance that you want to use in separate windows. For example, assume that you have two instances called test and prod, and their home directories are */u/test* and */u/prod*.

In window 1:

- In Bourne or Korn Shell, enter:

  ```
  . /u/test/sqllib/db2profile
  ```

- In C Shell, enter:

  ```
  source /u/test/sqllib/db2cshrc
  ```

In window 2:

- In Bourne or Korn Shell, enter:

  ```
  . /u/prod/sqllib/db2profile
  ```

- In C Shell, enter:

  ```
  source /u/prod/sqllib/db2cshrc
  ```

Use window 1 to work with the test instance and window 2 to work with the prod instance.

Issue the **which db2** command to ensure that your search path has been set up correctly. This command returns the absolute path of the DB2 CLP executable. Verify that it is located under the instance's *sqllib* directory.

Updating Instances

Existing instances are designed to be as independent as possible from the effects of subsequent installation and removal of products.

In most cases, existing instances automatically inherit or lose access to the function of the product being installed or removed. However, if certain executables or components are installed or removed, existing instances do not automatically inherit the new system configuration parameters or gain access to all the additional function. The instance must be updated.

If DB2 is updated by installing a PTF or a patch, all the existing DB2 instances should be updated using the **db2iupdt** command. You should also update the Administration Server using the **dasiupdt** command. See “Updating the Administration Server” on page 187 for further information.

Running the **db2iupdt** command updates the specified instance by performing the following:

1 Replaces the files in the *INSTHOME/sqllib* directory, where `INSTHOME` is the home directory of the instance.

2 If the node type is changed, then a new database manager configuration file is created. This is done by merging relevant values from the existing database manager configuration file with the default database manager configuration file for the new node type. If a new database manager configuration file is created the old file is backed up to:

```
INSTHOME/sqllib/backup/db2systm.old
```

The **db2iupdt** command is available in the *DB2DIR/instance* directory,

where DB2DIR	= /usr/lpp/db2_05_00	on AIX
	= /opt/IBMdb2/V5.0	on Solaris

> Because *INSTHOME* is mounted, you only have to run the **db2iupdt** command once to update the instance.

The syntax of the **db2iupdt** command is:

```
►►── db2iupdt ──[-h| -?]───[-d]────[-a AuthType]────[-u FencedID]──►

►──── InstName ─────────────────────────────────────────────────────►◄
```

where:

-h | -? Displays a help menu for this command.

-d Sets the debug mode, which you can use for problem analysis.

-a *AuthType* Is an optional parameter that specifies the authentication type for the instance. Valid authentication types are SERVER, CLIENT, and DCS. If the -a parameter is not specified, the authentication type defaults to SERVER, if a server product is installed. Otherwise, the *AuthType* is set to CLIENT.

Notes:

1. The authentication type of the instance applies to all databases that are owned by the instance.
2. The authentication type DCE is valid. However, it is not valid to choose DCE as an *AuthType* for this command. To enable DCE authentication for an instance, refer to the *Administration Guide*.

-u *FencedID*
: Is the user under which the fenced UDFs and stored procedures will run. This is not required if you install the DB2 Software Developer's Kit or the DB2 Client Application Enabler. For other products, this is a required parameter.

  Note: *FencedID* may not be root or bin.

InstName
: Is the login name of the instance owner.

If a server product is installed, the syntax is:

```
db2iupdt -u FencedID InstName
```

Examples:

- If you installed the DB2 Extended Enterprise Edition after the instance was created, you can use the following command to update that instance:

  ```
  db2iupdt -u db2fenc1 db2inst1
  ```

- To update all other instances, you can use the following command:

  ```
  db2iupdt db2inst1
  ```

The **db2iupdt** command updates the files in the *INSTHOME/sqllib* directory, where `INSTHOME` is the home directory of the instance owner.

Updating Version 5 Single-Partition to Multipartition Instances

To update a Version 5 single-partition instance to a multipartition instance, perform the following steps. It is assumed that you have already installed DB2 Extended Enterprise Edition on the machines that you want to participate in this instance.

1 Enable the Fast Communications Manager (FCM) for all instances. For more information, see “Step 3. Enable the Fast Communication Manager” on page 61 for AIX or “Step 5. Enable the Fast Communication Manager” on page 80 for Solaris.

2 Issue the **db2iupdt** command to update the instance to the multipartition format. For more information, see “Updating Instances” on page 180

3 Create the node configuration file. For more information, see “Step 5. Create the Node Configuration File” on page 64 for AIX or “Step 8. Create the Node Configuration File” on page 84 for Solaris.

4 Make DB2 Extended Enterprise Edition available. For more information, see “Step 6. Make DB2 Extended Enterprise Edition Available” on page 66 for AIX or “Step 9. Make DB2 Extended Enterprise Edition Available” on page 86 for Solaris.

5 Update the Administration Server using the **dasiupdt**. For more information see “Updating the Administration Server” on page 187.

You will also have to reinstall the Control Center on any administrative client if you want to use that client to administer the multipartition instance.

After the instance is migrated to the multipartition format, any existing databases will have partitions at each database partition server. In addition, the default nodegroups are also created. For more information on the default nodegroups, see “Nodegroups and Data Partitioning” on page 26. For information about redistributing data to the new database partitions, refer to the *Administration Guide*.

Removing Instances

To remove an instance, perform the following steps:

1 Log in as the instance owner.

2 Stop the instance by issuing the **db2stop** command.

3 Back up files in the *INSTHOME/sqllib* directory, if needed.

For example, you might want to save the database manager configuration file, *db2systm*, the *db2nodes.cfg* file, or user defined function or fenced stored procedure applications in *INSTHOME/sqllib/function*, where INSTHOME is the home directory of the instance owner.

4 Log off as the instance owner.

5 Log in as root.

6 Remove the instance by executing the **db2idrop** command:

```
DB2DIR/instance/db2idrop InstName
```

where DB2DIR	= /usr/lpp/db2_05_00	on AIX
	= /opt/IBMdb2/V5.0	on Solaris

and *InstName* is the login name of the instance.

Because *INSTHOME* is mounted, you only have to run the **db2idrop** command once to remove the instance.

7 As *root*, remove the instance owner's user ID and group (if used only for that instance). Do not remove these if you are planning to re-create the instance.

This step is optional since the instance owner and the instance owner group may be used for other purposes.

The **db2idrop** command removes the instance entry from the list of instances and removes the *INSTHOME/sqllib* directory.

Chapter 18. Using the Administration Server

The Administration Server enables you to administer DB2 server systems from a local or remote client, using the Control Center. You must have a running Administration Server if you want to use the Client Configuration Assistant or the Control Center. For example, you can start or stop DB2, or schedule jobs to be run at the server from a local or remote system.

The Administration Server is also used to support client configuration using the Client Configuration Assistant, and system reporting using the Control Center. You can create only one Administration Server on a machine.

This section shows you how to manually create an Administration Server. It also describes how to start, stop, and remove an Administration Server, set up an Administration Server to use the CCA and the Control Center, and set an Administration Server's user account.

Creating the Administration Server

Use the **dasicrt** command to create an Administration Server. You must run this command on the machine where you have DB2 physically installed and have root authority.

The syntax of the **dasicrt** command is as follows:

```
DB2DIR/instance/dasicrt ASName
```

where DB2DIR	= /usr/lpp/db2_05_00	on AIX
	= /opt/IBMdb2/V5.0	on Solaris

and where *ASName* is the name of the Administration Server, which is composed of a string of up to eight alphanumeric characters long. See Appendix F, “Naming Rules” on page 315 for more information.

Note: If you are running NIS or NIS+, you need to set up the user and group names in such a way that:

- The primary group of the Administration Server must be in the secondary group of all the instances.

- The secondary group of the Administration Server must contain the primary group of all the instances

Secondary group lists will be modified automatically only if NIS or NIS+ is running on your system.

Because a user ID can only own one instance, you must have a separate user ID to own each Administration Server that you create.

Starting the Administration Server

To start the Administration Server, you must perform the following steps:

1 Log in as the Administration Server owner.

2 Run the start up script as follows:

```
. INSTHOME/sqllib/db2profile    (for Bourne or Korn shell)
source INSTHOME/sqllib/db2cshrc (for C shell)
```

where INSTHOME is the home directory of the instance.

3 Start the Administration Server using the **db2admin** command as follows:

```
db2admin start
```

The Administration Server is automatically started after each system reboot.

Stopping the Administration Server

To stop the Administration Server, you must perform the following steps:

1 Log in as the Administration Server owner.

2 Run the start up script as follows:

```
. INSTHOME/sqllib/db2profile    (for Bourne or Korn shell)
source INSTHOME/sqllib/db2cshrc (for C shell)
```

where INSTHOME is the home directory of the instance.

3 Stop the Administration Server using the **db2admin** command as follows:

```
db2admin stop
```

where DB2DIR	= /usr/lpp/db2_05_00	on AIX
	= /opt/IBMdb2/V5.0	on Solaris

Listing the Administration Server

To obtain the name of the Administration Server on your machine, issue the **dasilist** command as follows:

```
DB2DIR/bin/dasilist
```

where DB2DIR	= /usr/lpp/db2_05_00	on AIX
	= /opt/IBMdb2/V5.0	on Solaris

Updating the Administration Server

If DB2 is updated by installing a PTF or a patch, all Administration Servers as well as all existing instances should be updated. To update an Administration Server, use the **dasiupdt** command. The **dasiupdt** command is available in the *DB2DIR/instance* directory,

where DB2DIR	= /usr/lpp/db2_05_00	on AIX
	= /opt/IBMdb2/V5.0	on Solaris

The syntax of the **dasiupdt** command is:

```
►►── dasiupdt ── [-h| -?] ── [-d] ── InstName ──────────────►◄
```

where:

-h | -? Displays a help menu for this command.

-d Sets the debug mode, which you can use for problem analysis.

InstName Is the login name of the instance owner.

Removing the Administration Server

To remove an Administration Server, you must perform the following steps:

1 Log in as the Administration Server owner.

2 Run the start up script as follows:

```
. INSTHOME/sqllib/db2profile    (for Bourne or Korn shell)
source INSTHOME/sqllib/db2cshrc (for C shell)
```

where INSTHOME is the home directory of the instance.

3 Stop the Administration Server using the **db2admin** command as follows:

```
db2admin stop
```

4 Backup the files in the *ASHOME/sqllib* directory, if needed, where *ASHOME* is the home directory of the Administration Server.

5 Log off.

6 Log in as *root* and remove the Administration Server using the **dasidrop** command as follows:

```
DB2DIR/instance/dasidrop ASName
```

where DB2DIR	= /usr/lpp/db2_05_00	on AIX
	= /opt/IBMdb2/V5.0	on Solaris

and ASName is the name of the instance being removed.

Note: The **dasidrop** command removes the *sqllib* directory under the home directory of the Administration Server.

Chapter 19. Running Your Own Applications

Various types of applications can access DB2 databases:

- Applications developed using the DB2 Software Developer's Kit that include embedded SQL, APIs, stored procedures, user defined functions or calls to the DB2 CLI.
- ODBC applications such as Lotus Approach.
- JDBC applications and applets.
- Net.Data macros containing HTML and SQL.

An application on a DB2 client can access a remote database without knowing its physical location. The DB2 client determines the location of the database, manages the transmission of the requests to the database server, and returns the results.

In general, to run a database client application, use the following steps:

1. Ensure the server is configured and running.

   Be sure that the database manager is started on the database server to which the application program is connecting. If it is not, you must issue the **db2start** command at the server before starting the application.

2. Ensure that you can connect to the database that the application uses.
3. Bind the utilities and the applications to the database. See "Binding Database Utilities" for information about binding the utilities.
4. Run the application program.

Binding Database Utilities

You must bind the database utilities (import, export, reorg, the command line processor, and DB2 CLI) to each database before they can be used with that database. In a network environment, if you are using multiple clients that run on different operating systems or are at different versions of DB2, you must bind the utilities once for each operating system/DB2-version combination.

Binding a utility creates a *package*, which is an object that includes all of the information that is needed to process specific SQL statements from a single source file.

The bind files are grouped together in different *.lst* files in the *bnd* directory, under the installation directory. Each file is specific to a server.

How you bind the database utilities to a database depends on your workstation's operating system:

- On OS/2, Windows 95, and Windows NT operating systems, you can use the Client Configuration Assistant:

  1. Start the Client Configuration Assistant (CCA).
  2. Select the database to which you want to bind the utilities.
  3. Click on the **Bind** push button.
  4. Select the **Bind DB2 Utilities** radio button.
  5. Click on the **Continue** push button.
  6. Enter a user ID and password to connect to the database. The user ID must have the authority to bind new packages against the database.
  7. Select the utilities you want to bind and click on **OK**.

- On all operating systems, you can use the command line processor:

  1. Change to the *bnd* directory in the install path. For example: *INSTHOME/sqllib/bnd*
  2. Connect to the database using the command:

     ```
     connect to database_alias
     ```

     where *database_alias* is the name of the database to which you want to connect.
  3. Issue the following commands in the Command Center or the command line processor:

     ```
     "bind @db2ubind.lst messages bind.msg grant public"
     "bind @db2cli.lst messages clibind.msg grant public"
     ```

     In this example, `bind.msg` and `clibind.msg` are the output message files, and EXECUTE and BIND privileges are granted to *public*.
  4. Reset the connection to the database by issuing the following command:

     ```
     connect reset
     ```

     For more information on the **bind** command, refer to the *Command Reference*.

Notes:

1. The *db2ubind.lst* file contains the list of bind (.bnd) files required to create the packages for the database utilities. The *db2cli.lst* file contains the list of bind (.bnd) files required to create packages for the DB2 CLI and the DB2 ODBC driver.
2. Binding may take a few minutes to complete.
3. To bind to databases that reside on OS/390, MVS, VM, or AS/400, refer to the *DB2 Connect User's Guide* for instructions.
4. If you have bind authority, the first time you use the DB2 CLI or ODBC driver, the DB2 CLI packages will be bound automatically.

> If the applications that you are using require binding to the database, you can use the Client Configuration Assistant's Bind facility, or the command line processor, to perform the bind action.

Running CLI/ODBC Programs

The DB2 Call Level Interface (CLI) run-time environment and the ODBC driver are included with the DB2 Client Application Enabler. This is contained on the DB2 Client Application Enablers CD-ROM or can be downloaded from the Web page at http://www.software.ibm.com/data/db2.

This support enables applications developed using ODBC and DB2 CLI APIs to work with any DB2 server. DB2 CLI application development support is provided by the DB2 Software Developer's Kit (DB2 SDK) which is part of the separately orderable DB2 Application Development Kit product.

Before DB2 CLI or ODBC applications can access DB2, the DB2 CLI packages must be bound on the server. Although this will occur automatically on the first connection if the user has the required authority to bind the packages, it is recommended that the administrator do this first with each version of the client on each platform that will access the server. See “Binding Database Utilities” on page 189 for specific details.

The following general steps are required on the client system to give DB2 CLI and ODBC applications access to DB2 databases. These instructions assume that you have successfully connected to DB2 using a valid user ID and password. Depending on the platform many of these steps are automatic. For complete details, see the section that deals specifically with your platform.

1 Use the CCA to add the remote system (if you have separate client and server machines) so that its instances and databases can be made known to the Control Center, then add the instances and databases for that system. (Your local system is represented by **Local** icon.) If you do not have access to this program you can use the **catalog** command in the command line processor.

2 On all platforms other than OS/2 and Windows 3.1, the DB2 CLI/ODBC driver is automatically installed when the DB2 Client Application Enabler is installed, and therefore nothing needs to be done. On OS/2 and Windows 3.1 you must use the **Install ODBC Driver** icon to install both the DB2 CLI/ODBC driver and the ODBC driver manager.

3 To access the DB2 database from ODBC:

  a The Microsoft, Visigenic, or other ODBC Driver Manager must already be installed (this is done by default during the installation of DB2).

  b The DB2 databases must be registered as ODBC data sources. The ODBC driver manager does not read the DB2 catalog information; instead it references its own list of data sources.

  c If a DB2 table does not have a unique index then many ODBC applications will open it as read-only. Create a unique index for each DB2 table that is to be updated by an ODBC application. Refer to the **CREATE INDEX** statement in the *SQL Reference*. Using the Control Center you would alter the settings of the table, then select the **Primary Key** tab and move one or more columns from the available columns list over to the primary key columns list. Any column you select as part of the primary key must be defined as NOT NULL.

4 Various CLI/ODBC Configuration Keywords can be set to modify the behavior of DB2 CLI/ODBC and the applications using it.

5 If you followed the above steps to install ODBC support, and added DB2 databases as ODBC data sources, your ODBC applications will now be able to access them.

After the platform specific instructions there are further details on the following topics:

- “How to Bind the DB2 CLI/ODBC Driver to the Database” on page 202
- “How to Set CLI/ODBC Configuration Keywords” on page 202
- “Configuring db2cli.ini” on page 203

Platform Specific Details for CLI/ODBC Access

The platform specific details on how to give DB2 CLI and ODBC applications access to DB2 are divided into the following categories:

- “Windows 3.1, Windows 95 and Windows NT Client Access to DB2 using CLI/ODBC” on page 193
- “OS/2 Client Access to DB2 using CLI/ODBC” on page 195
- “UNIX Client Access to DB2 using CLI/ODBC” on page 197

Windows 3.1, Windows 95 and Windows NT Client Access to DB2 using CLI/ODBC

Before DB2 CLI and ODBC applications can successfully access a DB2 database from a Windows 3.1, Windows 95, or Windows NT client, perform the following steps on the client system:

1 The DB2 database (and node if the database is remote) must be cataloged. To do so, use the CCA (or the command line processor). See "Configuring Database Connections" on page 243 for more details.

For more information refer to the on-line help in the CCA (or the **CATALOG DATABASE** and **CATALOG NODE** commands in the *Command Reference*).

2 Verify that the Microsoft ODBC Driver Manager and the DB2 CLI/ODBC driver are installed. On Windows 95 and Windows NT they are both installed with DB2 unless the ODBC component is manually unselected during the install. On Windows 3.1 you must use the **Install ODBC Driver** icon to install the Microsoft ODBC Driver Manager and the DB2 CLI/ODBC driver.

To verify that they both exist on the machine:

a Run the Microsoft ODBC Administrator from the icon in the Control Panel, or issue the appropriate command from the command line: **odbcad32.exe** for Windows 95 or Windows NT operating systems, **odbcadm.exe** on Windows 3.1.

b Click on the **Drivers** push button, or the **ODBC Drivers** tab (depending on the platform).

c Verify that "IBM DB2 ODBC Driver" is shown in the list.

If either the Microsoft ODBC Driver Manager or the IBM DB2 CLI/ODBC driver is not installed, then rerun the DB2 install and select the ODBC component in Windows 95 or Windows NT, or double-click on the **Install ODBC Driver** icon in the DB2 program group in Windows 3.1.

3 Register the DB2 database with the ODBC driver manager as a *data source*. On Windows 95 and Windows NT you can make the data source available to all users of the system (a system data source), or only the current user (a user data source). Use either of these methods to add the data source:

- Using the CCA:

a Select the DB2 database alias that you want to add as a data source.

b Click on the **Properties** push button. The Database Properties window opens.

c Select the **Register this database for ODBC** check box.

d On Windows 95 and Windows NT you can use the radio buttons to add the data source as either a user or system data source.

- Using the Microsoft **32bit ODBC Administration tool**, which you can access from the icon in the Control Panel or by running **odbcad32.exe** from the command line:

  a On Windows 95 and Windows NT the list of user data sources appears by default. If you want to add a system data source click on the **System DSN** button, or the **System DSN** tab (depending on the platform).

  b Click on the **Add** push button.

  c Double-click on the IBM DB2 ODBC Driver in the list.

  d Select the DB2 database to add and click on **OK**.

- On Windows 95 and Windows NT there is a command that can be issued in the command line processor to register the DB2 database with the ODBC driver manager as a data source. An administrator could create a command line processor script to register the required databases. This script could then be run on all of the machines that require access to the DB2 databases through ODBC.

  The *Command Reference* contains more information on the CATALOG command:

  ```
  CATALOG [ user | system ] ODBC DATA SOURCE
  ```

4 Optional: Configure the DB2 CLI/ODBC driver using the CCA:

a Select the DB2 database alias you want to configure.

b Click on the **Properties** push button. The Database Properties window opens.

c Click on the **Settings** push button. The CLI/ODBC Settings window opens.

d Click on the **Advanced** push button. You can set the configuration keywords in the window that opens. These keywords are associated with the database *alias name*, and affect all DB2 CLI/ODBC applications that access the database. The online help explains all of the keywords, as does an appendix in the *Installing and Configuring DB2 Clients* manual.

For information on manually editing this file (*db2cli.ini*), see “Configuring db2cli.ini” on page 203.

5 Optional: Using 16-bit ODBC applications:

Although not directly related to DB2, some users have experienced problems when running 16-bit ODBC applications on Windows 95 or Windows NT.

Both 16- and 32-bit applications use the same 32-bit IBM DB2 CLI/ODBC Driver. The Microsoft ODBC Driver Manager takes care of the conversion between the 16-bit application and the 32-bit ODBC driver.

Some 16-bit applications were shipped with an old set of 16-bit Driver Manager DLLs (that is, before 1995). These DLLs do not function well in the 32-bit environment. The following symptoms may occur when running the old DLLs:

- Traps occur in odbc.dll or other ODBC related DLLs.
- 16-bit applications do not see data sources cataloged using the 32-bit ODBC Administrator (from the Control Panel).

If (and only if) you are experiencing these problems you may want to update the Microsoft 16-bit ODBC Driver Manager DLLs. These files are provided with DB2 in the *SQLLIB\MSODBC16* subdirectory. Use these new DLLs to replace the older versions that are located in the *system* subdirectory of the Windows operating system directory.

6 If you have installed ODBC access (as described above), you can now access DB2 data using ODBC applications. Start the ODBC application and go to the Open window. Select the **ODBC databases** file type. The DB2 databases that you added as ODBC data sources will be selectable from the list. Many ODBC applications will open the table as read-only unless a unique index exists.

If you require additional information at this point you can refer to the following topics in “Detailed Configuration Information” on page 201:

- “How to Bind the DB2 CLI/ODBC Driver to the Database” on page 202
- “How to Set CLI/ODBC Configuration Keywords” on page 202
- “Configuring db2cli.ini” on page 203

OS/2 Client Access to DB2 using CLI/ODBC

Before DB2 CLI and ODBC applications can successfully access a DB2 database from an OS/2 client, perform the following steps on the client system:

1 The DB2 database (and node if the database is remote) must be cataloged. To do so, use the CCA (or the command line processor).

For more information see the on-line help in the CCA or “Configuring Database Connections” on page 243.

2 If you are using ODBC applications to access DB2 data, perform the following steps. (If you are using only CLI applications, skip this step and go to the next step.)

a Check that the ODBC Driver Manager (Visigenic or Intersolv) and the DB2 CLI/ODBC driver are installed:

1 Run the ODBC Administration tool in one of two ways:

- Double-click on the **Control Panel** icon from the **Main** Folder in WIN-OS/2 or the **ODBC** Folder in OS/2, and double-click on the **ODBC Administrator** icon.
- Run **odbcadm.exe** from the command line.

The Data Sources window opens.

2 Click on the **Drivers** push button. The Drivers window opens.

3 Verify that "IBM DB2 ODBC Driver" is shown in the list.

If either the ODBC Driver Manager or the IBM DB2 CLI/ODBC driver is not installed then double-click on the **Install ODBC Driver** icon in the DB2 folder to install both the DB2 CLI/ODBC driver and the ODBC driver manager.

b Register the DB2 database with the ODBC driver manager as a *data source* using either of these methods:

- Using the CCA:

  1 Select the DB2 database alias that you want to add as a data source.

  2 Click on the **Properties** push button.

  3 Select the **Register this database for ODBC** check box.

- Using the Visigenic **ODBC Administration tool**, which you can access from the **ODBC** folder or by running **odbcadm.exe** from the command line:

  1 Click on the **Add** push button from the Data Sources window. The Add Data Source Window opens.

  2 Double-click on the IBM DB2 ODBC Driver in the list.

  3 Select the DB2 database to add and click on **OK**.

3 Optional: Configure the DB2 CLI/ODBC driver using the CCA:

a Select the DB2 database alias you want to configure.

b Click on the **Properties** push button. The Database Properties window opens.

c Click on the **Settings** push button. The CLI/ODBC Settings window opens.

d Click on the **Advanced** push button. You can set the configuration keywords in the window that appears. These keywords are associated with the database *alias name*, and affect all DB2 CLI/ODBC applications that access the database. The online help explains all of the keywords, as does an appendix in the *Installing and Configuring DB2 Clients* manual.

For information on manually editing this file (*db2cli.ini*), see “Configuring db2cli.ini” on page 203.

4 If you have installed ODBC access (as described above), you can now access DB2 data using ODBC applications. Start the ODBC application and go to the Open window. Select the **ODBC databases** file type. The DB2 databases that you added as ODBC data sources will be selectable from the list. Many ODBC applications will open the table as read-only unless a unique index exists.

If you require additional information at this point you can refer to the following topics in “Detailed Configuration Information” on page 201:

- “How to Bind the DB2 CLI/ODBC Driver to the Database” on page 202
- “How to Set CLI/ODBC Configuration Keywords” on page 202
- “Configuring db2cli.ini” on page 203

UNIX Client Access to DB2 using CLI/ODBC

Before DB2 CLI and ODBC applications can successfully access a DB2 database from a UNIX client, perform the following steps on the client system:

1 The DB2 database (and node if the database is remote) must be cataloged. To do so, use the command line processor.

For more information see Chapter 15, “Configuring Client-to-Server Communications Using the Command Line Processor” on page 147 or the **CATALOG DATABASE** and **CATALOG NODE** commands in the *Command Reference*.

2 The DB2 CLI/ODBC driver is automatically installed when DB2 is installed, and therefore nothing needs to be done.

3 If you are using ODBC applications to access DB2 data, perform the following steps. (If you are using only CLI applications, skip this step and go to the next step.)

a When using an ODBC application you must ensure that the Visigenic ODBC Driver Manager components are installed and that each user that will use ODBC has access to it.

- If DB2 installed the Driver Manager it would be located in the *sqllib/odbclib* subdirectory.
- If the Driver Manager was installed by another program, the file *.odbc.ini* (begins with a period) would be located in the root directory of the user ID that runs the ODBC application.

b The Driver Manager uses two initialization files. A sample template of these files are provided in the *sqllib/odbclib* subdirectory.

odbcinst.ini ODBC Driver Manager's configuration file indicating which database drivers are installed. Each user that will use ODBC must have access to this file.

.odbc.ini End-user's data source configuration. Each user has a separate copy of this file.

Setting up odbcinst.ini

The settings in this file impact all of the ODBC drivers on the machine.

Use an ASCII editor to update this file. It must have a stanza (section) called [IBM DB2 ODBC DRIVER], with a line starting with "Driver" indicating the full path to the DB2 ODBC driver *(db2.o)*. For example, if the home directory of your end user is */u/thisuser/* and the sqllib directory is installed there, then the correct entry would be:

```
[IBM DB2 ODBC DRIVER]
Driver=/u/thisuser/sqllib/lib/db2.o
```

See the sample file in the *sqllib/odbclib* subdirectory for an example.

Setting up odbc.ini

The settings in this file are associated with a particular user on the machine; different users can have different *odbc.ini* files.

The *odbc.ini* file must be copied into the end user's home directory and called *.odbc.ini*. Update this file, using an ASCII editor, to reflect the appropriate data source configuration information. To register a DB2 database as an ODBC data source there must be one stanza (section) for each DB2 database.

The sample *odbc.ini* template shows:

- line 2 (part of the [ODBC Data Source] stanza):

  ```
  SAMPLE=IBM DB2 ODBC DRIVER
  ```

  Indicates that there is a data source called SAMPLE that used the IBM DB2 ODBC DRIVER.

- lines 4-6 (part of the [SAMPLE] stanza):

  ```
  [SAMPLE]
  Driver=/u/thisuser/sqllib/lib/db2.o
  Description=Sample DB2 ODBC Database
  ```

  Indicates that the SAMPLE database is part of the DB2 instance located in the directory */u/thisuser*.

- line 11 (part of the [ODBC] stanza):

  ```
  InstallDir=/u/thisuser/sqllib/odbclib
  ```

  Indicates that */u/thisuser/sqllib/odbclib* should be treated as the location where ODBC is installed.

  Note: If your application specifically installed the ODBC components (such as the Driver Manager) elsewhere, you must update the line starting with InstallDir under the [ODBC] stanza to reflect this new location.

  For example, if ODBC has been installed in */opt/odbc*, the [ODBC] stanza would look like:

```
[ODBC]
Trace=0
TraceFile=odbctrace.out
InstallDir=/opt/odbc
```

See the sample file in the *sqllib/odbclib* subdirectory for an example. You can also see "How to Configure ODBC.INI" on page 203 for more detailed information.

Once the .ini files are set up you can run your ODBC application and access DB2 databases. Refer to the documentation that comes with your ODBC application for additional help and information.

4 Configure the DB2 CLI/ODBC driver (**optional**).

There are various keywords and values that can be used to modify the behavior of DB2 CLI/ODBC and the applications using it. The keywords are associated with the database *alias name*, and affect all DB2 CLI/ODBC applications that access the database.

For information on manually editing this file (*db2cli.ini*), see "Configuring db2cli.ini" on page 203. For information about the specific keywords see the *CLI Guide and Reference*.

If you require additional information at this point you can refer to the following topics in "Detailed Configuration Information" on page 201:

- "How to Bind the DB2 CLI/ODBC Driver to the Database" on page 202
- "How to Set CLI/ODBC Configuration Keywords" on page 202
- "Configuring db2cli.ini" on page 203

Macintosh Client Access to DB2 using CLI/ODBC

Before DB2 CLI and ODBC applications can successfully access a DB2 database from a Macintosh client, perform the following steps on the client system:

1 The DB2 database (and node if the database is remote) must be cataloged. To do so, use the command line processor.

For more information see "Configuring Database Connections" on page 243.

2 If you are using ODBC applications to access DB2 data, perform the following steps. (If you are using only CLI applications, skip to step #4.)

a An ODBC driver manager must be installed to handle the communications between the ODBC applications and the DB2 ODBC driver. DB2 does not supply a Macintosh ODBC driver manager, only the DB2 ODBC driver itself. If your ODBC application does not supply an ODBC driver manager then you can obtain the MacODBC driver manager directly from Apple.

b Register the DB2 database with the ODBC driver manager list of data sources (contained in the ODBC Preferences file) as a *data source* using either of these methods:

- Running **ODBC Setup** (in a 68K environment) or **ODBC Setup PPC** (in a PowerMacintosh environment) from the Control Panel
- Editing the initialization (ODBC Preferences or ODBC Preferences PPC) file directly; refer to the next step in this process for more information.

3 Configuring ODBC Preferences or ODBC Preferences PPC: The ODBC initialization file is used to record information such as the available drivers and data sources. See the documentation for your driver manager for procedures on updating this file.

The MacODBC driver manager uses the ODBC Preferences file or the ODBC Preferences PPC file to record information about the available drivers and data sources.

It is also possible to modify these files manually. Do not change any of the existing entries in the files.

a Use an ASCII editor to edit the *ODBC Preferences* file or the *ODBC Preferences PPC* file.

68K Environment
The following is a sample *ODBC Preferences* file:

```
[ODBC Data Sources]
GLOBALDB=IBM ODBC DB2 for Macintosh
SAMPLE=IBM ODBC DB2 for Macintosh

[GLOBALDB]
Driver=appl:ODBC$DB2DriverFunctionSet
Description=My GLOBAL database

[SAMPLE]
Driver=appl:ODBC$DB2DriverFunctionSet
Description=My SAMPLE database
```

PowerMacintosh Environment
The following is a sample *ODBC Preferences PPC*:

```
[ODBC Data Sources]
SAMPLE=IBM ODBC DB2 for PPC

[SAMPLE]
Driver=DB2ODBCDriverPPC
Description=My SAMPLE database
```

b The [ODBC Data Source] section lists the name of each available data source and the description of the associated driver.

For each data source listed in the [ODBC Data Source] section, there is a section that lists additional information about that data source. These are called the *Data Source Specification* sections.

Under the `[ODBC DATA SOURCE]` entry, add the following line:

68K Environment
database_alias=IBM ODBC DB2 for Macintosh

PowerMacintosh Environment
database_alias=IBM ODBC DB2 for PPC

where *database_alias* is the alias of the database cataloged in the database directory (the database name used by the Command Line Processor CONNECT TO statement).

c Add a new entry in the Data Source Specification section to associate the data source with the driver:

68K Environment

```
[database_alias]
Driver=appl:ODBC$DB2DriverFunctionSet
```

PowerMacintosh Environment

```
[database_alias]
Driver=DB2ODBCDriverPPC
```

Where *database_alias* is the alias of the database cataloged in the database directory, and listed under the Data Source Specification section.

4 The DB2 CLI/ODBC driver can be configured further by editing the *db2cli.ini* file. This file contains various keywords and values that can be used to modify the behavior of DB2 CLI and the applications using it. The keywords are associated with the database *alias name*, and affect all DB2 CLI applications that access the database. For a complete description of all the keywords and their usage, refer to an appendix in the *Installing and Configuring DB2 Clients* manual.

5 If you have installed ODBC access (as described above), you can now access DB2 data using ODBC applications. Start the ODBC application and go to the Open window. Select the **ODBC databases** file type. The DB2 databases that you added as ODBC data sources will be selectable from the list. Many ODBC applications will open the table as read-only unless a unique index exists.

Detailed Configuration Information

The section “Platform Specific Details for CLI/ODBC Access” on page 192 should provide you with all of the information you require. The following additional information is useful where DB2 tool support is not available, and for administrators who require more detailed information.

The following topics are covered in this section:

- “How to Bind the DB2 CLI/ODBC Driver to the Database” on page 202
- “How to Set CLI/ODBC Configuration Keywords” on page 202
- “Configuring db2cli.ini” on page 203

How to Bind the DB2 CLI/ODBC Driver to the Database

The CLI/ODBC driver will autobind on the first connection to the database, provided the user has the appropriate privilege or authorization. The administrator may want to perform the first connect or explicitly bind the required files.

See “Binding Database Utilities” on page 189 for more information.

How to Set CLI/ODBC Configuration Keywords

DB2 CLI can be configured further by using either the CCA or the DB2 Client Setup administration tool, whichever is applicable for your platform, or by manually editing the *db2cli.ini* file.

This file contains various keywords and values that can be used to modify the behavior of DB2 CLI and the applications using it. The keywords are associated with the database *alias name*, and affect all DB2 CLI and ODBC applications that access the database.

By default, the location of the CLI/ODBC configuration keyword file is as follows:

Platform:	**Location:**
OS/2	*sqllib* directory
Windows NT	*sqllib* directory
Windows 95	*sqllib* directory
Windows 3.1	*sqllib\win* directory
UNIX	*sqllib/cfg* directory of the database instance running the CLI/ODBC applications

The environment variable DB2CLIINIPATH can also be used to override the default and specify a different location for the file.

The configuration keywords enable you to:

- Configure general features such as data source name, user name, and password.
- Set options that will affect performance.
- Indicate query parameters such as wild card characters.
- Set patches or work-arounds for various ODBC applications.
- Set other, more specific features associated with the connection, such as code pages and IBM Graphic data types.

For a complete description of all the keywords and their usage, refer to the *Installing and Configuring DB2 Clients*.

Configuring db2cli.ini: The *db2cli.ini* initialization file is an ASCII file which stores values for the DB2 CLI configuration options. A sample file is shipped to help you get started. Refer to the *CLI Guide and Reference* for information on each keyword.

See “Platform Specific Details for CLI/ODBC Access” on page 192 for more information on how to modify this file on your platform.

How to Configure ODBC.INI

Microsoft's 16-bit ODBC Driver Manager and Visigenic's ODBC Driver Manager use the *odbc.ini* file to record information about the available drivers and data sources. Visigenic's ODBC Driver Manager also uses the *odbcinst.ini* file on UNIX platforms. Although the necessary files are updated automatically by the tools on most platforms, users of ODBC on UNIX platforms will have to edit them manually. The file *odbc.ini* (and *odbcinst.ini* where required) are located:

Platform:	**Location:**
Windows	*drive:\windows* (where *drive* is the drive where Windows is installed)
Win-OS/2	*drive:\os2\mdos\winos2* (where *drive* is the drive where OS/2 is installed)
UNIX	Home directory of user ID running ODBC application

It is also possible to modify this file manually. Do not change any of the existing entries in the file.

1 Use an ASCII editor to edit the *odbc.ini* file.

The following is an example odbc.ini file:

```
[ODBC Data Sources]
MS Access Databases=Access Data (*.mdb)

[MS Access Databases]
Driver=D:\WINDOWS\SYSTEM\simba.dll
FileType=RedISAM
SingleUser=False
UseSystemDB=False
```

The [ODBC Data Sources] section lists the name of each available data source and the description of the associated driver.

For each data source listed in the [ODBC Data Sources] section, there is a section that lists additional information about that data source. These are called the *Data Source Specification* sections.

2 Under the `[ODBC DATA SOURCE]` entry, add the following line:

```
database_alias=IBM DB2 ODBC DRIVER
```

where `database_alias` is the alias of the database cataloged in the database directory (the database name used by the command line processor CONNECT TO statement).

3 Add a new entry in the Data Source Specification section to associate the data source with the driver:

```
[database_alias]
Driver=drive:\windows\system\db2cliw.dll
```

Where:

- *database_alias* is the alias of the database cataloged in the database directory, and listed under the Data Source Specification section.
- *drive* is the drive where Windows is installed.

The following shows the example file with the IBM data source entries added:

```
[ODBC Data Sources]
MS Access Databases=Access Data (*.mdb)
SAMPLE=IBM DB2 ODBC DRIVER

[MS Access Databases]
Driver=D:\WINDOWS\SYSTEM\simba.dll
FileType=RedISAM
SingleUser=False
UseSystemDB=False

[SAMPLE]
Driver=D:\WINDOWS\SYSTEM\db2cliw.dll
Description=Sample DB2 Client/Server database
```

Note: If you are running the ODBC application under WIN-OS/2, specify the equivalent path *\OS2\MDOS\WINOS2\SYSTEM* in place of the *\WINDOWS\SYSTEM* path.

UNIX Configuration of .ini files

The section “UNIX Client Access to DB2 using CLI/ODBC” on page 197 contains detailed steps on how to update both the *odbc.ini* and *odbcinst.ini* files.

Running Java Programs

You can use DB2 Java Database Connectivity (JDBC) support to run the following types of Java programs:

- Java applications, which rely on the DB2 Client Application Enabler to connect to DB2.
- Java applets, that do not require any other DB2 component code on the client.

Java can also be used on the server to write user-defined functions, stored procedures, and table functions.

For further information on developing Java applications that access DB2 databases refer to the Web page located at http://www.software.ibm.com/data/db2/java/ and the *Road Map to DB2 Programming*.

Configuring the Environment

The following environment variables must be set to run Java programs:

OS/2, Windows 95 and Windows NT

- CLASSPATH includes "." and the file *sqllib\java\db2java.zip*
- PATH includes the directory *sqllib\bin*
- LIBPATH includes the directory *sqllib\dll* (OS/2 only)

UNIX

In the following settings *instance_name* is the home directory of the DB2 instance owner.

- CLASSPATH includes "." and the file *instance_name/sqllib/java/db2java.zip*
- PATH includes the directory *instance_name/sqllib/bin*
- LD_LIBRARY_PATH includes the directory *instance_name/sqllib/lib*

Java Applications

Start your application from the desktop or command line, like any other application. The DB2 JDBC driver handles the JDBC API calls from your application and uses the DB2 Client Application Enabler to communicate the requests to the server and receive the results.

Java Applets

Because Java applets are delivered over the Web, you treat them a bit differently than Java applications. You must install DB2 (server or client) on the same machine as your Web server, then install the Java Development Kit (JDK) Version 1.1 from Sun Microsystems on the server as well (refer to http://www.software.ibm.com/data/db2/java for details).

To run your applet, you need only a Java-enabled Web browser on the client machine. When you load your HTML page, the applet tag downloads the Java applet to your machine, which then downloads the Java class files, including the *COM.ibm.db2.java.sql* and *COM.ibm.db2.jdbc.net* classes and DB2's JDBC driver. When your applet calls the JDBC API to connect to DB2, the JDBC driver establishes separate communications with the DB2 database through the JDBC applet server residing on the DB2 server.

To run your applets, do the following:

1 Start the DB2 JDBC applet server on your Web server by entering:

```
db2jstrt portno
```

where *portno* is the number of the unused TCP/IP port that you specified in the *DB2Applt.java* file.

2 On your client system, start your Web browser and load the HTML file that imbeds your applet.

Chapter 20. Entering DB2 Commands and SQL Statements

You can use the Command Center to enter DB2 commands and SQL statements if you are working with DB2 on an OS/2, Windows 95, or Windows NT workstation. Otherwise, you can enter commands in the command line processor or at a system command prompt.

Task: Entering DB2 commands, SQL statements, and operating system commands	
Operating System	**Tools**
Macintosh	• command line processor
OS/2	• Command Center • command line processor • command prompt
UNIX	• command line processor • command line prompt
Windows 3.x	• command line processor
Windows 95 or Windows NT	• Command Center • command line processor • command window

Using the Command Center

On OS/2, Windows 95 and Windows NT operating systems, the Command Center provides an interactive window that allows you to:

- Run SQL statements, DB2 commands, and operating system commands.
- See the execution result of one or many SQL statements and DB2 commands in a result window. You can scroll through the results and save the output to a file.
- Save a sequence of SQL statements and DB2 commands to a script file. You can then schedule the script to run as a job. When a saved script is modified, all jobs dependent on the saved script inherit the new modified behavior.
- Recall and run a script file.
- See the execution plan and statistics associated with a SQL statement before execution. You do this by invoking Visual Explain in the interactive window.

- Get quick access to database administrative tools such as the Control Center and the Journal from the main tool bar.
- Display all the command scripts known to the system through the Script Center, with summary information listed for each.

To invoke the Command Center on OS/2:

1. Double-click the **DB2 for OS/2** folder.
2. Double-click on the **Command Center** icon.

To invoke the Command Center on Windows 95 and Windows NT:

1. Click on **Start**.
2. Select **Programs**->**DB2 for Windows**->**Command Center**.

The Command Center contains a large input area in which you type your commands. To run the commands you have entered, select the **Start or stop execution** icon from the menu bar, or use CTRL+Enter.

Do not type the commands with a `db2` prefix; instead, enter the command as follows:

```
   list node directory;
```

If you want to run multiple commands, you must end each command with a semicolon, then press Enter to start the next command on a new line.

After you run a command, the Command Center displays the Results window, which informs you how the commands are proceeding.

To recall commands that you have typed in interactive mode, click on the arrow beside the narrow input window. To save commands entered in interactive mode as scripts, select **Script**->**Save as** from the menu bar.

If you want to store commonly used SQL statements or DB2 commands as scripts, click on the **Script** icon from the main tool bar.

Using the Command Line Processor

You can use the command line processor to enter SQL statements and DB2 commands. It operates in the following modes:

Interactive Input Mode
You can enter DB2 commands or SQL statements and view their output.

Command Line Mode
The DB2 command line processor behaves like a command window from your operating system. You can enter DB2 commands or SQL statements and view their output.

File Input Mode
Refer to the *Command Reference* for information on the file input mode.

Interactive Input Mode

To invoke the command line processor in interactive input mode, do the following:

- On OS/2:
  1. Double-click the **DB2 for OS/2** folder.
  2. Double-click on the **Command Line Processor** icon.
- On Windows 95 and Windows NT operating systems:
  1. Click on **Start**.
  2. Select **Programs**->**DB2 for Windows**->**Command Line Processor**.
- On Windows 3.x use one of the following methods:
  – From a DOS full screen, enter the **win db2clpw** command.
  – From Windows:
    1. Double-click the **DB2** folder.
    2. Double-click on the **Command Line Processor** icon.
- On Macintosh:
  1. Double-click the **DB2** folder.
  2. Double-click on the **Command Line Processor** icon.

You can also invoke the command line processor in interactive input mode, by entering the following command at a command prompt:

- On OS/2 and UNIX, enter **db2**.
- On Windows 95 and Windows NT, enter **db2cmd**, then enter **db2**.

In interactive input mode, the prompt looks like this:

```
db2 =>
```

This prompt indicates that you do not type DB2 commands with a db2 prefix; instead, you just enter the DB2 command. For example:

```
list node directory
```

To run operating system commands in interactive mode, precede the operating-system command with an exclamation mark (!). For example:

```
!dir db2*.log
```

Windows 3.x does not support this method of running operating system commands.

If you need to enter a long command that does not fit on one line, use the line continuation character, \. For example:

```
db2 => select firstname, lastname, dateofbirth, from \
staff where height=68, order by \
lastname
```

Note: You do not need to use a line continuation character when entering long commands in interactive input mode on Windows 3.x workstations.

To end interactive input mode, type **terminate**.

Command Line Mode

On UNIX, you can issue DB2 commands from a command prompt. You must include the db2 prefix. For example:

```
db2 list node directory
```

To invoke the command line processor in command line mode on Windows 95 and Windows NT operating systems, use one of the following methods:

- Click on **Start** and select **Programs**->**DB2 for Windows**->**Command Window**.
- Enter the **db2cmd** command at a command prompt.

After you invoke the DB2 command environment, you can enter DB2 commands at the command prompt. You must include the db2 prefix.

Notes:

1. If the DB2 command contains characters that have special meanings on the operating system you are using, you will need to enter the command in quotation marks to ensure that it is run properly. For example, the following command will successfully retrieve all information from the *employee* table, even if the * character has a special meaning on the operating system:

   ```
   db2 "select * from employee"
   ```

2. Command line mode is not available on Windows 3.x systems.

For further information on using the command line processor, refer to the *Command Reference*.

Connecting to a Logical Node

The *default logical node* is one that has a value of 0 for the *logical_port* parameter in the *db2nodes.cfg* file. By default, a client connects to the default logical node unless the *DB2NODE* environment variable is set to another node number. For all commands and applications that connect to a database, you must ensure that the *DB2NODE* environment variable is set to reference the node number of the logical node that you want the operation to run on. The node number of the logical node is in the *db2nodes.cfg* file. If the database partition server being connected to is not started on the machine, the application returns an error. In DB2 Extended Enterprise Edition, the **set client** command can also be used to target a specific node within a partitioned database system. For more information, refer to the *Command Reference*.

Remote applications always connect to the default node, unless the *DB2NODE* environment variable is set to another node number. Because all logical nodes on an instance share the same database manager configuration file, only one logical node per instance on a machine can service remote applications. The default node starts up the communication subsystem, if it was configured to do so. If the default node is shut down, the communication subsystem also ends. You will have to stop and restart a logical node on the machine to re-establish the communication subsystem. If you move the default node to another machine, on which logical nodes for that instance are already running, and one of the logical nodes already has a communication subsystem for remote clients, the restarted node cannot have one.

As an example, assume that you want to update the configuration of a database partition server that belongs to a database called `TEST`, and that database partition server is on a machine called MACH1. Also assume that you defined nodes 0 and 1 on MACH1. Because database configuration is defined at the database partition level, you would connect to each logical node on the machine and change the database configuration. You could do the following in two sessions on MACH1:

1. In session 1:

   a. Export *DB2NODE*=0.

   b. Connect to the database partition server on logical node 0. Issue the following statement:

      ```
      db2 connect to test
      ```

   c. Update the database configuration file for the remote server.

   d. End the connection by issuing the following statement:

      ```
      db2 connect reset
      db2 terminate
      ```

2. In session 2:

   a. Export DB2NODE=1.

   b. Connect to the database partition server on logical node 1. Issue the following statement:

      ```
      db2 connect to test
      ```

   c. Update the database configuration file for the remote server.

   d. End the connection by issuing the following statement:

      ```
      db2 connect reset
      db2 terminate
      ```

If you want, you can run session 1 and 2, one after the other.

When you are connected to a database, you can list the local database directory by issuing the **list database directory on** *path* command. The output will show the catalog node number for the database and the current node number where you are connected.

Chapter 21. Controlling Your DB2 Environment

Registry values, environment variables, and configuration parameters control your database environment.

- To set registry values through the DB2 profile registry, see "Controlling the DB2 Profile Registry" on page 215.
- To set DB2 environment variables, see:
  - "Setting Environment Variables" on page 214.
  - "Setting Your Environment on OS/2" on page 216.
  - "Setting Your Environment on Windows 95 and Windows NT" on page 217.
  - "Setting Your Environment on UNIX Systems" on page 218.
  - "Setting Your Environment on Windows 3.x" on page 219.
- To set database manager configuration parameters on a client instance, see "Configuration Parameters" on page 225.

Prior to the introduction of the DB2 profile registry in Version 5, changing your DB2 environment on OS/2, Windows 95 or Windows NT workstations required you to change your system's environment variables. On UNIX platforms, changing your DB2 environment required you to change the *sqllib/db2profile* file in the home directory of the instance.

With Version 5, almost all of the environment variables have been moved to the DB2 profile registry. Users with system administrative (SYSADM) authority for a given instance can update registry values for that instance. Use the **db2set** command to update DB2 registry values without rebooting your system. The DB2 registry applies the updated information to DB2 server instances and DB2 applications started after the changes are made.

The *DB2PATH* and *DB2INSTANCE* environment variables must be specified in the *db2profile* script file. Other environment variables can be set in the registry. Some environment variables cannot be set in the registry; for example *DB2NODE*. This environment variable should be set in the shell environment. For more information on environment variables and where they should be set, see "Setting Environment Variables" on page 214.

Note: The concept of the DB2 profile registry is not supported on Windows 3.x. See "Setting Your Environment on Windows 3.x" on page 219 for more information.

DB2 configures its operating parameters by checking for variable values according to the following search order:

- The environment variable settings, or a variable set in the db2profile or the db2cshrc files.

- Profile registry values set with the **db2set** command in the node-level profile.
- Profile registry values set with the **db2set** command in the instance-level profile.
- Profile registry values set with the **db2set** command in the global-level profile.

See “DB2 Registry Values and Environment Variables” on page 220 for descriptions of the subset of registry values and environment variables that you may want to adjust to get DB2 up and running. For more detailed information about all registry values and environment variables, refer to the *Administration Guide*.

Setting Environment Variables

The *db2profile* script file lists all the environment variables that are used to control the environment for the instance. In the file, each environment variable is preceded by a brief description of its purpose, and a range of possible values where appropriate. For example, the script sets the *DB2INSTANCE* and *PATH* environment variables as follows:

DB2INSTANCE Same as `$USER` for the instance owner

PATH *INSTHOME/sqllib/adm*, *INSTHOME/sqllib/bin*, and *INSTHOME/sqllib/misc*

INSTHOME is the home directory of the `db2instc`, who is defined to be the instance owner.

For additional information, read the description of each environment variable listed in the file.

The instance owner or a SYSADM user can customize the script for the instance. All users in the instance (that is, all users who can access the *INSTHOME/sqllib* directory for the instance) must call the same script that is executed when the **db2start** command is issued. The script can be invoked directly or added to a user's *.profile* file.

You can code the *db2profile*, or the *db2cshrc* script files following the Korn, Bourne, or C shell syntax.

You can use the *db2options* registry value to set options for the command line processor. For more information, refer to the *Command Reference*.

Controlling the DB2 Profile Registry

The DB2 profile registry stores DB2 registry values. The "levels" of registry values are as follows:

DB2 node-level profile:
This profile is only used with DB2 Universal Database Extended Edition. It contains node-level variable settings and overrides. Values defined in this level will override their settings in the instance and global levels.

DB2 instance-level profile:
This profile contains instance level variable settings and overrides. Values defined in this level will override their settings in the global level.

DB2 global-level profile:
This profile contains machine global variable settings. Any variable not defined at the node or instance levels will be evaluated at this level. (In DB2 Extended Enterprise Edition, one global-level profile exists at each machine.)

To modify registry variable values, use the **db2set** command. The syntax of the **db2set** command is as follows:

- To set a parameter for a specific DB2 Extended Enterprise Edition node, enter:

  ```
  db2set parameter=value -i instance_name node_number
  ```

- To set a parameter for the current instance, enter:

  ```
  db2set parameter=value
  ```

- To set a parameter's value for a specific instance, enter:

  ```
  db2set parameter=value -i instance_name
  ```

- To set a parameter at the global profile level, enter:

  ```
  db2set parameter=value -g
  ```

  Notes:

  1. Some parameters will always default to the global level profile. They cannot be set at the instance or node level profiles; for example, *DB2SYSTEM* and *DB2INSTDEF*.
  2. On UNIX, you must have system administrative (SYSADM) authority to change registry values for an instance. Only users with root authority can change parameters in global-level registries.

- To delete a parameter's value at a specified level, you can use the same command syntax to set the parameter but specify nothing for the parameter value. For example, to delete the parameter's setting at the node level, enter:

  ```
  db2set parameter= -i instance-name node-number
  ```

- To delete the parameter's setting at the instance level and prevent use of the variable at the global level, enter:

  ```
  db2set parameter= -null -i instance-name
  ```

  The `-null` option allows you to set the value of a parameter globally, but unset the value for a specific instance. For example, you can set *DB2COMM* to `tcpip` at the global level, and if you have four instances on your system, you can set three of the instances to default to the global setting, but use the -null option to unset *DB2COMM* on the fourth instance.

- To view the current session's parameter's value, enter:

  ```
  db2set parameter
  ```

- To view the parameter's value at all levels, enter:

  ```
  db2set parameter -all
  ```

- To view a list of all values defined in the profile registry, enter:

  ```
  db2set -all
  ```

For more information on the **db2set** command and remote administering of the DB2 profile registry, refer to the *Command Reference*.

Setting Your Environment on OS/2

It is strongly recommended that all DB2 specific registry values be defined in the DB2 profile registry. If DB2 variables are set outside of the registry, remote administration of those variables will not be possible, and the workstation will have to be rebooted in order for the variable values to take effect.

On OS/2, there are three system environment variables that are not stored in the DB2 profile registry: *DB2INSTANCE*, *DB2PATH*, and *DB2INSTPROF*. *DB2INSTANCE* and *DB2PATH* are set when DB2 is installed; *DB2INSTPROF* can be set after installation. The environment variable *DB2PATH* must be set; normally it is set during install and you should not modify it. Setting *DB2INSTANCE* and *DB2INSTPROF* is optional. Because the system environment variables are not set in the profile registry, you will need to reboot if you change their settings.

To change the setting of an environment variable, enter the following command:

```
set parameter=value
```

To determine the setting of an environment variable, enter:

```
echo %variable-name%
```

To modify system environment variables you must edit the config.sys file, then reboot to make the changes take effect.

Setting Your Environment on Windows 95 and Windows NT

It is strongly recommended that all DB2 specific registry values be defined in the DB2 profile registry. If DB2 variables are set outside of the registry, remote administration of those variables will not be possible, and the workstation will have to be rebooted in order for the variable values to take effect.

Windows 95 and Windows NT have one system environment variable, *DB2INSTANCE*, that can only be set outside the profile registry; however, you are not required to set *DB2INSTANCE*. The DB2 profile registry variable *db2instdef* may be set in the global level profile to specify the instance name to use if *DB2INSTANCE* is not defined.

To determine the setting of an environment variable, enter:

```
echo %variable-name%
```

To set system environment variables, do the following:

On Windows 95:
Edit the *autoexec.bat* file and reboot to make the change take effect.

On Windows NT:
Set environment variables as follows:

1. Click on **Start** and select **Settings**->**Control Panel**.
2. Double-click on the **System** icon.
3. In the System Properties window, select the **Environment** tab and do the following:
   - **a** If the variable does not exist:
     1. Select any environment variable in the System Variables window.
     2. Change the name in the **Variable** field to the name of the environment variable you want to set, for example *DB2INSTANCE*.
     3. Change the **Value** field to the instance name, for example, db2inst.
   - **b** If the variable already exists in the System Variables window, you can set a new value:
     1. Select the environment variable you want to append, for example *DB2INSTANCE*.
     2. Change the **Value** field to the instance name, for example, db2inst.
   - **c** Click on the **Set** push button.

d Click on **OK**.

e You may have to reboot your system for these changes to take effect.

Note: The environment variable *DB2INSTANCE* can also be set at the session (process) level. For example, if you want to start a second DB2 instance called `TEST`, issue the following commands in a command window.

```
set db2instance=TEST
db2start
```

Setting Your Environment on UNIX Systems

It is strongly recommended that all DB2 specific registry values be defined in the DB2 profile registry. If DB2 variables are set outside of the registry, remote administration of those variables will not be possible.

On UNIX, you must set the system environment variable *DB2INSTANCE*.

Note: Except for *PATH* and *DB2INSTANCE*, all other DB2-supported variables must be set in the DB2 profile registry. To set variables that are not supported by DB2, define them in your *db2profile* file.

An instance owner or SYSADM user may customize these scripts for all users of an instance. Alternatively, you can copy and customize a script, then invoke a script directly or add it to your *.profile* or *.login* file.

To work with another instance for the current session, execute the *db2profile* (for Bourne shell or Korn shell) script or the db2cshrc (for C shell) script in the *sqllib* directory of the instance's home directory.

Issue the **which db2** command to ensure that your search path has been set up correctly. This command returns the absolute path of the DB2 CLP executable. Verify that it is located under the instance's *sqllib* directory.

Setting Your Environment on Windows 3.x

The DB2 environment on Windows 3.x is not controlled by profile registries. Instead, Windows 3.x clients define environment keywords in the file *db2.ini* (typically found in *C:\windows* directory).

> On Windows 3.x, the parameters that control the DB2 environment are called environment keywords. However, because many Windows 3.x keywords are also used on operating systems that use the DB2 profile registries, environment keywords may also be referred to as registry values in this manual.

The *db2.ini* initialization file is an ASCII file that stores values for the Windows 3.x client environment keywords. Within the file, there is just one section header titled:

```
[DB2 Client Application Enabler]
```

The parameters are set by specifying a keyword with its associated keyword value in the form:

```
   Keyword=keywordValue
```

Notes:

1. All the keywords and their associated values must be located below the section header.
2. The keywords are not case sensitive; however, their values can be if the values are character based.
3. Comment lines use a semicolon in the first position of a new line.
4. Blank lines are permitted. If duplicate entries for a keyword exist, the first entry is used (and no warning is given).

The following is a sample *db2.ini* file:

```
   [DB2 Client Application Enabler]
   DB2PATH=C:\SQLLIB\WIN
   DB2INSTANCE=DB2
   DB2INSTPROF=C:\SQLLIB
   DB2TRACEON=N
```

This file is located in the Windows product directory.

On Windows 3.x, the Client Application Enabler for DB2 Version 2.x and Version 5 must set this information *only* in the *db2.ini* file.

DB2 Registry Values and Environment Variables

The following subset of the DB2 registry values and environment variables are those that you may need to know about to get up and running. Each has a brief description; some may not apply to your environment.

You can view a list of all supported variables for your platform. On all operating systems except Windows 3.1, enter the following db2set command:

```
   db2set -lr
```

Note: Windows 3.1 clients define environment variables in the file *db2.ini*. See "Setting Your Environment on Windows 3.x" on page 219 for details.

Parameter	**Operating System**	**Values**	**Description**
		General	
DB2ACCOUNT	All	Default: (not set)	The accounting string that is sent to the remote host. Refer to the *DB2 Connect User's Guide* for details.
DB2CODEPAGE	All	Default: derived from the language ID, as specified by the operating system.	Specifies the code page of the data presented to DB2 for database client application. The user should not set *db2codepage* unless explicitly stated in DB2 documents, or asked to do so by DB2 service. Setting *db2codepage* to a value not supported by the operating system can produce unexpected results. Normally, you do not need to set *db2codepage* because DB2 automatically derives the code page information from the operating system.
DB2COUNTRY	All	Default: derived from the language ID, as specified by the operating system.	Specifies the country code of the client application, which influences date and time formats.
DB2DBDFT	All	Default: (not set)	Specifies the database alias name of the database that will be implicitly connected to when applications are started and no implicit connect has been done. This keyword is ignored if it is not set.
DB2DISCOVERYTIME	OS/2, Windows 95, and Windows NT	Default=40 seconds, minimum=20 seconds	Specifies the amount of time that SEARCH discovery will search for DB2 systems.

Parameter	Operating System	Values	Description
DB2ENVLIST	UNIX	Default: (not set)	Lists specific variable names for either stored procedures or user defined functions. By default, the **db2start** command filters out all user environment variables except those prefixed with **DB2** or **db2**. If specific environment variables must be passed to either stored procedures or user defined functions, you can list the variable names in the *db2envlist* environment variable. Separate each variable name by one or more spaces. DB2 constructs its own PATH and LIBPATH, so if PATH or LIBPATH is specified in *db2envlist*, the actual value of the variable name is appended to the end of the DB2-constructed value. For example, LIBPATH becomes DB2LIBPATH.
DB2INCLUDE	All	Default=current directory	Specifies a path to be used during the processing of the SQL INCLUDE text-file statement during DB2 PREP processing. It provides a list of directories where the INCLUDE file might be found. Refer to the *Embedded SQL Programming Guide* for descriptions of how *db2include* is used in the different precompiled languages.
DB2INSTDEF	OS/2, Windows 95 and Windows NT	Default=DB2 (on OS/2, Windows 95 and Windows NT)	Sets the value to be used if *DB2INSTANCE* is not defined.
DB2LIBPATH	UNIX	Default: (not set)	Specifies the value of LIBPATH in the *db2libpath* environment variable. On UNIX operating systems, the value of LIBPATH cannot be inherited between parent and child processes if the user ID has changed. Since the **db2start** executable is owned by root and its execute permissions are setuid to root, DB2 cannot inherit the LIBPATH settings of end users. If you list the variable name, LIBPATH, in the *db2envlist* environment variable, you must also specify the value of LIBPATH in the *db2libpath* registry value. The **db2start** executable will then read the value of *db2libpath* and append this value to the end of the DB2-constructed LIBPATH.
DB2NTMEMSIZE	Windows NT	Default=(not set)	Used to override the upper limit of the DB2 shared memory segment.

Parameter	Operating System	Values	Description
DB2OPTIONS	All except Windows 3.1 and Macintosh	Default=null	Sets command line processor options.
DB2SLOGON	Windows 3.1	Default: NO	Enables a secure logon in DB2 for Windows 3.1. If *db2slogon*=YES DB2 does not write user IDs and passwords to a file, but instead uses a segment of memory to maintain them. When *db2slogon* is enabled, the user must logon each time Windows 3.1 is started.
DB2TIMEOUT	Windows 3.1 and Macintosh	Default: (not set)	Used to control the timeout period for Windows 3.1 and Macintosh clients during long SQL queries. After the timeout period has expired a dialog box pops up asking if the query should be interrupted or allowed to continue. The minimum value for this variable is 30 seconds. If *db2timeout* is set to a value between 1 and 30, the default minimum value will be used. If *db2timeout* is set to a value of 0, or a negative value, the timeout feature is disabled. This feature is disabled by default.
DB2TRACENAME	Windows 3.1 and Macintosh	Windows 3.1 Default=DB2WIN.TRC Macintosh Default=DB2MAC.TRC	On Windows 3.1 and Macintosh, specifies the name of the file where trace information is stored. The default is on Windows 3.1 is *db2tracename*=DB2WIN.TRC, and is saved in your current instance directory (for example, *\sqllib\db2*). The default on Macintosh is *db2tracename*=DB2MAC.TRC. We strongly recommend that you specify the full path name when naming the trace file.
DB2TRACEON	Windows 3.1 and Macintosh	Default=N, values: Y, N	On Windows 3.1 and Macintosh, turns trace on to provide information to IBM in case of a problem. (It is not recommended that you turn trace on unless you encounter a problem you cannot resolve.) Refer to the *Troubleshooting Guide* for information on using the trace facility with DB2 Client Application Enabler.

Parameter	Operating System	Values	Description
DB2TRCFLUSH	Windows 3.1 and Macintosh	Default=N, values: Y, N	On Windows 3.1 and Macintosh, *db2trcflush* can be used in conjunction with *db2traceon*=Y. *db2trcflush*=Y will cause each trace record to be written immediately into the trace file. Setting this will slow down your DB2 system considerably, so the default setting is *db2trcflush*=N. This setting is useful in cases where an application hangs the system and therefore needs to be rebooted. Setting this keyword guarantees that the trace file and trace entries are not lost by the reboot.
DB2TRCSYSERR		Default=1, values: 1 - 32767	Specifies the number of system errors to trace before the client turns off tracing. The default value traces one system error, after which, trace is turned off.
DB2YIELD	Windows 3.1	Default=N, values: Y, N	Specifies the behavior of the Windows 3.1 client while communicating with a remote server. When N is set, the client will not yield the CPU to other Windows 3.1 applications, and the Windows environment is halted while the client application is communicating with the remote server. You must wait for the communications operation to complete before you can resume any other tasks. When set to Y, your system functions as normal. It is recommended that you try to run your application with *db2yield*=YES. If your system crashes, you will need to set *db2yield*=NO. For application development, ensure your application is written to accept and handle Windows messages while waiting for a communications operation to complete.
System Environment			
DB2INSTANCE	All	Default=*db2instdef* on OS/2, Windows 95, and Windows NT	The environment variable used to specify the instance that is active by default. On UNIX, users must set the environment variable *DB2INSTANCE*.
DB2INSTPROF	OS/2, Windows 95, and Windows NT		The environment variable used to specify the location of the instance directory on OS/2, Windows 95 and Windows NT operating systems, if different than *DB2PATH*.

Parameter	Operating System	Values	Description
DB2PATH	OS/2, Windows 95 and Windows NT		The environment variable used to specify the directory where the product is installed on OS/2, Windows 95 and Windows NT operating systems.
Communications			
DB2COMM	All, server only	Default= null, values: any combination of APPC,IPXSPX, TCPIP	Specifies the communication managers that are started when the database manager is started. If this is not set, no DB2 communications managers are started at the server.
DCE Directories			
DB2DIRPATHNAME	OS/2 and UNIX		Specifies a temporary override of the DIR_PATH_NAME parameter value in the database manager configuration file. If a directory server is used and the target of a CONNECT statement or ATTACH command is not explicitly cataloged, then the target is concatenated with *db2dirpathname* (if specified) to form the fully qualified DCE name. **Note:** The *db2dirpathname* value has no effect on the instance's global name, which is always identified by the database manager configuration parameters DIR_PATH_NAME and DIR_OBJ_NAME.
DB2CLIENTCOMM	OS/2 and UNIX		Specifies a temporary override of the DFT_CLIENT_COMM parameter value in the database manager configuration file. If both DFT_CLIENT_COMM and *db2clientcomm* are not specified, then the first protocol found in the object is used. If either one or both of them are specified, then only the first matching protocol will be used. In either case, no retry is attempted if the first connect fails.
DB2ROUTE	OS/2 and UNIX	Default	Specifies the name of the Routing Information Object the client uses when it connects to a database with a different database protocol. The *db2route* value overrides the ROUTE_OBJ_NAME parameter value in the database manager configuration file.

Note: There are three registry values that are related to ADSM:

- *DSMI_CONFIG*
- *DSMI_DIR*
- *DSMI_LOG*

Refer to the *Administration Guide* for information about ADSM and these values.

Configuration Parameters

Configuration parameters are values that affect the operating characteristics of a database or database management system.

Database manager configuration parameters exist on servers and clients; however, only certain database manager configuration parameters can be set on the client. These parameters are a subset of the database management configuration parameters that can be set on the server. In DB2 Extended Enterprise Edition, one database manager configuration file is shared between all database partition servers in the instance.

Database configuration parameters can be set only on the server, or remotely from the Control Center. In DB2 Extended Enterprise Edition, each database partition has its own configuration file.

For information on setting database or database manager configuration parameters, refer to the *Administration Guide*.

Setting Database Manager Configuration Parameters on a Client Instance

To control database management configuration parameters on a client instance on OS/2, Windows 95, or Windows NT, use the Client Configuration Assistant (CCA).

- On OS/2:

  1. Double-click on the **Client Configuration Assistant** icon in the **DB2 for OS/2** folder.
  2. Click on the **Client Settings** push button.

- On Windows 95 and Windows NT:

  1. Click on **Start** and select **Programs**->**DB2 for Windows**->**Client Configuration Assistant**. The Client Configuration window opens.

2 Click on the **Client Settings** push button.

Follow the hints and online help to modify configuration parameter values.

To control database management configuration parameters on a client instance on UNIX, use the command line processor.

- For information on using Client Configuration Assistant, see Chapter 24, “Administering Clients Using the CCA” on page 243.
- For information on using the command line processor, see “Using the Command Line Processor” on page 209.

Part 6. Advanced Configuration and Deinstallation

Chapter 22. Configuring Server Communications Using the Control Center

The information in this section describes how to use the Control Center to configure communications on the server.

To configure an Administration Server for communications, go to Chapter 23, “Setting up Communications on the Server Using the Command Line Processor” on page 231.

The Control Center is a graphical tool available on OS/2, Windows 95, and Windows NT. Use the Control Center's setup communications function to configure communications on the server. The Control Center allows you to:

- Display the protocols and configuration parameters that a server instance is configured to use.
- Maintain the configured protocols:
  - You can modify the parameter values of a configured protocol.
  - You can add or delete a protocol.

    When you add support for a new protocol to the server system, the setup communications function detects and generates parameter values for the new protocol; you can accept or modify them before use. When you remove support for an existing protocol from the server system, the setup communications function detects the protocol that has been removed and disables its use by the instance.

    You can add a protocol that has not been detected, however, you must supply all parameter values before you proceed.

The setup communications function can be used to maintain communications of both local and remote server instances.

Modifying the DB2 Communication Configuration of Server Instances

1. To start the Control Center on an OS/2 system, double-click on the **Control Center** icon in the **Administration Tools** folder. (The **Administration Tools** folder is located in the DB2 folder.) To start the Control Center on a Windows NT system, click on the **Start** button and select **Programs**->**DB2 for Windows NT**->**Administration Tools**->**Control Center**.

2. Click on the **[+]** sign beside the **Systems** icon in the Control Center to get a list of systems.

3. Click on the **[+]** sign beside the system name to get a list of that system's database instances.

4. Select the instance you want to configure and click on the right mouse button.

5. Select the **Setup communications** option from the pop-up list. The Setup Communications window opens.

6. Use the Setup Communications window to configure communication protocols for the instance. Click on the **Help** push button for more information.

7. You must stop and start the instance for these changes to take effect.

   a. To stop the database manager instance, select the instance and click with the right mouse button on the instance and select the **Stop** option from the pop-up menu.

   b. To start the database manager instance, select the instance and click with the right mouse button on the instance and select the **Start** option from the pop-up menu.

Modifying an instance's communications settings may require you to update the database connection catalogs on the client.

Chapter 23. Setting up Communications on the Server Using the Command Line Processor

This section describes how to configure your DB2 server to communicate with remote client workstations. Also described in this section are considerations when configuring the DB2 Administration Server for communications. The Control Center and the Discovery function of the Client Configuration Assistant are dependent on the Administration Server's protocol configuration.

Follow the instructions in this section if you deselected a detected communication protocol during the installation or, you have added a communication protocol to your network since running the DB2 setup program.

If you have installed the Control Center, you can use the Setup Communications function to configure an instance for communications; however, it cannot be used to set up communications for an Administration Server. The Control Center can be installed on OS/2 and Windows NT servers, or on OS/2, Windows 95, and Windows NT clients.

The Setup Communications function cannot update the APPC communications subsystem; however, it will update the required DB2 instance parameters on the server.

Go to Chapter 22, “Configuring Server Communications Using the Control Center” on page 229 for more information.

For instructions on using the command line processor, see Chapter 20, “Entering DB2 Commands and SQL Statements” on page 207. For a description of database naming rules, see Appendix F, “Naming Rules” on page 315.

Setting the DB2COMM Registry Value

Your DB2 server can support multiple communication protocols concurrently; however, you only need to enable the protocols that you want to use.

If you deselected a detected protocol during installation, or have added a communication protocol to your network since installation, you must update the *db2comm* registry value with the protocol that you now want to support.

The *db2comm* registry value determines which protocol's connection managers will be enabled when the database manager is started. You can set this value for multiple communication protocols by separating the keywords with commas.

For your DB2 server, *DB2COMM* can be any combination of the following keywords:

appc	starts APPC support
ipxspx	starts IPX/SPX support
tcpip	starts TCP/IP support

To avoid problems with the Control Center and the Client Configuration Assistant, ensure that *DB2COMM* is set in the DB2 registry. It is not recommended that you use any other mechanism to set the *DB2COMM* value.

To set the *db2comm* registry value for the current instance, enter the **db2set DB2COMM=***protocol_names* command. Change the *protocol_names* parameter to reflect those protocols that you want to start when the database manager is started.

For example, to set the database manager to start connection managers for the APPC and TCP/IP communication protocols, enter:

```
db2set DB2COMM=appc,tcpip
db2stop
db2start
```

If you are setting up communications for the Administration Server, use the **db2set** command as follows:

```
db2set DB2COMM=appc,tcpip -i DB2DAS00
db2admin stop
db2admin start
```

where:

DB2DAS00
The name of the Administration Server's instance. If you are not sure of the Administration Server name, enter the following command to view it:

```
db2set DB2ADMINSERVER
```

When the **db2start** command is reissued, connection managers for the protocols specified by the *db2comm* registry value are started.

If there are problems starting a protocol's connection managers, a warning message is displayed and the error messages are logged in the *db2diag.log* file, located in the *instance* directory.

Additionally, any failure encountered while executing the **db2start** command is logged to a time-stamped file in the log subdirectory within the instance directory. This file contains the **db2start** results from each database partition server in the instance.

If there is no error when the **db2start** command is run, these files are removed.

Refer to the *Troubleshooting Guide* for information on the *db2diag.log* file.

If *db2comm* is undefined or set to null, no protocol connection managers are started when the database manager is started.

For more information on setting DB2 parameters, see “Controlling the DB2 Profile Registry” on page 215.

You are now ready to configure the server workstation to use any of the following communication protocols:

- TCP/IP - see “Configuring TCP/IP on the Server”
- IPX/SPX - see “Configuring IPX/SPX on the Server” on page 237
- APPC - see “Configuring APPC on the Server” on page 240

Configuring TCP/IP on the Server

To access a remote server through TCP/IP, you must first have installed and configured communication software for both the client and DB2 server workstations. See “Software Requirements” on page 41 for the communication protocol requirements for your platform. See “Possible Client-to-DB2 Server Connectivity Scenarios” on page 47 for the supported communication protocols for your particular client and server.

Before completing the steps in this section, be sure that you have completed the instructions in “Setting the DB2COMM Registry Value” on page 231.

If you are setting up communications for the Administration Server, it was configured for TCP/IP (using the registered Port number *523*) when it was created. There are no further steps required to enable the Administration Server to use TCP/IP.

The following steps are required to set up TCP/IP communications:

1. Identify and record parameter values.
2. At the server:
   - **a** Update the *services* file.
   - **b** Update the database manager configuration file.

> Due to the characteristics of the TCP/IP protocol, the TCP/IP subsystem may not be immediately notified of the failure of a partner on another host. As a result, a client application accessing a remote DB2 server using TCP/IP, or the corresponding agent at the server, may sometimes appear to be hung. DB2 uses the TCP/IP SO_KEEPALIVE socket option to detect when there has been a failure and the TCP/IP connection has been broken.
>
> If you are experiencing problems with your TCP/IP connection, refer to the *Troubleshooting Guide* for information on this parameter and other common TCP/IP problems.

Step 1. Identify and Record Parameter Values

As you proceed through the configuration steps, complete the *Your Value* column in the following table. You can fill in some of the values before you start configuring this protocol.

Table 18. TCP/IP Values Required at the Server

Parameter	Explanation	Sample Value	Your Value

Connection Port
• Connection Service name (*svcename*)
• Port number/Protocol (port_number/tcp)
Interrupt Port
• Interrupt Service name
• Port number/Protocol (port_number + 1/tcp)
	Values required in the *services* file.
	The Connection and Interrupt Service names are arbitrary, but each must be unique within the *services* file; ensure that this number does not conflict with the port numbers used by the Fast Communication Manager (FCM).
	The Connection and Interrupt Ports must be unique within the *services* file, with the Interrupt Port number equal to the Connection Service Port number plus one.
		`server1`
		`3700/tcp`
		`server1i`
		`3701/tcp`	

Service name (*svcename*)	The name used to update the Service name (*svcename*) parameter in the database manager configuration file at the server. This value must be the same as the Connection Service name specified in the *services* file.	`server1`	

Step 2. Configure the Server

The following steps describe how to configure this protocol on the server. Replace the sample values with your values and record them on your worksheet.

A. Update the Services File

The TCP/IP *services* file specifies ports that the DB2 server will listen on for client requests. The *services* file must contain two entries for each database manager instance.

You need to update the *services* file and specify the ports that you want the DB2 server to listen on for incoming client requests. The first entry describes the Connection Port, and the second entry describes the Interrupt Port.

The *services* file is located in the */etc* directory.

If you are using Network Information Services (NIS) on your network, you must update the *services* file located on your master server. Also, ensure that you do not specify a port number that is being used by the Fast Communication Manager (FCM), or any other process.

The location of the *services* file can depend on the products you have installed on your system. See your TCP/IP documentation for more information.

Using a local text editor, add the Interrupt Port and Connection Port entries to the *services* file for TCP/IP support. For example:

```
server1     3700/tcp     # DB2 connection service port
server1i    3701/tcp     # DB2 interrupt connection service port
```

where:

server1 is the Connection Service name

server1i is the Interrupt Service name

3700 and 3701 are the port numbers for the Connection and Interrupt Ports

`tcp` is the communication protocol that you are using

B. Update the Database Manager Configuration File

You must update the database manager configuration file with the Service name (*svcename*) parameter.

To update the database manager configuration file, perform the following steps:

1 Log on to the system as a user with System Administrative (SYSADM) authority.

If you have trouble logging on to the system, see “Logging on to the System” on page 167.

2 Set up the instance environment and invoke the DB2 command line processor as follows:

a. Run the start up script as follows:

```
. INSTHOME/sqllib/db2profile    (for Bourne or Korn shell)
source INSTHOME/sqllib/db2cshrc (for C shell)
```

where INSTHOME is the home directory of the instance.

b. Start the DB2 command line processor by issuing the **db2** command.

3 Update the database manager configuration file with the Service name (*svcename*) parameter using the following commands in the command line processor:

```
update database manager configuration using svcename svcename
db2stop
db2start
```

For example, if the Connection Service name in the *services* file was entered as *server1*, use:

```
update database manager configuration using svcename server1
db2stop
db2start
```

The *svcename* used must match the Connection Service name specified in the *services* file.

After the database manager is stopped and started again, view the database manager configuration file to ensure that these changes have taken effect. Enter the following command in the command line processor:

```
get database manager configuration
```

Now that you have configured the server, you are ready to install a DB2 client. Go to Chapter 8, "Installing DB2 Clients" on page 111 for more information.

Configuring IPX/SPX on the Server

To access a remote database server through IPX/SPX, you must first have installed and configured communication software for both the client and DB2 server workstations. See "Software Requirements" on page 41 for the communication protocol requirements for your platform. See "Possible Client-to-DB2 Server Connectivity Scenarios" on page 47 for the supported communication protocols for your particular client and server.

Before completing the steps in this section, be sure that you have completed the instructions in "Setting the DB2COMM Registry Value" on page 231.

If you are setting up communications for the Administration Server, it was configured for IPX/SPX (using the registered Socket number *87A2*) when it was created. There are no further steps required to enable the Administration Server to use IPX/SPX.

Your DB2 server can be set up to support IPX/SPX client communications via Direct Addressing.

In Direct Addressing, the client connects to the DB2 server by directly specifying the IPX/SPX internetwork address of the server (bypassing the NetWare file server). Using this method, a NetWare file server is not required on the network. Configure DB2 servers using this addressing method if they will be accessed solely by clients using Direct Addressing.

The following steps are required to set up IPX/SPX communications:

1 Identify and record parameter values.

2 Configure the server:

  a Update the database manager configuration file.

Step 1. Identify and Record Parameter Values

As you proceed through the configuration steps, complete the *Your Value* column in the following table. You can fill in some of the values before you start configuring this protocol.

Table 19 (Page 1 of 2). IPX/SPX Values Required at the Server

Parameter	Explanation	Our Example	Your Value
File server name (*FILESERVER*)	A * value indicates that you are using Direct Addressing.	*	
DB2 server object name (*OBJECTNAME*)	A * value indicates that you are using Direct Addressing.	*	
Socket number (*IPX_SOCKET*)	Represents the connection end point in a DB2 server's internetwork address. It must be unique for all server instances, and all IPX/SPX applications running on the workstation. It is entered in hexadecimal format. DB2 has registered well known sockets with Novell in the range 0x879E to 0x87A1. If you run more than 4 instances on the server machine, you must prevent socket collisions for instances 5 and up by choosing a socket number that is not *0x0000*, in the dynamic socket range *0x4000* to *0x7FFF*, or in the range *0x8000* to *0x9100* (these are well known sockets that are registered to various applications). The maximum value for this parameter is *0xFFFF*.	879E (default)	

Table 19 (Page 2 of 2). IPX/SPX Values Required at the Server

Parameter	Explanation	Our Example	Your Value
IPX/SPX Internetwork Address	The internetwork address required when configuring a client to communicate with a server using Direct Addressing.	09212700.400011527745.879E	

Step 2. Configure the Server

The following steps describe how to configure this protocol on the server. Replace the sample values with your values and record them on your worksheet.

A. Update the Database Manager Configuration File

You must update the database manager configuration file with the File server (*FILESERVER*), DB2 server object name (*OBJECTNAME*), and Socket number (*IPX_SOCKET*) parameters.

To update the database manager configuration file, perform the following steps:

1 Log on to the system as a user with System Administrative (SYSADM) authority.

If you have trouble logging on to the system, see "Logging on to the System" on page 167.

2 Set up the instance environment and invoke the DB2 command line processor as follows:

a. Run the start up script as follows:

```
. INSTHOME/sqllib/db2profile    (for Bourne or Korn shell)
source INSTHOME/sqllib/db2cshrc (for C shell)
```

where INSTHOME is the home directory of the instance.

b. Start the DB2 command line processor by issuing the **db2** command.

3 In the database manager configuration file, update the File server (*FILESERVER*), DB2 server object name (*OBJECTNAME*), and Socket number (*IPX_SOCKET*) parameters using the following commands in the command line processor:

```
update dbm cfg using fileserver FILESERVER objectname OBJECTNAME ipx_socket IPX_SOCKET
db2stop
db2start
```

For example, if you have chosen a value of 879E for the Socket number (*IPX_SOCKET*) parameter, use:

```
update dbm cfg using fileserver * objectname * ipx_socket 879E
db2stop
db2start
```

After the database manager is stopped and started again, view the database manager configuration file to ensure that these changes have taken effect. Enter the following command in the command line processor:

```
get database manager configuration
```

> If you are planning to only support clients that use Direct Addressing, you will be required to enter the server's IPX/SPX internetwork address as the DB2 server object name when cataloging the node on the client.
>
> Determine the value for the *OBJECTNAME* parameter by issuing the **db2ipxad** command on the server. This command is located in the *sqllib/misc/ directory*.
>
> Make note of this output on your worksheet for use when you configure an IPX/SPX client using the command line processor.

If you are planning to support clients using only Direct Addressing, you are now finished the configuration and are ready to install a DB2 client. Go to Chapter 8, “Installing DB2 Clients” on page 111 for more information.

Configuring APPC on the Server

To access a remote server through APPC, you must first have installed and configured communication software for both the client and DB2 server workstations. See “Software Requirements” on page 41 for the communication protocol requirements for your platform. See “Possible Client-to-DB2 Server Connectivity Scenarios” on page 47 for the supported communication protocols for your particular client and server.

Before completing the steps in this section, be sure that you have completed the instructions in “Setting the DB2COMM Registry Value” on page 231.

The following steps are required to set up APPC communications:

1 Identify and record parameter values.

2 Configure the server:

  a Update the *database manager configuration* file.

  b Configure the APPC communications subsystem.

Step 1. Identify and Record Parameter Values

Before you proceed through the configuration steps, enter your value for the Transaction program name in the following table.

Table 20. APPC Values Required at the Server

Parameter	Description	Sample Value	Your Value
Transaction program name (*tpname*)	The Transaction program that the server listens on for client requests.	`nyserver`	

Step 2. Configure the Server

The following steps describe how to configure this protocol on the server. Replace the sample value with your value as recorded on your worksheet.

A. Update the Database Manager Configuration File

You must update the database manager configuration file with the transaction program name (*tpname*).

To update the database manager configuration file, perform the following steps:

1 Log on to the system as a user with System Administrative (SYSADM) authority.

If you have trouble logging on to the system, see “Logging on to the System” on page 167.

2 Set up the instance environment and invoke the DB2 command line processor as follows:

a. Run the start up script as follows:

```
. INSTHOME/sqllib/db2profile    (for Bourne or Korn shell)
source INSTHOME/sqllib/db2cshrc (for C shell)
```

where INSTHOME is the home directory of the instance.

b. Start the DB2 command line processor by issuing the **db2** command.

3 Update the database manager configuration file with the server's transaction program name (*tpname*) using the following commands in the command line processor:

```
update dbm cfg using tpname tpname
db2stop
db2start
```

For example, if the server's transaction program name (*tpname*) is *nyserver*, use:

```
update dbm cfg using tpname nyserver
db2stop
db2start
```

If you are configuring the Administration Server to use APPC, you must update the admin server configuration file. Use the following command:

```
update admin configuration using tpname nyserver
db2admin stop
db2admin start
```

B. Configure the APPC Communications Subsystem

To configure your DB2 server to accept remote clients using APPC, you need to update the APPC communications subsystem to support the Transaction program name (*tpname*) that the server will use.

Refer to the *DB2 Connect Enterprise Edition Quick Beginnings* manual for detailed instructions on how to configure your APPC communications subsystem to support communications from the Transaction program that you specified for the server.

Chapter 24. Administering Clients Using the CCA

Use the information in this section to use the Client Configuration Assistant to perform administration tasks on OS/2, Windows 95, and Windows NT workstations.

When you install the DB2 server, if your protocols are set up and configured, DB2 server communications are automatically configured to allow DB2 to accept requests from remote clients.

> The CCA can be installed on OS/2, Windows 95, and Windows NT workstations. If it is not installed on your system, you can rerun the setup program and select to install the CCA.

With the CCA, you can:

- Configure database connections that applications can use.
- Update or delete existing configured database connections.
- Display the information for existing configured connections.
- Test a connection to a database.
- Enable or disable database connections to be configured as CLI/ODBC data sources.
- Import or export client profiles which contain information for the setup of a client.
- Update client configuration settings.
- Discover remote databases (if enabled).
- Bind user applications and utilities to databases.

Configuring Database Connections

> To complete the steps in this section, you must be logged on to the local system as a user with System Administrative (SYSADM) authority on the instance.

You can configure database connections through the Add Database SmartGuide, by:

- Using a profile as a source of information to add database connections, see “Using an Access Profile” on page 244.
- Searching the network for databases, see “Searching the Network for Databases” on page 247.
- Adding database connections manually, see “Manually Configure a Connection to a DB2 Database” on page 252.

Each of these is covered in more detail in the material that follows.

Using an Access Profile

You can use either a Server profile or a Client profile to configure database connections on a client.

Server Profiles

Server profiles can be generated for a DB2 server. They contain information about instances on the server system, and databases within each instance. The information for each instance includes the protocol information required to set up a client to connect to databases in that instance.

To generate server profiles, use the Generate Access Profile function provided in the Control Center. When a profile is generated for a DB2 server system, it includes server instances that have the *discover_inst* configuration parameter and databases with the *discover_db* configuration parameter set to ENABLE. The *discover* parameter, for the Administration Server, must be set to either *SEARCH* or *KNOWN* to generate a profile for a server system.

For information on setting the *discover_inst*, *discover_db* and *discover* configuration parameters, see “Setting Discovery Parameters” on page 251.

Generating a Server Profile: To generate an access profile, perform the following steps:

1 Start the Control Center.

2 Click on the **[+]** sign beside the **Systems** icon to get a list of systems.

3 Select the system to be profiled and click on the right mouse button.

4 Select the **Generate access profile** option.

5 Select the path and type in a file name for the profile, then click on the **Generate** push button.

To process a server profile and add its databases to the client's connection configuration list, use the Client Configuration Assistant's Import or Add functions. Using the Add function is the preferred method.

Using a Server Profile on the Client: To add a database using the Add function:

1. Start the Client Configuration Assistant (CCA).
2. Click on the **Add** push button.
3. Select the **Use an Access Profile** radio button, and click on the **Next** push button.
4. Click on the **Browse** push button to select the server profile that you want to access, or enter the path and filename in the **File** field.
5. Select a database to be added.
6. If desired, you can select the **Alias** and **ODBC** tabs at the top of the panel to specify a database alias name for the database, or to select CLI/ODBC options for the database. This step is optional.
7. Click on the **Done** push button.
8. You can test the connection that has been added by clicking on the **Test Connection** push button.

Client Profiles

Information in Client profiles can be added to another client through the following:

- Using the Add function, which allows you to selectively process the database connection data in the client profile and add it to a new client. (This function invokes the Add Database SmartGuide.)
- Using the Import function, which allows you to selectively process database connection data, client configuration settings, and CLI/ODBC common parameters.

Note: This scenario assumes that the database connections configured on one client will be exported and used to set up one or more additional clients.

Exporting a Client Profile: Client profiles are generated from clients using the Export function of the CCA. Use the export function to copy the database information used by one client to other clients.

The information contained in a client profile is determined during the export process. Depending on the settings chosen, it can contain the existing client's:

- Database connection information (including CLI/ODBC settings).
- Client settings (including database manager configuration parameters).
- CLI/ODBC common parameters.
- Configuration data for the local APPC communications subsystem.

Export can be used to generate a customized profile that can be imported on another client to set it up initially, or to update it.

To export a profile from the client, configure the client for communications and do the following:

1. Enter the **db2cca admin** command at a command prompt to start the Client Configuration Assistant in administrator mode. (You must be an administrator on the local system to perform this command.)

> The Export function of the CCA is only available when the CCA is started in administrator mode. The CCA can be started permanently in administrator mode by modifying the **Client Configuration Assistant** icon to add the parameter *admin* to the startup of the CCA, or by issuing the **db2cca admin** command.

2. Click on the **Export** push button.
3. Select the databases to be exported from the **Available DB2 Databases** window, and add them to the **Databases to be exported** window.
4. Select the check boxes that correspond to the options that you want to set up for the target client:
   - **Client Settings**
   - **CLI/ODBC Common parameters**
   - **APPC local stack configuration**

   To customize settings, click on the appropriate **Customize** push button. The settings that you customize will only affect the profile to be exported, no changes will be made to your workstation. For more information, click on the **Help** push button.
5. Click on **OK**.
6. Enter a path and file name for the Client profile.

Importing a Client Profile:

> The Import function can be used to easily set up new client systems. The Import function works as follows:
>
> **Client Settings**
> Configuration parameters found in the export file replace existing ones on the client; all other parameters are set to default.
>
> **CLI/ODBC Common Parameters**
> Values found in the exported file replace those on the client.
>
> **Local Configuration Data for the APPC Communications Subsystem**
> If it is not already configured, this information is used to configure the APPC communications subsystem.

Perform these steps at the client that you want to set up. You can use this process to initially set up a new client, or to update an existing one.

1 Start the Client Configuration Assistant (CCA).

2 Click on the **Import** push button.

3 Select the path and filename of the client profile you want to import and click on **OK**.

4 The Import Client Profile window opens. Select the items you want to import. For more information, click on the **Help** push button.

5 Click on **OK**.

> If databases are contained in the client profile that you are importing, and you select to import them, the Add Database SmartGuide starts to allow you to selectively import the databases you want to connect to.

Searching the Network for Databases

Instead of entering protocol information to make a connection to remote database servers, you can use the CCA to find all the databases on your local network by following these steps:

> The following scenario assumes that messages used by the *Search* method of discovery are not filtered by your network, and that the installation defaults on the client and the server have not been changed. In order to run the following scenario, you must ensure that the *discover* parameter in the database manager configuration file on the client, and the Administration Server configuration file on the server are set to *Search*.

1 Start the Client Configuration Assistant (CCA).

2 Click on the **Add** push button on the CCA's main panel to start the Add Database SmartGuide.

3 Select the **Search the network** radio button, and click on the **Next** push button.

4 Click on the **[+]** sign beside the **Known Systems** icon to list all the systems known to your client.

a Click on the **[+]** sign beside the system to get a list of the instances and databases on it. Select the database that you want to add and proceed to Step 5.

b If the system that contains the database that you want is not listed, click on the **[+]** sign beside the **Other Systems (Search the network)** icon to search the network for additional systems. Click on the **[+]** sign beside the system to get a list of the instances and databases on it. Select the database that you want to add and proceed to Step 5.

c If the system you want is still not listed, it can be added to the list of systems by clicking on the **Add Systems** push button.

5 If desired, you can select the **Alias** and **ODBC** tabs at the top of the panel to specify a database alias name for the database, or to select CLI/ODBC options for the database. This step is optional.

6 Click on the **Done** push button.

7 You can test the connection that has been added by clicking on the **Test Connection** push button.

Searching the network can be customized to meet the needs of individual organizations. The material that follows provides details on this customization. Refer to the *Administration Guide* for more information on individual configuration parameters and profile registry values.

Choosing a Discovery Method

Network searching uses a DB2 facility called *Discovery* to obtain information from DB2 servers. This information is used to configure clients for database connections.

Note: Discovery cannot find information about pre-Version 5 DB2 systems.

Two discovery methods are available for searching the network:

- **Known**

  Known discovery allows you to discover instances and databases on systems that are known to your client, and add new systems so that their instances and databases can be discovered.

  Click on the **[+]** sign beside the **Known Systems** icon to get a list of known DB2 server systems. Click on the **[+]** sign beside the system to get a list of the instances and databases on it. Select the database that you want to add and complete the other panels in the Add Database SmartGuide.

  Initially, the list of systems will be blank; however, if you are running the CCA on the server, an entry for the local server will be shown. Add systems to the list by clicking on the **Add System** push button. To use this option you must know a few details about the Administration Server on the DB2 system to be searched:

  – A protocol configured and running on the Administration Server.
  – The protocol's configuration information.

  The Administration Server will listen for KNOWN discovery requests, from clients, on the protocols specified by the *db2comm* registry value in the Administration Server.

- **Search**

  This mode provides all of the facilities of Known discovery, and adds the option to allow your local network to be searched for DB2 servers.

  Searching does not require information about the Administration Server. When you click on the **[+]** sign beside the **Other Systems (Search the network)** icon, a list of DB2 server systems is displayed. Click on the **[+]** beside the system to get a list of the instances and databases on it. Select the database that you want to add and complete the other panels in the Add Database SmartGuide.

> *Search* may appear to be a simpler discovery method. However, in larger networks, network routers and bridges can filter the messages search uses to find DB2 servers on the network, resulting in an incomplete or even empty list. In this case, use the *Add System* method; its messages are not filtered by routers and bridges. If in doubt, contact your network administrator for assistance.

To have the server support Known discovery, set the *discover* parameter in the Administration Server to KNOWN. To have it support Search discovery, set this parameter to SEARCH. To prevent discovery of the server, and all of its instances and databases, set *discover* to DISABLE.

On the client, enabling discovery is also done using the *discover* parameter; however, in this case, the *discover* parameter is set in the client instance (or a server acting as a client) as follows:

KNOWN Allows the CCA to refresh systems in the known list, and to add new systems to the list by using the **Add Systems** button. When *discover* is set to KNOWN, the CCA will not be able to search the network.

SEARCH Enables all the facilities of the KNOWN setting, and enables network searching.

DISABLE Disables discovery. In this case, the **Search the network** option is not available in the Add Database SmartGuide.

> Administration Servers configured with the *discover* parameter set to KNOWN will not respond to search requests from clients. It is important that you consider this when changing the *discover* parameter.

Additional Settings for SEARCH Discovery

Search discovery requires that the configuration parameter *discover_comm* be set on both the server (in the Administration Server's configuration file) and the client (in the database manager configuration file).

The *discover_comm* parameter is used to control the communication protocols that the server will listen on for search requests from clients, and that clients will use to send out search requests. The *discover_comm* parameter can be set to TCP/IP; this is the only protocol supported by SEARCH discovery.

On the Administration Server, the values specified for *discover_comm* must be equal to, or a subset of, the values set for *db2comm*.

To avoid problems with the Control Center and the Client Configuration Assistant, ensure that *DB2COMM* is set in the DB2 registry. It is not recommended that you use any other mechanism to set the *DB2COMM* value.

On the server, the *discover_comm* parameter is set in the Administration Server. On the client (or a server acting as a client), *discover_comm* is set in the instance.

Note: When using discovery search mode, at least one protocol specified by the *discover_comm* parameter on the client must match those specified by *discover_comm* on the Administration Server. If there is no match, the server will not respond to the client's requests.

Check the settings for the *db2comm* registry value by issuing the **db2set DB2COMM** command. For more information, see Chapter 21, “Controlling Your DB2 Environment” on page 213.

In addition, there are two DB2 profile registry values that can be used to tune search discovery on the client: *db2discoverytime* and *db2nbdiscoverrcvbufs*. The default values should be suitable in most cases. For more information, refer to the *Administration Guide*.

Hiding Server Instances and Databases from Discovery

You may have multiple instances, and multiple databases within these instances, on a server. You may want to hide some of these from the discovery process.

To allow clients to discover server instances on a system, set the *discover_inst* database manager configuration parameter in each server instance on the system to ENABLE (this is the default value). Set this parameter to DISABLE to hide this instance and its databases from discovery.

To allow a database to be discovered from a client, set the *discover_db* database configuration parameter to ENABLE (this is the default value). Set this parameter to DISABLE to hide the database from discovery.

Setting Discovery Parameters

discover and discover_comm
The *discover* and *discover_comm* parameters are set in the Administration Server on the server system, and in the client instance. Set these parameters as follows:

- On the Administration Server:

  Update the Administration Server's configuration file, in the command line processor, as follows:

  ```
  update admin cfg using discover [ DISABLE | KNOWN | SEARCH ]
  update admin cfg using discover_comm [ TCPIP ]
  db2admin stop
  db2admin start
  ```

  Note: Search Discovery will only operate on TCP/IP and NetBIOS.

- On the client:

  1 Start the Client Configuration Assistant (CCA).

  2 Click on the **Client Settings** push button.

  3 Select the **Communications** tab.

  4 Select and change the parameters that you want to modify in the **Parameters** window.

> If the *discover_comm* includes *NETBIOS*, you must ensure that the Workstation name (*nname*) configuration parameter is set for the both the client and the Administration Server. Also, you must ensure that the *db2nbadapters* registry value is set to the Adapter number that you want to use. For more information, refer to the *Administration Guide*.

discover_inst and discover_db
Use the Control Center to set the *discover_inst* and *discover_db* parameters. To set these parameters, follow these steps:

1 Start the Control Center.

2 Select the instance that you want to configure and click on the right mouse button.

3 Select the **Configure** option from the pop-up menu. The Configuration window opens.

4 Select the **Environment** tab and select the *discover_inst* parameter.

5 Enter the desired value in the **Value** box and click on **OK**.

6 Select the database that you want to configure and click on the right mouse button.

7 Select the **Configure** option from the pop-up menu. The Configuration window opens.

8 Select the **Environment** tab and select the *discover_db* parameter.

9 Enter the desired value in the **Value** box and click on **OK**.

db2discoverytime and db2nbdiscoverrcvbufs
The *db2discoverytime* and *db2nbdiscoverrcvbufs* profile registry values are set in the client instance (or a server acting as a client). Set these parameters as follows:

- To set the *db2discoverytime* registry value to 35 seconds, enter the following command:

  ```
  db2set db2discoverytime=35
  ```

  This specifies that the searched discovery should wait 35 seconds for a response from servers.

- To set the *db2nbdiscoverrcvbufs* registry value to 20, enter the following command:

  ```
  db2set db2nbdiscoverrcvbufs=20
  ```

  This specifies the number of NetBIOS buffers that will be allocated for concurrent response messages from discovered servers.

Manually Configure a Connection to a DB2 Database

Manually configuring a database connection requires you to know:

- One of the protocols supported by the server instance containing the database.
- The protocol connection information required to configure the connection to the server instance.
- The name of the database on the server system.

With this information, the SmartGuide will guide you through the steps necessary to add the database connection.

1 Start the Client Configuration Assistant (CCA).

2 Click on the **Add** push button on the CCA's main panel to start the Add Database SmartGuide.

3 Select the **Manually configure a connection to a DB2 database** radio button, and click on the **Next** push button.

4 Select the protocol that you will use to connect to the database. Click on the **Next** push button.

5 Type in the required protocol parameters. Click on the **Next** push button.

6 Type in the name of the database in the **Target database** field.

7 If desired, you can select the **Alias** and **ODBC** tabs at the top of the panel to specify a database alias name for the database, or to select CLI/ODBC options for the database. This step is optional.

8 Click on the **Done** push button.

9 You can test the connection that has been added by clicking on the **Test Connection** push button.

Chapter 25. Using the DB2 Workstation as a Server for Host Applications

You can set up DB2 on the workstation to be a database server for clients or applications that use Distributed Relational Database Architecture (DRDA). The DRDA Application Server (AS) feature enables DB2 to function as a database server for DRDA Application Requesters.

DRDA AS is installed automatically with your DB2 server. However, there are some steps required to set up the connection with an Application Requester. These steps are described in this section.

For more information about DRDA concepts and a detailed description of DRDA commands and bind options, refer to the following publications:

- *Distributed Relational Database Architecture Reference* (SC26-4651)
- *DDM Architecture Reference Manual Level 4* (SC21-9526)

Application Requesters Supported

DB2 Universal Database servers support the following IBM Application Requesters:

- DB2 for MVS/ESA or DB2 for OS/390 Version 3.1 (or higher), DB2 for MVS Version 4 (or higher), and DB2 for VM Version 5 (or higher) AR

  For information on setting up the connection to a DB2 server via the DRDA-AS feature, see the instructions in “Setting Up DRDA-AS Functionality” on page 257.

- DB2 for OS/390 Version 5 (or higher) AR

  For information on setting up the connection to a DB2 server via the DRDA-AS feature, refer to the *DB2 Connectivity Supplement*.

- DB2/400 AR

  For information on setting up the connection to a DB2 server via the DRDA-AS feature, refer to the *DB2 Connectivity Supplement*.

- DB2 for VM & VSE (SQL/DS) Version 3.3 and 3.4 (or higher) AR

  For information on setting up the connection to a DB2 server via the DRDA-AS feature, refer to the *DB2 Connectivity Supplement*.

For information about non-IBM ARs, contact the software support for those products.

PTFs Required for DRDA Application Servers

The PTF numbers for DB2 for MVS APAR PN60988, which allows DB2 for MVS to work with the DB2 DRDA-AS, are:

UN75959 for DB2 for MVS 3.1

UN73393 for DB2 for MVS 3.1

For DB2 for MVS Version 3 Release 1 to use two-phase commit with CICS for AIX, the following PTF needs to be applied:

UN62361 for DB2 for MVS 3.1

For DB2 for VSE to use two-phase commit with CICS for AIX, the following PTFs need to be applied:

PQ01680 for DB2 for VSE

PQ03829 for DB2 for VSE

For DB2 for VM/ESA to use two-phase commit, the following PTFs need to be applied:

VM60922 for DB2 for VM/ESA

VM61072 for DB2 for VM/ESA

For DB2 for OS/390 Version 5 to connect to DB2 Universal Database Version 5, the following PTF needs to be applied:

PQ07537 for DB2 for OS/390 Version 5

For OS/400 DRDA ARs to work against the DB2 DRDA-AS, PTFs need to be applied on the OS/400 as follows:

Table 21. OS/400 PTFs Required for DRDA-AS

OS/400 PTR#	OS/400 V2R3 PTF	OS/400 V3R0.5 PTF	OS/400 V3R1 PTF
SA42461	SF23100	SF23950	SF23270
SA43024	SF23205	SF23950	SF23277
3989448	SF23101	SF23994	SF23271
3993315	SF23722	SF23988	SF23721
3994088	SF23987	SF23986	SF23985
3994297	SF23990	SF23989	SF23960

DRDA-AS Setup for the DRDA Application Requester

This section describes the steps and conditions required to set up a connection between a DB2 Universal Database DRDA Application Server and a DRDA application requester.

Before you begin, you must determine whether your connection will use the APPC communication protocol, or the TCP/IP communication protocol.

Platform	Supported Protocols
AIX	APPC or TCP/IP
Solaris	APPC or TCP/IP

Notes:

1. The protocol you choose may depend on the application requester:
   - APPC is supported by all application requesters
   - TCP/IP is supported by only DB2 for OS/390 V5.1 and DB Connect V5.0.
2. APPC supports both one-phase and two-phase commit; TCP/IP currently supports only one-phase commit.
3. To use APPC on AIX, the optional communication support component for SNA (`db2_05_00.cs.sna`) must also be installed.
4. To use APPC on Solaris, the optional communication support component for SNA (`db2cssna`) must also be installed.

Setting Up DRDA-AS Functionality

This section gives an overview of the steps required to set up DRDA-AS functionality. The example shows how to set up a connection between the DRDA AS on your DB2 server and a DB2 for MVS/ESA or DB2 for OS/390 AR:

1 Ensure DB2 for MVS/ESA or DB2 for OS/390 is installed and operational on the host.

Refer to the *DB2 Connectivity Supplement* for information on the configuration required for the DB2 for MVS/ESA or DB2 for OS/390 AR.

2 If you use APPC, ensure VTAM is installed and operational on the host.

Refer to the *DB2 Connect Enterprise Edition Quick Beginnings* and the *DB2 for OS/390 V5 Installation Guide* (GC26-8970) manual for information on configuring the DB2 for DB2 for MVS/ESA or DB2 for OS/390 AR.

3 Set up the DB2 server communications, if required. (Typically, DB2 server communications are set up as part of the DB2 installation. However, complete details are provided in Chapter 23, “Setting up Communications on the Server Using the Command Line Processor” on page 231. If you want to use two-phase commit, refer to the *Release Notes* for more information.)

4 Update the tables at the DB2 for MVS/ESA or DB2 for OS/390 host.

Refer to the *DB2 Connectivity Supplement* for more information.

5 Stop and restart the DB2 Universal Database instance named in DB2SERVICETPINSTANCE (that is, the instance you updated TPName in). Enter the following commands:

```
db2stop
db2start
```

6 Test the host-server DRDA connection by logging onto TSO and using DB2I/SPUFI.

Using the Application Server

DRDA-AS connections over TCP/IP or APPC are treated like any other connection to the DB2 server to ensure consistency in the way the maximum number of concurrent connections to a server is measured, from both DRDA and non-DRDA clients.

When using APPC, a DRDA-AR connects to the Application Server by specifying the Transaction Program Name (TPN) for the server. The TPN can correspond to the *tpname* parameter value in the instance's database manager configuration file. The TPN used by the DRDA-AS can also be the service transaction program X'07'6DB. When using this TPN, each separate instance of the DRDA-AS must specify a different local LU name.The local LU name is configured by the SPM-NAME parameter in the server's database manager configuration file.

Authentication

If you choose APPC as your communication protocol, the communications subsystem may restrict the types of database manager configuration authentication settings you can use at the DB2 Universal Database server. Not all communications subsystems expose the application requester's password to the application server when security (program) is used. When this is the case, the database manager configuration authentication must not be set to SERVER.

The following communications subsystems have this restriction:

AIX Bull SNA

Solaris all

To overcome the restriction that prevents you from using authentication SERVER, you can set the database manager authentication to DCS. This allows DRDA APPC connections that have been authenticated by the communication subsystem to succeed. However, with this setting the DB2 Universal Database will act as if authentication SERVER is being used for all other types of connections.

Troubleshooting

The DB2 DRDA Trace utility (**db2drdat**) is provided for tracing the DRDA data flow between an Application Requester and the DB2 Application Server. For more information on setting up this trace, refer to the *Troubleshooting Guide*.

DRDA Functions Supported

DRDA functions are categorized as required or optional. Table 22 identifies which functions are implemented in the AS on the DB2 Universal Database server. The subsequent table lists the supported bind options.

Table 22. Supported DRDA Functions

Description	Required (R) Optional (O)	Supported	Not Supported
DRDA level 1 required function	R	X*	
Rebind	O	X	
Describe user privileges	O		X
Describe RDB table	O		X
Interrupt RDB request	O		X

Note: * Certain required functions are not supported.

Bind Options Supported by the DB2 DRDA Application Server

Table 23 (Page 1 of 5). Bind Options Supported by the DB2 DRDA Application Server

Bind Option	Value	Sup-ported	Not Sup-ported	DB2 for MVS/ESA Precompile Option (Note 1)	DB2/VM Preprocessing Option	OS/400 Precompile Option	DB2 Prep or Bind Option
Package Version Name	**Null**	X		VERSION			VERSION
	Any other value		X				
Bind Existence Checking	**Object existence optional**		X	VALIDATE (**RUN**)[b]	**NOEXIST**	GENLVL(**10**, 11-40)	VALIDATE **RUN**
	Object existence required	X		VALIDATE (BIND)[b]	EXIST	GENLVL(00-09)	VALIDATE BIND
Package Replacement Option	**Replacement allowed**	X		ACTION (**REPLACE**)	**REPLACE**	REPLACE(***YES**)	ACTION **REPLACE**
	Replacement not allowed		X	ACTION(ADD)	NEW	REPLACE(*NO)	ACTION ADD
Package Authorization Option	**Keep authorizations**	X			**KEEP**		RETAIN **YES**
	Revoke authorizations		X		REVOKE		RETAIN NO
Statement String Delimiter (Note 2)	Apostrophe	X		**APOSTSQL**	**SQLAPOST**	OPTION([...] *APOSTSQL) (Note 3)	STRDEL APOSTROPHE
	Double quote		X	QUOTESQL	SQLQUOTE	OPTION([...] *QUOTESQL) (Note 4)	STRDEL QUOTE
Statement Decimal Delimiter (Note 5)	Period	X		**PERIOD**	**PERIOD**	OPTION([...] *PERIOD) or OPTION([...] ***SYSVAL**) (Note 6)	DECDEL PERIOD
	Comma		X	COMMA	COMMA	OPTION([...] *COMMA) or OPTION([...] ***SYSVAL**) (Note 6)	DECDEL COMMA
Date Format (Note 7)	**ISO**	X		DATE(ISO) (Note 8)	DATE(**ISO**)	DATFMT(*ISO) (Note 8)	DATETIME **ISO** (Note 9)

Table 23 (Page 2 of 5). Bind Options Supported by the DB2 DRDA Application Server

Bind Option	Value	Sup-ported	Not Sup-ported	DB2 for MVS/ESA Precompile Option (Note 1)	DB2/VM Preprocessing Option	OS/400 Precompile Option	DB2 Prep or Bind Option
	USA	X		DATE(USA)	DATE(USA)	DATFMT(*USA)	DATETIME USA
	EUR	X		DATE(EUR)	DATE(EUR)	DATFMT(*EUR)	DATETIME EUR
	JIS	X		DATE(JIS)	DATE(JIS)	DATFMT(*JIS)	DATETIME JIS
Time Format (Note 7)	**ISO**	X		TIME(ISO) (Note 8)	TIME(**ISO**)	TIMFMT(*ISO) (Note 8)	DATETIME **ISO** (Note 9)
	USA	X		TIME(USA)	TIME(USA)	TIMFMT(*USA)	DATETIME USA
	EUR	X		TIME(EUR)	TIME(EUR)	TIMFMT(*EUR)	DATETIME EUR
	JIS	X		TIME(JIS)	TIME(JIS)	TIMFMT(*JIS)	DATETIME JIS
Package Isolation Level (Note 10)	Repeatable read	X		ISOLATION(**RR**)[b]	ISOLATION(**RR**)		ISOLATION RR
	Read Stability (All)	X			ISOLATION(RS)	COMMIT(*ALL)	ISOLATION RS
	Cursor stability	X		ISOLATION(CS)[b]	ISOLATION(CS)	COMMIT(*CS)	ISOLATION **CS**
	Uncommitted Read (Change)	X			ISOLATION(UR)	COMMIT(***CHG**)	ISOLATION UR
	No commit		X (Note 11)			COMMIT(*NONE)	ISOLATION NC
Bind Creation Control	**No errors allowed**	X		SQLERROR (**NOPACKAGE**)[b]	**NOCHECK**	OPTION([...] *GEN) GENLVL(00-09, **10**, 11-20)	SQLERROR **NOPACKAGE**
	Check only	X			CHECK	OPTION([...] *NOGEN)	SQLERROR CHECK

Table 23 (Page 3 of 5). Bind Options Supported by the DB2 DRDA Application Server

Bind Option	Value	Supported	Not Supported	DB2 for MVS/ESA Precompile Option (Note 1)	DB2/VM Preprocessing Option	OS/400 Precompile Option	DB2 Prep or Bind Option
	Errors allowed		X	SQLERROR (CONTINUE)[b]	ERROR	OPTION([...] *GEN) GENLVL(21-40)	SQLERROR CONTINUE
Bind Explain Option	**No SQL statements**	X		EXPLAIN(**NO**)[b]	EXPLAIN(**NO**)		EXPLAIN **NO**
	All explainable SQL statements		X	EXPLAIN(YES)[b]	EXPLAIN(YES)		EXPLAIN YES
Package Owner Identifier	<***Authorization ID***>	X		OWNER[b]	OWNER		OWNER
	Any other value		X				
RDB Release Option	**Release at commit**	X		RELEASE (**COMMIT**)[b]	RELEASE (**COMMIT**)		RELEASE **COMMIT**
	Release at conversation deallocation		X	RELEASE (DEALLOCATE)[b]	RELEASE (DEALLOCATE)		RELEASE DEALLOCATE
Default RDB Collection ID	<***Authorization ID***>	X		QUALIFIER[b]	QUALIFIER	DFTRDBCOL	QUALIFIER
	Any other value		X				
Title (Package Description)	Any value (ignored by DB2)	X			LABEL	TEXT	TEXT
Query Block Protocol Control	**Fixed row**	X		CURRENTDATA (**YES**)[b]	SBLOCK	ALWBLK(***READ**)	BLOCKING **UNAMBIG**
	Limited block	X		CURRENTDATA (NO)[b]	BLOCK	ALWBLK (*ALLREAD)	BLOCKING ALL
	Forced fixed row	X			**NOBLOCK**	ALWBLK(*NONE)	BLOCKING NO
Package Default Char. Subtype							
	Use system default	X					CHARSUB **DEFAULT**

Table 23 (Page 4 of 5). Bind Options Supported by the DB2 DRDA Application Server

Bind Option	Value	Sup-ported	Not Sup-ported	DB2 for MVS/ESA Precompile Option (Note 1)	DB2/VM Preprocessing Option	OS/400 Precompile Option	DB2 Prep or Bind Option
If Default CCSID is SBCS	BIT		X		CHARSUB(BIT)		CHARSUB BIT
If Default CCSID is SBCS	SBCS	X			CHARSUB(SBCS)		CHARSUB SBCS
If Default CCSID is SBCS	MBCS		X		CHARSUB(MBCS)		CHARSUB MBCS
If Default CCSID is MBCS	BIT		X		CHARSUB(BIT)		CHARSUB BIT
If Default CCSID is MBCS	SBCS		X		CHARSUB(SBCS)		CHARSUB SBCS
If Default CCSID is MBCS	MBCS	X			CHARSUB(MBCS)		CHARSUB MBCS
	Any other value		X				

Package Default CCSID	**Value specified when DB2 database was created**	X			CCSIDSBCS()
					CCSIDGRAPHIC()
					CCSIDMIXED()
							CCSIDS CCSIDG CCSIDM

	Any other value		X				
Decimal Precision (Note 12)	31	X		DEC(31)			DEC 31
	Any other value		X	DEC(**15**)			DEC 15
Replaced Package Version Name	**Null**	X		REPLVER[b]			REPLVER
	Any other value		X				

Table 23 (Page 5 of 5). Bind Options Supported by the DB2 DRDA Application Server

Bind Option	Value	Sup-ported	Not Sup-ported	DB2 for MVS/ESA Precompile Option (Note 1)	DB2/VM Preprocessing Option	OS/400 Precompile Option	DB2 Prep or Bind Option

Note:

(*) Default values are in **bold**.
(1) Most are precompile options. Bind options are indicated by b.
(2) Defaults to what the target database supports. For DB2 the default is apostrophe.
(3) Default for non-COBOL applications.
(4) Default for COBOL applications.
(5) Defaults to what the target database supports. For DB2 the default is period.
(6) Depending on the installation, *SYSVAL is equivalent to *PERIOD or *COMMA.
(7) Date and time formats must be the same for the DB2 DRDA AS.
(8) Default is dependent on the installation.
(9) Format applies to both date and time. If not specified, it defaults based on the country code. This default is mapped to ISO in DRDA flow.
(10) Package isolation level has no default because an explicit value is always present in the DRDA datastream.
(11) The isolation level will be escalated to Uncommitted Read (Change).
(12) Defaults to what the target database supports. For DB2 the default is 31.

Special Considerations for DB2 for VM (SQL/DS)

Additional steps are needed to ensure the following DB2 for VM utilities work properly against a DB2 database through the Application Server.

- SQLDBSU

  1. Make sure a PTF for DB2 for VM APAR PN69073 is installed on your DB2 for VM system, either PTF UN91171 or PTF UN91172. (At the time of publication of this manual, there were no PTFs for DB2 for VM V4 or V5.)
  2. Set up dummy tables in your DB2 database by executing the **sqldbsu** utility provided with DB2 as `sqldbsu` *database_name*.
  3. Bind SQLDBSU from DB2 for VM. Refer to the “Using a DRDA Environment” section in the *SQL/DS System Administration for IBM VM Systems* manual for details. (You can skip the step on creating and populating the SQLDBA.DBSOPTIONS table because this is done by the **sqldbsu** utility in the previous step.)

- ISQL

  1. Complete the steps described above for SQLDBSU.
  2. Set up dummy tables in your DB2 database by executing the **isql** utility provided with DB2 as `isql` *database_name*.
  3. Bind ISQL from DB2 for VM. Refer to the “Using a DRDA Environment” in the *SQL/DS System Administration for IBM VM Systems* manual for details.

Note: The **sqldbsu** and **isql** utilities are in *INSTHOME/sqllib/misc* where *INSTHOME* is the home directory of the instance owner.

If you installed DB2 on drive *C*, using the default directory *sqllib*, then no special setup is required for RXSQL; refer to the *SQL/DS Procedures Language Interface Installation* manual for details.

Security and Auditability

Under CPIC, DB2 system security (authentication CLIENT, SERVER, or DCS) must be used with CPIC security SAME or PROGRAM. When these combinations are used, the user ID and password sent in the AR's APPC ALLOCATE is used to CONNECT to the requested database. CPIC security level NONE is allowed only with DCE authentication. In this case, the encrypted dce ticket is flown as part of the CONNECT attempt.

Under TCP/IP, all security information is flown in the CONNECT attempt.

User ID translation is not supported by this AS.

Configuration Considerations

You configure the Application Server using regular DB2 database manager configuration parameters and specifically DRDA Heap Size. You may need to change the setting for some parameters because of the additional resources required for the Application Server.

DRDA Heap Size (drda_heap_sz)

DRDA heap size specifies the amount of memory, in pages, that is allocated for use by the Application Server.

Refer to the *Administration Guide* for more information about database manager configuration.

Chapter 26. Removing DB2 Products

This section shows you how to remove DB2 products. You need to perform the following steps:

1 Stop the Administration Server.

2 Stop all DB2 Instances.

3 Remove the Administration Server.

4 Remove DB2 Instances - this step is optional.

5 Remove the DB2 products.

Step 1. Stop the Administration Server

You must stop the Administration Server before you remove DB2 products. To stop the Administration Server, you need to perform the following steps:

1 Log in as root.

2 Obtain the name of the Administration Server using the following command:

```
DB2DIR/bin/dasilist
```

where DB2DIR	= /usr/lpp/db2_05_00	on AIX
	= /opt/IBMdb2/V5.0	on Solaris

3 Stop the Administration Server.

To stop the Administration Server, you must perform the following steps:

a Log in as the Administration Server owner.

b Run the start up script as follows:

```
. INSTHOME/sqllib/db2profile    (for Bourne or Korn shell)
source INSTHOME/sqllib/db2cshrc (for C shell)
```

where INSTHOME is the home directory of the instance.

c Stop the Administration Server using the **db2admin** command as follows:

```
db2admin stop
```

where DB2DIR	= /usr/lpp/db2_05_00	on AIX
	= /opt/IBMdb2/V5.0	on Solaris

Step 2. Stop All DB2 Instances

You must stop all DB2 Instances before you remove DB2 products. To stop DB2 Instances, you need to perform the following steps:

1 Obtain a list of the names of all DB2 instances on your system using the following command:

```
DB2DIR/bin/db2ilist
```

where DB2DIR	= /usr/lpp/db2_05_00	on AIX
	= /opt/IBMdb2/V5.0	on Solaris

2 Stop the instance.

To stop a database instance:

a Log in as the instance owner.

b Run the start up script as follows:

```
. INSTHOME/sqllib/db2profile    (for Bourne or Korn shell)
source INSTHOME/sqllib/db2cshrc (for C shell)
```

where `INSTHOME` is the home directory of the instance that you want to use.

c Stop the DB2 database manager by entering the **db2stop** command from a command line.

3 Repeat steps 2 and 3 for each instance.

Step 3. Remove the Administration Server

You must remove the Administration Server before you remove DB2 products. To remove the Administration Server, you need to perform the following steps:

1 Log in as root.

2 Obtain the name of the Administration Server using the following command:

```
DB2DIR/bin/dasilist
```

where DB2DIR	= /usr/lpp/db2_05_00	on AIX
	= /opt/IBMdb2/V5.0	on Solaris

3 Log in as the Administration Server owner.

4 Run the start up script as follows:

```
. INSTHOME/sqllib/db2profile    (for Bourne or Korn shell)
source INSTHOME/sqllib/db2cshrc (for C shell)
```

where INSTHOME is the home directory of the instance.

5 Stop the Administration Server using the **db2admin** command as follows:

```
db2admin stop
```

6 Backup the files in the *ASHOME/sqllib* directory, if needed, where *ASHOME* is the home directory of the Administration Server.

7 Log off.

8 Log in as *root* and remove the Administration Server using the **dasidrop** command as follows:

```
DB2DIR/instance/dasidrop ASName
```

where DB2DIR	= /usr/lpp/db2_05_00	on AIX
	= /opt/IBMdb2/V5.0	on Solaris

and `ASName` is the name of the instance being removed.

Note: The **dasidrop** command removes the *sqllib* directory under the home directory of the Administration Server.

Step 4. Remove DB2 Instances

You can optionally remove some or all of the DB2 Version 5 Instances on your system. Once an instance is removed, all the DB2 databases owned by the instance, if any, will not be usable. Remove DB2 Instances only if you are not planning to use DB2 Version 5 products, or if you do not want to migrate existing instances to a later version of DB2. See Chapter 7, “Migrating from Previous Versions” on page 91 for more information on how to migrate or upgrade to higher versions of DB2.

To remove DB2 Instances, you need to perform the following steps:

1 Log in as root.

2 Obtain a list of the names of all DB2 instances on your system using the following command:

```
DB2DIR/bin/db2ilist
```

where DB2DIR	= /usr/lpp/db2_05_00	on AIX
	= /opt/IBMdb2/V5.0	on Solaris

3 Remove the instance.

To remove an instance, perform the following steps:

a Log in as the instance owner.

b Stop the instance by issuing the **db2stop** command.

c Back up files in the *INSTHOME/sqllib* directory, if needed.

For example, you might want to save the database manager configuration file, *db2systm*, the *db2nodes.cfg* file, or user defined function or fenced stored procedure applications in *INSTHOME/sqllib/function*, where `INSTHOME` is the home directory of the instance owner.

d Log off as the instance owner.

e Log in as root.

f Remove the instance by executing the **db2idrop** command:

```
DB2DIR/instance/db2idrop InstName
```

where DB2DIR	= /usr/lpp/db2_05_00	on AIX
	= /opt/IBMdb2/V5.0	on Solaris

and *InstName* is the login name of the instance.

Because *INSTHOME* is mounted, you only have to run the **db2idrop** command once to remove the instance.

g As *root*, remove the instance owner's user ID and group (if used only for that instance). Do not remove these if you are planning to re-create the instance.

This step is optional since the instance owner and the instance owner group may be used for other purposes.

The **db2idrop** command removes the instance entry from the list of instances and removes the *INSTHOME/sqllib* directory.

Step 5. Remove the Products

> You must stop all outstanding DB2 processes before removing the DB2 products.

The following steps describe how to remove DB2 products.

Remove DB2 Products on AIX Systems

You can remove the DB2 products on AIX using the **SMIT** interface as follows:

1. Log in as root.
2. Type **smit install_remove** to proceed directly to the Remove Software Products screen.
3. Press F4 to display a list of the software to remove. Press F7 at some or all of the entries that have a prefix of `db2_05_00`.

   Note: For a detailed description of the software packages, see Appendix D, "Contents of the DB2 Products" on page 309.
4. Press Enter to start removing the DB2 products.

> You can also remove all DB2 Version 5 products on AIX, using the **installp** command with the de-install option:
>
> ```
> installp -u db2_05_00
> ```

Remove DB2 Products on Solaris Systems

You can remove the DB2 products on the Solaris operating system using the **pkgrm** program as follows:

1. Log in as root.
2. Determine the packages for all DB2 for Solaris related products you have installed on your system by typing:

   ```
   pkginfo | grep -i db2 | grep 50
   ```
3. Remove all packages listed in step 2 with the **pkgrm** command.
4. Select **Yes** at the prompt for each package to be removed.

Before removing a package, all its dependent packages must be removed first. You must remove packages in the following order:

a. db2ht*%L*50
b. db2ms*%L*50
c. db2adts50
d. db2adt50
e. db2cent50
f. db2entp50
g. db2wkgp50
h. db2cnvt50
i. db2cnvc50
j. db2cnvk50
k. db2cnvj50
l. db2smpl50
m. db2repl50
n. db2conn50
o. db2das50
p. db2cipx50
q. db2cdrd50
r. db2csna50
s. db2crte50
t. db2engn50
u. db2rte50
v. db2jdbc50
w. db2odbc50
x. db2cliv50

Where *%L* is the language/locale identifier

Note: See Appendix D, “Contents of the DB2 Products” on page 309 for a list of packages for Solaris.

Part 7. Appendixes

Appendix A. Obtaining Information

Information on the DB2 product is available in several different formats:

- Online Help
- PostScript (.ps)
- HTML (.htm)

All books described in this section are also available in printed form. To order printed versions, contact your IBM authorized dealer or marketing representative, or phone 1-800-879-2755 in the United States or 1-800-IBM-4YOU in Canada.

How the DB2 Library Is Structured

The DB2 Universal Database library consists of online help, and books. This section describes the information that is provided, and how to access it.

To help you access product information online, DB2 provides the Information Center on OS/2, Windows 95, and the Windows NT operating systems. You can view task information, DB2 books, troubleshooting information, sample programs, and DB2 information on the Web. "About the Information Center" on page 281 has more details.

Online Help

Online help is available with all DB2 components. The following table describes the various types of help.

Type of Help	Contents	How to Access...
Command Help	Explains the syntax of commands in the command line processor.	From the command line processor in interactive mode, enter: **?** *command* where *command* is a keyword or the entire command. For example, **?** *catalog* displays help for all the CATALOG commands, whereas **?** *catalog database* displays help for the CATALOG DATABASE command.

Type of Help	Contents	How to Access...
Control Center Help	Explains the tasks you can perform in a window or notebook. The help includes prerequisite information you need to know, and describes how to use the window or notebook controls.	From a window or notebook, click on the **Help** push button or press the F1 key.
Message Help	Describes the cause of a message number, and any action you should take.	From the command line processor in interactive mode, enter: **?** *message number* where *message number* is a valid message number. For example, **?** *SQL30081* displays help about the SQL30081 message. To view message help one screen at a time, enter: **?** *XXXnnnnn* **| more** where *XXX* is the message prefix, such as SQL, and *nnnnn* is the message number, such as 30081. To save message help in a file, enter: **?** *XXXnnnnn* > *filename.ext* where *filename.ext* is the file where you want to save the message help.
SQLSTATE Help	Explains SQL states and class codes.	From the command line processor in interactive mode, enter: **?** *sqlstate* or **?** *class-code* where *sqlstate* is a valid five digit SQL state and *class-code* is a valid two digit class code. For example, **?** *08003* displays help for the 08003 SQL state, whereas **?** *08* displays help for the 08 class code.

DB2 Books

The table in this section lists the DB2 books. They are divided into two groups:

- Cross-platform books: These books are for DB2 on any of the supported platforms.
- Platform-specific books: These books are for DB2 on a specific platform. For example, there is a separate *Quick Beginnings* book for DB2 on OS/2, Windows NT, and UNIX-based operating systems.

Most books are available in HTML and PostScript format, and in hardcopy that you can order from IBM. The exceptions are noted in the table.

You can obtain DB2 books and access information in a variety of different ways:

View To view an HTML book, you can do the following:

- If you are running DB2 administration tools on OS/2, Windows 95, or the Windows NT operating systems, you can use the Information Center. “About the Information Center” on page 281 has more details.
- Use the open file function of the Web browser supplied by DB2 (or one of your own) to open the following page:

  ```
  sqllib/doc/html/index.htm
  ```

  The page contains descriptions of and links to the DB2 books. The path is located on the drive where DB2 is installed.

  You can also open the page by double-clicking on the **DB2 Online Books** icon. Depending on the system you are using, the icon is in the main product folder or the Windows Start menu.

Search To search for information in the HTML books, you can do the following:

- Click on **Search the DB2 Books** at the bottom of any page in the HTML books. Use the search form to find a specific topic.
- Click on **Index** at the bottom of any page in an HTML book. Use the Index to find a specific topic in the book.
- Display the Table of Contents or Index of the HTML book, and then use the find function of the Web browser to find a specific topic in the book.
- Use the bookmark function of the Web browser to quickly return to a specific topic.
- Use the search function of the Information Center to find specific topics. “About the Information Center” on page 281 has more details.

Print To print a book on a PostScript printer, look for the file name shown in the table. The following is an example of the file name and extension of the PostScript version of the DB2 books:

- db2d0e50.psz on OS/2 and Windows operating systems

- db2d0e50.ps.Z on Unix-based operating systems.

For more information, see “Printing the PostScript Books” on page 283.

Order To order a hardcopy book from IBM, use the form number, which you can find in the online version of the book.

Book Name	Book Description	File Name
Cross-Platform Books		
Administration Getting Started	Introduces basic DB2 database administration concepts and tasks, and walks you through the primary administrative tasks.	db2k0x50
Administration Guide	Contains information required to design, implement, and maintain a database to be accessed either locally or in a client/server environment.	db2d0x50
API Reference	Describes the DB2 application programming interfaces (APIs) and data structures you can use to manage your databases. Explains how to call APIs from your applications.	db2b0x50
CLI Guide and Reference	Explains how to develop applications that access DB2 databases using the DB2 Call Level Interface, a callable SQL interface that is compatible with the Microsoft ODBC specification.	db2l0x50
Command Reference	Explains how to use the command line processor, and describes the DB2 commands you can use to manage your database.	db2n0x50
DB2 Connect Enterprise Edition Quick Beginnings	Provides planning, migrating, installing, configuring, and using information for DB2 Connect Enterprise Edition. Also contains installation and setup information for all supported clients.	db2cyx50
DB2 Connect Personal Edition Quick Beginnings	Provides planning, installing, configuring, and using information for DB2 Connect Personal Edition.	db2c1x50
DB2 Connect User's Guide	Provides concepts, programming and general using information about the DB2 Connect products.	db2c0x50
DB2 Connectivity Supplement	Provides setup and reference information for customers who want to use DB2 for AS/400, DB2 for OS/390, DB2 for MVS, or DB2 for VM as DRDA Application Requesters with DB2 Universal Database servers, and customers who want to use DRDA Application Servers with DB2 Connect (formerly DDCS) application requesters. **Note:** Available in HTML and PostScript formats only.	db2h1x50 No form number
Embedded SQL Programming Guide	Explains how to develop applications that access DB2 databases using embedded SQL, and includes discussions about programming techniques and performance considerations.	db2a0x50

Book Name	Book Description	File Name
Glossary	Provides a comprehensive list of all DB2 terms and definitions. **Note:** Available in HTML format only.	db2t0x50 No form number
Installing and Configuring DB2 Clients	Provides installation and setup information for all DB2 Client Application Enablers and DB2 Software Developer's Kits. **Note:** Available in HTML and PostScript formats only.	db2iyx50 No form number
Master Index	Contains a cross reference to the major topics covered in the DB2 library. **Note:** Available in PostScript format and hardcopy only.	db2w0x50
Message Reference	Lists messages and codes issued by DB2, and describes the actions you should take.	db2m0x50
Replication Guide and Reference	Provides planning, configuring, administering, and using information for the IBM Replication tools supplied with DB2.	db2e0x50
Road Map to DB2 Programming	Introduces the different ways your applications can access DB2, describes key DB2 features you can use in your applications, and points to detailed sources of information for DB2 programming.	db2u0x50
SQL Getting Started	Introduces SQL concepts, and provides examples for many constructs and tasks.	db2y0x50
SQL Reference	Describes SQL syntax, semantics, and the rules of the language. Also includes information about release-to-release incompatibilities, product limits, and catalog views.	db2s0x50
System Monitor Guide and Reference	Describes how to collect different kinds of information about your database and the database manager. Explains how you can use the information to understand database activity, improve performance, and determine the cause of problems.	db2f0x50
Troubleshooting Guide	Helps you determine the source of errors, recover from problems, and use diagnostic tools in consultation with DB2 Customer Service.	db2p0x50
What's New	Describes the new features, functions, and enhancements in DB2 Universal Database. **Note:** Available in HTML and PostScript formats only.	db2q0x50 No form number
	Platform-Specific Books	
Building Applications for UNIX Environments	Provides environment setup information and step-by-step instructions to compile, link, and run DB2 applications on a UNIX system.	db2axx50

Book Name	Book Description	File Name
Building Applications for Windows and OS/2 Environments	Provides environment setup information and step-by-step instructions to compile, link, and run DB2 applications on a Windows or OS/2 system.	db2a1x50
DB2 Personal Edition Quick Beginnings	Provides planning, installing, migrating, configuring, and using information for DB2 Universal Database Personal Edition on OS/2, Windows 95, and the Windows NT operating systems.	db2i1x50
DB2 SDK for Macintosh Building Your Applications	Provides environment setup information and step-by-step instructions to compile, link, and run DB2 applications on a Macintosh system. **Note:** Available in PostScript format and hardcopy for DB2 Version 2.1.2 only.	sqla7x02
DB2 SDK for SCO OpenServer Building Your Applications	Provides environment setup information and step-by-step instructions to compile, link, and run DB2 applications on a SCO OpenServer system. **Note:** Available for DB2 Version 2.1.2 only.	sqla9x02
DB2 SDK for Silicon Graphics IRIX Building Your Applications	Provides environment setup information and step-by-step instructions to compile, link, and run DB2 applications on a Silicon Graphics system. **Note:** Available in PostScript format and hardcopy for DB2 Version 2.1.2 only.	sqlaax02
DB2 SDK for SINIX Building Your Applications	Provides environment setup information and step-by-step instructions to compile, link, and run DB2 applications on a SINIX system. **Note:** Available in PostScript format and hardcopy for DB2 Version 2.1.2 only.	sqla8x00
Quick Beginnings for OS/2	Provides planning, installing, migrating, configuring, and using information for DB2 Universal Database on OS/2. Also contains installing and setup information for all supported clients.	db2i2x50
Quick Beginnings for UNIX	Provides planning, installing, configuring, migrating, and using information for DB2 Universal Database on UNIX-based platforms. Also contains installing and setup information for all supported clients.	db2ixx50
Quick Beginnings for Windows NT	Provides planning, installing, configuring, migrating, and using information for DB2 Universal Database on the Windows NT operating system. Also contains installing and setup information for all supported clients.	db2i6x50

Book Name	Book Description	File Name
Quick Beginnings for Extended Enterprise Edition for UNIX	Provides planning, installing, configuring, and using information for DB2 Universal Database Extended Enterprise Edition for UNIX. This book supercedes the DB2 Extended Enterprise Edition Quick Beginnings for AIX book, and is suitable for use with all versions of DB2 Extended Enterprise Edition that run on UNIX-based platforms.	db2v3x50
Quick Beginnings for Extended Enterprise Edition for Windows NT	Provides planning, installing, configuring, and using information for DB2 Universal Database Extended Enterprise Edition for Windows NT.	db2v6x50

Notes:

1. The character in the sixth position of the file name indicates the language of a book. For example, the file name `db2d0e50` indicates that the *Administration Guide* is in English. The following letters are used in the file names to indicate the language of a book:

Language	Identifier	Language	Identifier
Brazilian Portuguese	B	Japanese	J
Bulgarian	U	Korean	K
Czech	X	Norwegian	N
Danish	D	Polish	P
English	E	Russian	R
Finnish	Y	Simp. Chinese	C
French	F	Slovenia	L
German	G	Spanish	Z
Greek	A	Swedish	S
Hungarian	H	Trad. Chinese	T
Italian	I		

2. For late breaking information that could not be included in the DB2 books, see the *Release.Notes* file. This file is located in the *DB2DIR/Readme/%L* directory, where `%L` is the locale name and DB2DIR=*/usr/lpp/db2_0500*.

About the Information Center

The Information Center provides quick access to DB2 product information. The Information Center is available on OS/2, Windows 95, and the Windows NT operating systems. You must install the DB2 administration tools to see the Information Center.

Depending on your system, you can access the Information Center from the:

- Main product folder
- Toolbar in the Control Center
- Windows Start menu.

The Information Center provides the following kinds of information. Click on the appropriate tab to look at the information:

Tasks Lists tasks you can perform using DB2.

Reference Lists DB2 reference information, such as keywords, commands, and APIs.

Books Lists DB2 books.

Troubleshooting Lists categories of error messages and their recovery actions.

Sample Programs Lists sample programs that come with the DB2 Software Developer's Kit. If the Software Developer's Kit is not installed, this tab is not displayed.

Web Lists DB2 information on the World Wide Web. To access this information, you must have a connection to the Web from your system.

When you select an item in one of the lists, the Information Center launches a viewer to display the information. The viewer might be the system help viewer, an editor, or a Web browser, depending on the kind of information you select.

The Information Center provides search capabilities so you can look for specific topics, and filter capabilities to limit the scope of your searches.

Ordering the Printed DB2 Books

You can also order the printed DB2 manuals. You can either order the books individually by the form number (to see the form number, view the online HTML book), or you can a set of books. There are two sets of books available. The form number for the entire set of DB2 books is `SBOF-8915-00`. The form number for the books listed under the heading "Cross-Platform Books" is `SBOF-8914-00`.

Note: These form numbers only apply if you are ordering books that are printed in the English language.

Printing the PostScript Books

If you prefer to have printed copies of the manuals, you can decompress and print PostScript versions as follows.

On UNIX systems:

1 Mount the CD-ROM. Refer to the following procedures to mount the CD-ROM on different UNIX operating systems:

On AIX:

Perform the following steps to mount the CD-ROM on AIX operating systems:

a Log in as root.

b Insert the CD-ROM in the drive.

c Create a root directory by entering the following command:

```
mkdir -p /cdrom
```

d Allocate a CD-ROM file system by typing the following command:

```
smit storage
```

e Select **FileSystems**.

f Select **Add/Change/Show/Delete FileSystems**.

g Select **CD-ROM FileSystems**.

h Select **Add CDROM FileSystems**.

i Select the **DEVICE Name**.

> Device names for CD-ROM file systems must be unique. You may need to delete a previously-defined CD-ROM file system or use another name for your directory, if there is a duplicate device name.

j Respond to the `mount point` prompt by typing the following:

```
/cdrom
```

k Mount the CD-ROM file system by typing the following command:

```
smit mountfs
```

l Select the **FileSystem** name. For example, the name could be `/dev/cd0`.

m Select the **Directory** name, `/cdrom`.

n Select the **Type of filesystem**, `cdrfs`.

o Set the **Mount as READ-ONLY system** to `Yes`.

p Log off.

On Solaris:

Perform the following steps to mount the CD-ROM on Solaris operating systems:

a Log in as root.

b If the Volume Manager is not running on your system, enter the following commands to mount the CD-ROM:

```
mkdir -p /cdrom/unnamed_cdrom
mount -F hsfs -o ro /dev/dsk/c0t6d0s2 /cdrom/unnamed_cdrom
```

where */cdrom/unnamed_cdrom* is the CD-ROM mount directory.

> If you are mounting the CD-ROM drive from a remote system using NFS, the CD-ROM file system on the remote machine must be exported with *root* access. You must also mount that file system with *root* access on the local machine.

If the Volume Manager (vold) is running on your system, the CD-ROM is automatically mounted as:

```
/cdrom/unnamed_cdrom
```

c Log out.

2 Change to */cdrom/doc/%L/ps* directory on the CD-ROM, where */cdrom* is the mount point of the CD-ROM and *%L* is the name of the desired locale. The manuals will be installed in the previously-mentioned directory with file names ending with *.ps.Z*.

> Specify the full-path name for the file you intend to print.

Viewing Online Books

The manuals included with this product are in Hypertext Markup Language (HTML) softcopy format. Softcopy format enables you to search or browse the information more easily, using hypertext links for related information. It also makes it easier to share the library across your site.

Although you can use any HTML Version 3.2-compliant browser to view the online books, a copy of the Netscape browser is available on the Portfolio CD-ROM provided with DB2 Universal Database Version 5; you can optionally install it on your system.

Installing the Netscape Browser

If you do not have an HTML Version 3.2-compliant browser installed on your system, you can install the Netscape browser from the Portfolio CD-ROM.

To install Netscape, use the following instructions:

1. Mount the Portfolio CD-ROM.
2. Refer to the *README.TXT* file located at the root directory of the CD-ROM for installation instructions.

The *README.TXT* file also contains information about the products.

Setting up a Document Server

By default the DB2 information is installed on your local system. This means that each person who needs access to the DB2 information must install the same files. To have the DB2 information stored in a single location, use the following instructions:

1. Copy all files and sub-directories from \sqllib\doc\html on your local system to a web server. Each book has its own sub-directory containing all the necessary HTML and GIF files that make up the book. Ensure that the directory structure remains the same.
2. Configure the web server to look for the files in the new location. For information on NetQuestion, a search facility for HTML documents, see DB2 Online Books on an Intranet (http://www.software.ibm.com/data/pubs/papers/db2html.html)
3. The Information Center allows you to specify a base URL for all HTML files. The URL for the list of books should be used.
4. Once you are able to view the book files, you should bookmark commonly viewed topics such as:
   - List of books.
   - Tables of contents of frequently used books.
   - Frequently referenced articles like the "How to use Syntax Diagrams" topic.
   - Search form.

The search form provided with DB2 allows you to use several different search variations. These include:

- Local indexes and files — this works by default.
- Local indexes, remote files — use the **nqmap** command to set this up.
- Remote indexes, remote files — use the **nqmap** command and edit the search form to set this up.
- Create your own indexes for some other search engine — read the documentation provided with this search engine, and index all files in the \sqllib\doc\html directory.

Appendix B. Issuing Commands to Multiple Database Partition Servers

In a partitioned database system, you may want to issue commands to be run on machines in the instance, or on database partition servers (nodes). You can do so using the **rah** command or the **db2_all** command. The **rah** command allows you to issue commands that you want to run at machines in the instance. If you want the commands to run at database partition servers in the instance, you run the **db2_all** command. This section provides an overview of these commands. The information that follows applies to partitioned database systems only.

Note: Your login shell can be a Korn shell or any other shell; however, there are differences in the way the different shells handle commands containing special characters.

To determine the scope of a command, refer to the *Command Reference*. This book indicates whether a command runs on a single database partition server, or on all of them. If the command runs on one database partition server and you want it to run on all of them, use **db2_all**. The exception is the **db2trc** command, which runs on all the logical nodes (database partition servers) on a machine. If you want to run **db2trc** on all logical nodes on all machines, use **rah**.

Commands

You can run the commands sequentially at one database partition server after another, or you can run the commands in parallel. If you run the commands in parallel, you can either choose to have the output sent to a buffer and collected for display (the default behavior) or the output can be displayed at the machine where the command is issued.

To use the **rah** command, type:

```
rah command
```

To use the **db2_all** command, type:

```
db2_all command
```

To obtain help about **rah** syntax, type

```
rah "?"
```

The command can be almost anything which you could type at an interactive prompt, including, for example, multiple commands to be run in sequence. You separate multiple commands using a semicolon (;). Do not use the separator character following the last command.

The following example shows how to use the **db2_all** command to change the database configuration on all database partition servers that are specified in the *db2nodes.cfg* file. Because the ; character is placed inside double quotation marks, the request will run concurrently:

```
db2_all ";UPDATE DB CFG FOR sample USING LOGFILSIZ=100"
```

Command Descriptions

You can use the following commands:

Command	Description
rah	Runs the command on all machines.
db2_all	Runs the command on all database partition servers that you specify.
db2_kill	Abruptly stops all processes being run on multiple database partition servers and cleans up all resources on all database partition servers. This command renders your databases inconsistent. Do *not* issue this command except under direction from IBM service.
db2_call_stack	Causes all processes running on all database partition servers to write call traceback to syslog

These commands execute **rah** with certain implicit settings such as:

- Run in parallel at all machines
- Buffer command output in */tmp/$USER/db2_kill*, */tmp/$USER/db2_call_stack* respectively.

Specifying the Command to Run

You can specify the command:

- From the command line as the parameter
- In response to the prompt if you don't specify any parameter.

You should use the prompt method if the command contains the following special characters:

```
| & ; < > ( ) ` { } [ ] unsubstituted $
```

The command will be added to your command history just as if you typed it at the prompt. If you specify the command as the parameter on the command line, you must enclose it in double quotation marks if it contains any of the special characters just listed.

All special characters in the command can be entered normally (without being enclosed in quotation marks, except for \). If you need to include a \ in your command, you must type two backslashes (\\).

Note: If you are not using a Korn shell, all special characters in the command can be entered normally (without being enclosed in quotation marks, except for ` " \ unsubstituted $, and the single quotation mark (')). If you need to include one of these characters in your command, you must precede them by three backslashes (\\\). For example, if you need to include a \ in your command, you must type four backslashes (\\\\).

If you need to include a double quotation mark (") in your command, you must precede it by three backslashes, for example, \\\". You cannot include a single quotation mark (') in your command unless your command shell provides some way of entering a single quotation mark inside a singly quoted string.

Running Commands in Parallel

By default, the command is run sequentially at each machine, but you can specify to run the commands in parallel using background rshells by prefixing the command with certain prefix sequences. If the rshell is run in the background, then each command puts the output in a buffer file at its remote machine, This process retrieves the output in two pieces:

1. After the remote command completes.
2. After the rshell terminates, which may be later if some processes are still running.

The name of the buffer file is */tmp/$USER/rahout* by default, but it can be specified by the environment variables $RAHBUFDIR/$RAHBUFNAME.

When you specify that you want the commands to be run concurrently, by default, this script prefixes an additional command to the command sent to all hosts to check that $RAHBUFDIR and $RAHBUFNAME are usable for the buffer file. It creates $RAHBUFDIR. To suppress this, export an environment variable `RAHCHECKBUF=no`. You can do this to save time if you know the directory exists and is usable.

Before using **rah** to run a command concurrently at multiple machines, ensure that:

- A directory `/tmp/$USER` exists for your user ID at each machine. To create a directory if one does not already exist, run:

  ```
  rah ")mkdir /tmp/$USER"
  ```

- Add the following line to your *.kshrc* (for Korn shell syntax) or *.profile*, and also type it into your current session:

  ```
  export RAHCHECKBUF=no
  ```

- Ensure that each machine ID at which you run the remote command has an entry in its *.rhosts* file for the ID which runs **rah**; and the ID which runs **rah** has an entry in its *.rhosts* file for each machine ID at which you run the remote command.

Monitoring rah Processes

While any remote commands are still running or buffered output is still being accumulated, processes started by `rah` monitor activity to:

- Write messages to the terminal indicating which commands have not been run
- Retrieve buffered output.

The informative messages are written at an interval controlled by the environment variable RAHWAITTIME. See the help information for details on how specify this. All informative messages can be completely suppressed by exporting `RAHWAITTIME=0`.

The primary monitoring process is a command whose command name (as shown by the ps command) is **rahwait**>**or**. The first informative message tells you the pid (process id) of this process. All other monitoring processes will appear as **ksh** commands running the **rah** script (or the name of the symbolic link). If you want, you can stop all monitoring processes by the command:

```
kill <pid>
```

where <pid> is the process ID of the primary monitoring process. Do not specify a signal number. Leave the default of 15. This will not affect the remote commands at all, but will prevent the automatic display of buffered output. Note that there may be two or more different sets of monitoring processes executing at different times during the life of a single execution of **rah**. However, if at any time you stop the current set, then no more will be started.

If your regular login shell is not a Korn shell (for example */bin/ksh*), you can use **rah**, but there are some slightly different rules on how to enter commands containing the following special characters:

```
" ` unsubstituted $ '
```

For more information, type `rah "?"`. Also, in a UNIX-based environment, if the login shell at the ID which executes the remote commands is not a Korn shell, then the login shell at the ID which executes **rah** must also not be a Korn shell. (**rah** makes the decision as to whether the remote ID's shell is a Korn shell based on the local ID). The shell must not perform any substitution or special processing on a string enclosed in single quotation marks. It must leave it exactly as is.

Prefix Sequences

A prefix sequence is one or more special characters. Type one or more prefix sequences immediately preceding the characters of the command without any intervening blanks. If you want to specify more than one sequence, you can type them in any order, but characters within any multicharacter sequence must be typed in order. If you type any prefix sequences, you must enclose the entire command, including the prefix sequences in double quotation marks, as in the following example:

```
  rah "};ps -F pid,ppid,etime,args -u $USER"
```

The prefix sequences are:

Sequence **Purpose**

| Runs the commands in sequence in the background.

|& Runs the commands in sequence in the background and terminates the command after all remote commands have completed, even if some are still running (which may be later if for example, child processes are still running). In this case, the command starts a separate background process to retrieve any remote output generated after command termination and writes it back to the originating machine.

Note: Specifying & degrades performance, because more **rsh** commands are required.

|| Runs the commands in parallel in the background.

||& Runs the commands in parallel in the background and terminates the command after all remote commands have completed as described for the |& case above.

Note: Specifying & degrades performance, because more **rsh** commands are required.

; Same as ||& above. This is an alternative shorter form.

Note: Specifying ; degrades performance relative to ||, because more **rsh** commands are required.

] Prepends dot-execution of user's profile before executing command.

} Prepends dot-execution of file named in $RAHENV (probably *.kshrc*) before executing command.

]} Prepends dot-execution of user's profile followed by execution of file named in $RAHENV (probably *.kshrc*) before executing command.

) Suppresses execution of user's profile and of file named in $RAHENV.

' Echoes the command invocation to the machine.

< Sends to all the machines except this one.

<<–**nnn**< Sends to all-but-database partition server *nnn* (all database partition servers in *db2nodes.cfg* except for node number *nnn*, see the note below).

<<+**nnn**< Sends to only database partition server *nnn* (the database partition server in *db2nodes.cfg* whose node number is *nnn*, see the note below).

` Runs the remote command in the background with stdin, stdout and stderr all closed. This option is valid only when running the command in the background, that is, only in a prefix sequence which also includes | or ;. It allows the command to complete much sooner (as soon as the remote command has been initiated). If you specify this prefix character on the **rah** command line, then either enclose the command in single quotation marks, or enclose the command in double quotation marks, and precede the ` by \. For example,

```
rah ';`mydaemon'
```

or

```
rah ";\`mydaemon"
```

When run as a background process, the **rah** command will never wait for any output to be returned.

> Substitutes occurrences of <> with the machine name.

" Substitutes occurrences of () by the machine index, and substitutes occurrences of # by the node number.

Notes:

1. The machine index is a number that associated with a machine in the database system. If you are not running multiple logical nodes, the machine index for a machine corresponds to the node number for that machine in the node configuration file. To obtain the machine index for a machine in a multiple logical node environment, do not count duplicate entries for those machines that run multiple logical nodes. For example, if MACH1 is running two logical nodes and MACH2 is also running two logical nodes, the node number for MAC3 is 5 in the node configuration file. The machine index for MACH3, however, would be 3.
2. When " is specified, duplicates are not eliminated from the list of machines. See “Eliminating Duplicate Entries from the List of Machines” on page 293 if you want to eliminate duplicates.

When using the <<–nnn< and <<+nnn< prefix sequences, *nnn* is any 1-, 2- or 3-digit partition number which must match the *nodenum* value in the *db2nodes.cfg* file.

Note: Prefix sequences are considered to be part of the command. If you specify a prefix sequence as part of a command, you must enclose the entire command, including the prefix sequences, in double quotation marks.

Specifying the List of Machines

By default, the list of machines is taken from the *db2nodes.cfg* file. You can override this by:

- Specifying a pathname to the file that contains the list of machines by exporting the environment variable RAHOSTFILE.
- Specifying the list explicitly, as a string of names separated by spaces, by exporting the environment variable RAHOSTLIST. If both of these environment variables are exported, then RAHOSTLIST takes precedence.

Eliminating Duplicate Entries from the List of Machines

If you are running DB2 Extended Enterprise Edition with multiple logical nodes (database partition servers) on one machine, your *db2nodes.cfg* file will contain multiple entries for that machine. In this situation, the **rah** command needs to know whether you want the command to be executed once only on each machine or once for each logical node listed in the *db2nodes.cfg* file. Use the **rah** command to specify machines. Use the **db2_all** command to specify logical nodes.

If you specify machines, then **rah** will normally eliminate duplicates from the machine list, with the following exception:

- If you specify logical nodes, **db2_all** prepends the following assignment to your command:

  ```
  export DB2NODE=nnn  (for Korn shell syntax)
  ```

  where *nnn* is the node number taken from the corresponding line in the *db2nodes.cfg* file, so that the command will be routed to the desired database partition server.

When specifying logical nodes, you can restrict the list to include all logical nodes except one, or only specify one database partition server using the `<<−nnn<` and `<<+nnn<` prefix sequences. You may want to do this if you want to run a command at the catalog node first, and when that has completed, run the same command at all other database partition servers, possibly in parallel. This is usually required when running the **db2 restart database** command. You will need to know the node number of the catalog node to do this. See “Prefix Sequences” on page 291 for information about the prefix sequences.

If you execute **db2 restart database** using the **rah** command, duplicate entries are eliminated from the list of machines. However if you specify the ” prefix, then duplicates are not eliminated, because it is assumed that use of the ” prefix implies sending to each database partition server, rather than to each machine.

Controlling the rah Command

You can use the following environment variables to control the **rah** command.

Table 24.

Name	Meaning	Default
$RAHBUFDIR	directory for buffer	/tmp/$USER
$RAHBUFNAME	filename for buffer	rahout
$RAHOSTFILE	file containing list of hosts	*db2nodes.cfg*
$RAHOSTLIST	list of hosts as a string	extracted from $RAHOSTFILE
$RAHCHECKBUF	if set to "no", bypass checks	not set
$RAHSLEEPTIME	time in seconds this script will wait for initial output from commands run in parallel	86400 seconds for **db2_kill**, 200 seconds for all other
$RAHWAITTIME	interval in seconds between successive checks that remote jobs are still running and `rah: waiting for <pid> ...` messages. Specify any positive integer. Prefix value with a leading zero to suppress messages, for example, export RAHWAITTIME=045. It is not necessary to specify a low value as **rah** does not rely on these checks to detect job completion.	45 seconds
$RAHENV	specifies filename to be executed if $RAHDOTFILES=E or K or PE or B	$ENV
$RAHUSER	user ID under which the remote command is to be run	$USER

Note: The value of $RAHENV where **rah** is run is used, not the value (if any) set by the remote shell.

$RAHDOTFILES

Following are the .files that are run if no prefix sequence is specified:

P	.profile
E	File named in $RAHENV (probably *.kshrc*)
K	Same as E
PE	.profile followed by file named in $RAHENV (probably *.kshrc*)
B	Same as PE
N	None (or Neither)

Note: If your login shell is not a Korn shell, any dot files you specify to be executed will be executed in a Korn shell process, and so must conform to Korn shell syntax. So, for example, if your login shell is a C shell, to have your *.cshrc* environment set up for commands executed by **rah**, you should either create a Korn shell *INSTHOME/.profile* equivalent to your *.cshrc* and specify in your *INSTHOME/.cshrc*:

```
setenv RAHDOTFILES P
```

or you should create a Korn shell *INSTHOME/.kshrc* equivalent to your *.cshrc* and specify in your *INSTHOME/.cshrc*:

```
setenv RAHDOTFILES E
setenv RAHENV INSTHOME/.kshrc
```

Also, it is essential that your *.cshrc* does not write to stdout if there is no tty (as when invoked by **rsh**). You can ensure this by enclosing any lines which write to stdout by, for example,

```
if { tty -s } then echo "executed .cshrc";
endif
```

Determining Problems with rah

Here are suggestions on how to handle some problems that you may encounter when you are running **rah**:

1. **rah** hangs (or takes a very long time)

   This problem may be caused because:

   - **rah** has determined that it needs to buffer output, and you did not export `RAHCHECKBUF=no`. Therefore, before running your command, **rah** sends a command to all machines to check the existence of the buffer directory, and to create it if it does not exist.
   - One or more of the machines where you are sending your command is not responding. The **rsh** command will eventually time out but the time-out interval is quite long, usually about 60 seconds.

2. You have received messages such as:

   - Login incorrect
   - Permission denied

   Either one of the machines does not have the ID running **rah** correctly defined in its *.hosts* file, or the ID running **rah** does not have one of the machines correctly defined in its *.rhosts* file.

3. When running commands in parallel using background rshells, although the commands run and complete within the expected elapsed time at the machines, **rah** takes a long time to detect this and put up the shell prompt.

The ID running **rah** does not have one of the machines correctly defined in its *.rhosts* file.

4. Although **rah** runs fine when run from the shell command line, if you run **rah** remotely using rsh, for example,

   ```
   rsh somewher -l $USER db2_kill
   ```

   rah never completes.

   This is normal. **rah** starts background monitoring processes, which continue to run after it has exited. Those processes will normally persist until all processes associated with the command you ran have themselves terminated. In the case of **db2_kill**, this means termination of all database managers. You can terminate the monitoring processes by finding the process whose command is **rahwait**>**or** and kill <process_id>. Do not specify a signal number. Instead, use the default (15).

5. The output from **rah** is not displayed correctly, or **rah** incorrectly reports that $RAHBUFNAME does not exist, when multiple commands of **rah** were issued under the same $RAHUSER.

   This is because multiple concurrent executions of **rah** are trying to use the same buffer file (for example, $RAHBUFDIR/$RAHBUFNAME) for buffering the outputs. To prevent this problem, use a different $RAHBUFNAME for each concurrent **rah** command, for example in the following ksh:

   ```
   export RAHBUFNAME=rahout
   rah ";$command_1" &
   export RAHBUFNAME=rah2out
   rah ";$command_2" &
   ```

   or use a method that makes the shell choose a unique name automatically such as:

   ```
   RAHBUFNAME=rahout.$$ db2_all "....."
   ```

   Whatever method you use, you must ensure you clean up the buffer files at some point if disk space is limited. **rah** does not erase a buffer file at the end of execution, although it will erase and then re-use an existing file the next time you specify the same buffer file.

6. You entered

   ```
   rah '"print from ()'
   ```

   and received the message:

   ```
   ksh: syntax error at line 1 : `(' unexpected
   ```

   Prerequisites for the substitution of () and ## are:

   - Use **db2_all**, not **rah**.
   - Ensure a RAHOSTFILE is used either by exporting RAHOSTFILE or by defaulting to your *~/sqllib/db2nodes.cfg* file. Without these prerequisites, **rah** will leave the () and ## as is. You receive an error because the command **print from ()** is not valid.

For a performance tip when running commands in parallel, use | rather than |&, and use || rather than ||& or ; unless you truly need the function provided by &. Specifying & requires more **rsh** commands and therefore degrades performance.

Appendix C. Implementing Failover Support in the Solaris Operating Environment

You can set up your partitioned database system so that if a machine fails, the database partition server (or node) on it can run on another machine. On the Solaris Operating Environment, you implement failover support with Solstice HA 1.3.

Solstice HA 1.3 performs both failure detection and the restarting of resources in a clustered environment, as well as failover support for physical disks and IP addresses.

Solstice HA 1.3 Components

Solstice HA 1.3 consists of the following components and resources:

Physical machine
A Solstice HA 1.3 cluster consists of two physical machines. Each physical machine has one primary network interface, one or two logical network interfaces, a set of shared disks, and a disk for the operating system.

Logical host
A logical host consists of the following resources:

- One logical interface with its own IP *address* and *hostname*
- One Solstice Disksuite disk set
- One (or more) data service.

Two logical hosts can exist in a Solstice HA cluster. They reside on the machines within the HA cluster. The following is an example of the layout for a logical host filesystem for Solstice HA 1.3 with DB2 Extended Enterprise Edition. The name of the logical host in this example is `snap`:

/snap/	The logical host filesystem (needed for Solstice HA 1.3).
/snap/home/	The place to put the HA instance home directory.
/snap/disks/	The place to put SMS filesystems.

Private network
Private networks are used for communicating between two nodes. Heartbeat messages as well as Remote Procedure Calls (RPCs) travel over these networks to keep the two nodes in syncronous operation so that they can back up each other in the event of a failover.

Public networks
The public network includes all the primary and logical network interfaces and IP addresses. The logical network interfaces or logical hosts should be referred to when communicating with DB2 on the HA cluster.

Disk set
Disk sets contain one or more shared disks and a list of hosts which can access these disks. Only one host can own the disk sets for exclusive use at a time.

Figure 12 shows an example of the Solstice HA 1.3 components.

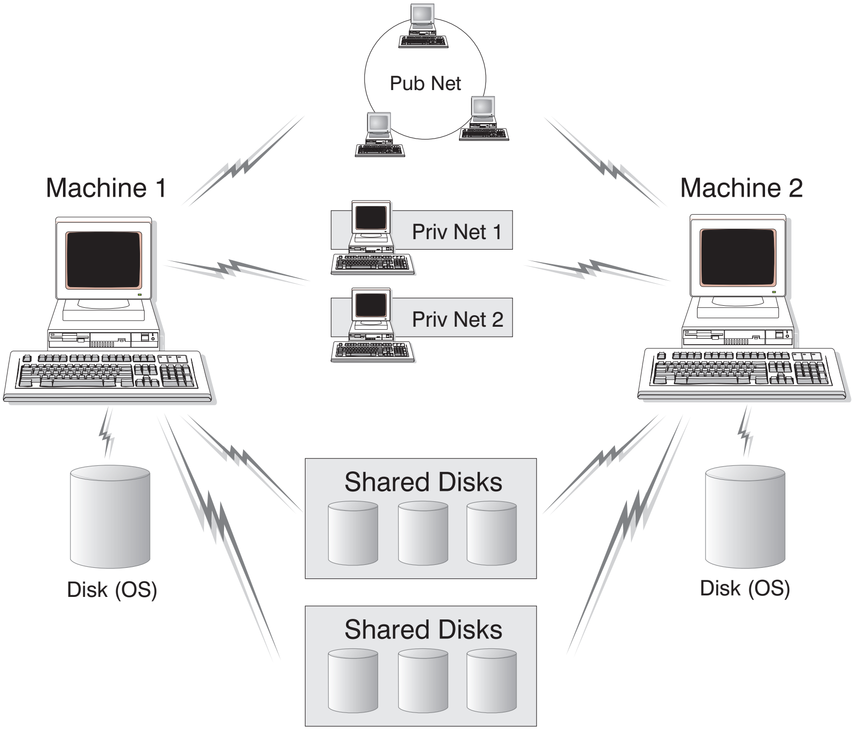


Figure 12. Solstice HA 1.3 Components

The following sections describe the different types of failover support, and how to implement them.

Failover Configurations

Two types of configuration are available in a DB2 system:

- Hot standby (asymmetric mode)
- Mutual takeover (symmetric mode)

Hot Standby Configuration

In a hot standby configuration, one machine in the Solstice HA 1.3 cluster provides dedicated backup support, and another machine participates in the partitioned database system. If the machine participating in the database system fails, the database partition server on it will be started on the backup machine. If you are running multiple logical nodes on a machine and it fails, the logical nodes will be started on the backup machine. Figure 13 shows an example of a hot standby configuration.

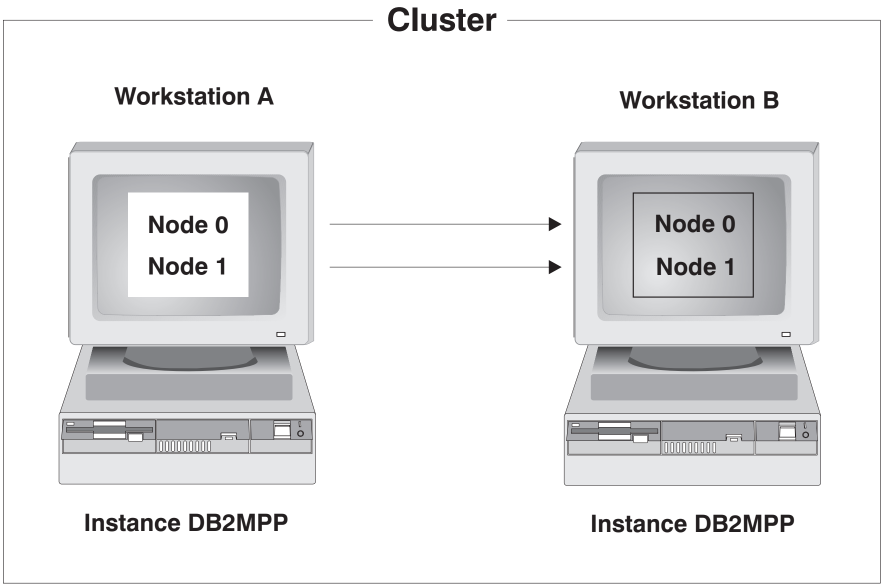


Figure 13. Hot Standby Configuration for Solstice HA 1.3

Mutual Takeover Configuration

In a mutual takeover configuration, both machines participate in the database system (that is, each machine has at least one database partition server running on it). If one of the machines in the cluster fails, the database partition server on the failing machine will be started to run as a logical node (or logical nodes) on the other machine. In a mutual takeover configuration, a database partition server on one machine can fail independently of the database partition server on another machine. Any database partition server can be active on any machine at any given point in time. Figure 14 shows an example of a mutual takeover configuration.

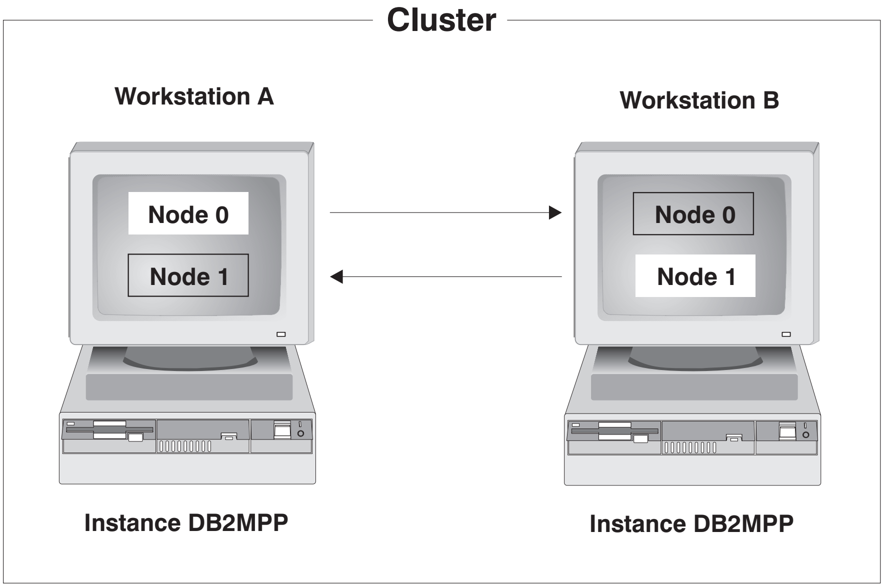


Figure 14. Mutual Takeover Configuration for Solstice HA 1.3

Setting up Solstice HA on DB2 for Solaris

To set up Solstice HA 1.3, perform the following steps:

- “Step 1. Choose an HA Configuration” on page 303.
- “Step 2. Select a Logical Host for Use as the HA-NFS Server for the HA Instance Home Directory” on page 303.
- “Step 3. Mount the Logical Host's Home Directory” on page 303.

- “Step 4. Create a DB2 Instance” on page 304.
- “Step 5. Run the HA-DB2 Extended Enterprise Edition Setup Program” on page 305.

Step 1. Choose an HA Configuration

To choose an HA configuration, perform the following steps:

1. Set up the two machines to use either a hot standby or mutual takeover configuration. For a hot standby configuration, use one logical host. For a mutual takeover, use two logical hosts, (which is the maximum for Solstice HA 1.3).

2. Decide on the amount of disk space that is required for each logical host and its resources, such as raw devices or SMS table space containers. Whether you use SMS or DMS (raw devices) table spaces, any disks belonging to a logical host must be included in its disk sets.

3. Decide on the location of the database partition servers on each logical host. For example, you may choose to locate all the database partition servers on one logical host or separate them over the two logical hosts.

Step 2. Select a Logical Host for Use as the HA-NFS Server for the HA Instance Home Directory

In an HA configuration, you can have one or two logical hosts. One logical host should contain the HA instance owner's physical directory. To choose a logical host to contain the HA instance owner's directory, create a directory under the logical host's filesystem, for example:

```
/logical_host/home/db2inst
```

where *db2inst* is the instance owner. Then use *HA-NFS* (which comes with Solstice HA 1.3) to export that directory to make it highly available.

Step 3. Mount the Logical Host's Home Directory

> Do not use the automounter command for this step. The automounter utility uses loopback virtual file system (LOFS) to mount the directories if they are local. Solstice HA 1.3 does not support LOFS.

On both machines, NFS-mount the highly available directory */logical_host/home/db2inst* from the HA-NFS server for use as the HA instance owner's home directory.

For example, a line in the */etc/vfstab* file for the logical host *snap* and the instance *db2inst* should look like the following:

```
snap:/snap/home/db2inst - /export/home/db2inst  nfs - yes hard,rw,bg
```

Step 4. Create a DB2 Instance

Create a DB2 instance using the NFS-mounted home directory as the instance owner's home directory. To create the instance, use the **db2icrt** command, which is located in the *DB2DIR/instance* directory, where `DB2DIR` is `/opt/IBMdb2/V5.0`.

> Run the **db2icrt** command from the machine where the home directory of the instance (*INSTHOME*) physically resides. Because the *INSTHOME* directory is NFS-mounted on all machines, you only have to run the **db2icrt** command on one machine.

The syntax of the **db2icrt** command is:

```
►►──db2icrt──[-h| -?]──[-d]──[-a AuthType]────[-u FencedID] ──────────►

►────── InstName ──────────────────────────────────────────────────────►◄
```

where:

-h | -? Display a help menu for this command.

-d Sets the debug mode that you can use for problem determination.

-a *AuthType* Is an optional parameter that specifies the authentication type for the instance. Valid authentication types are SERVER, CLIENT, and DCS. If the *-a* parameter is not specified, the authentication type defaults to SERVER, if a DB2 server is installed. Otherwise, the *AuthType* is set to CLIENT.

Notes:

1. All databases in the instance have the same authentication type.
2. DCE authentication is not valid for this command; however, you can enable DCE authentication for an instance. For more information, refer to the *Administration Guide*.

-u *FencedID* Is the user under which the fenced UDFs and stored procedures will execute. This is not required if you install the DB2 Client Application Enabler or the DB2 Software Developer's Kit. For other products, this is a required parameter.

Note: *FencedID* may not be *root* or *bin*.

InstName Is the login name of the instance owner.

When you create an instance, ensure that its primary and secondary groups are different from the Administration Server's primary (SYSADM) group. When you create an instance on the same machine as the Administration Server, its SYSADM group is automatically added to the secondary group list of the Administration Server so that you can use the Control Center to perform administration tasks on that instance.

Examples:

- To create an instance for a DB2 server, you can use the following command:

```
db2icrt -u db2fenc1 db2inst1
```

- To create an instance for a DB2 client, you can use the following command:

```
db2icrt db2inst1
```

When an instance is created, its name is also added to the list of instances on the system.

The **db2icrt** command creates the *INSTHOME/sqllib* directory, where *INSTHOME* is the home directory of the instance owner.

> To avoid a potential loss of data if an instance is deleted, you should not create user files or directories under the *INSTHOME/sqllib* directory, other than those created by DB2. Exceptions are:
>
> 1. If your system supports fenced user defined functions and fenced stored procedures, put the fenced applications in the *INSTHOME/sqllib/function* directory.
> 2. The location of the *db2nodes.cfg* file. This file should be put in the *INSTHOME/sqllib* directory, where *INSTHOME* is the home directory of the instance owner.

Step 5. Run the HA-DB2 Extended Enterprise Edition Setup Program

To setup the HA instance with Solstice HA, run the **hadb2eee.setup** program. The program will prompt you for the following information:

- `HA-INSTANCE:` The HA instance
- `HA-MAINNODE:` The logical host which is exporting the HA-NFS home directory of the HA instance
- `NODE:` A set of database partition server (node) numbers for each logical host.

When the setup program completes successfully, it does the following:

- Creates the *db2nodes.cfg* configuration file, see "Step 8. Create the Node Configuration File" on page 84.
- Creates the *INSTHOME/.rhosts* file, see "Step 9. Make DB2 Extended Enterprise Edition Available" on page 86.
- Creates the symbolic links from the *INSTHOME/INSTHOME/NODEnnnn* directory to the */logical_hosts/disks/instance/NODEx* directory, where INSTHOME is the home directory of the instance owner and *x* represents each node.
- Creates the *hadb2-mpp.cfg* configuration file for HA-DB2 Extended Enterprise Edition.
- Creates the */var/db2/v5/hadb2-insts.cfg* configuration file.

During this step, the setup program runs itself (the **hadb2eee.setup** script) on the other node non-interactively.

- Registers DB2 with Solstice HA 1.3.

A sample *hadb2-mpp.cfg* file might look like this:

```
HA-INSTANCE: db2inst1 ON
HA-MAINNODE: snap
NODE:whirl  0
NODE:whirl  1
NODE:whirl  2
NODE:whirl  3
NODE:snap 4
NODE:snap 5
NODE:snap 6
NODE:snap 7
```

where `HA-INSTANCE` is the username of the instance owner, `HA-MAINNODE` is the name of the logical host which contains the home directory of the instance owner, and `NODE` binds a database partition server (node) number with a logical host.

You can now start up DB2 by entering the following command:

```
hareg -y hadb2
```

After completing the setup of Solstice HA 1.3, go to "Step 7. Create the Administration Server" on page 83 to continue with the remaining installation steps.

Adding or Removing a DB2 Database Partition Server

To add or remove database partition servers, do the following:

1. Stop Solstice HA and DB2 by running the **hadb2** command as follows:

   ```
   hareg -n hadb2
   ```

   This will ensure that scripts which control DB2 through Solstice HA will not execute during the process.

2. Add or remove the database partition servers. Refer to *Administration Guide* for instructions.

3. Modify the *hadb2-mpp.cfg* configuration file.

   If you added a new database partition server, this tells Solstice HA where to failover the database partition server.

4. Start Solstice HA and DB2 by running the **hadb2** command as follows:

   ```
   hareg -y hadb2
   ```

For further information on adding and removing database partition servers, see the *Administration Guide*.

Appendix D. Contents of the DB2 Products

This section lists the contents of various DB2 Universal Database products.

Packaging

The following DB2 Universal Database components, packages, or filesets are available for installation:

Table 25 (Page 1 of 2). DB2 Components, Packages or Filesets

Description	Fileset Name	
	DB2 for AIX	**DB2 for Solaris**
DB2 Client	db2_05_00.client	db2cliv50
Open Database Connectivity (ODBC)	db2_05_00.odbc	db2odbc50
Java Database Connectivity (JDBC)	db2_05_00.jdbc	db2jdbc50
Administration Server	db2_05_00.das	db2das50
DB2 Run-time Environment	db2_05_00.db2.rte	db2rte50
DB2 Sample Database Source	db2_05_00.db2.samples	db2smpl50
DB2 Engine	db2_05_00.db2.engn	db2engn50
DB2 Extended Enterprise Edition Extentions	db2_05_00.pext	db2pext50
DB2 Replication	db2_05_00.repl	db2repl50
DB2 Connect	db2_05_00.conn	db2conn50
DB2 Communication Support - TCP/IP	db2_05_00.cs.rte	db2crte50
DB2 Communication Support - SNA	db2_05_00.cs.sna	db2csna50
DB2 Communication Support - DRDA Application Server	db2_05_00.cs.drda	db2cdrd50
DB2 Communication Support - IPX	db2_05_00.cs.ipx	db2cipx50
License Support of DB2 Extended Enterprise Edition	db2_05_00.xsrv	db2xsrv50
DB2 Application Development Tools (ADT)	db2_05_00.adt.rte	db2adt50
DB2 ADT Sample Programs	db2_05_00.adt.samples	db2adts50
Code Page Conversion Tables - Uni Code Support	db2_05_00.cnvucs	db2cucs50
Code Page Conversion Tables - Japanese	db2_05_00.conv.jp	db2cnvj50
Code Page Conversion Tables - Korean	db2_05_00.conv.kr	db2cnvk50

Table 25 (Page 2 of 2). DB2 Components, Packages or Filesets

Description	Fileset Name	
	DB2 for AIX	**DB2 for Solaris**
Code Page Conversion Tables - Simplified Chinese	db2_05_00.conv.sch	db2cnvc50
Code Page Conversion Tables - Traditional Chinese	db2_05_00.conv.tch	db2cnvt50
DB2 Product Messages - %L [1, 2]	db2_05_00.msg.%L[4]	db2ms%L50
DB2 Product Document (HTML) - %L [1]	db2_05_00.html.%L[4]	db2ht%L50

Notes:

1. %L in the fileset name represents the locale name. There is a separate fileset for each locale. For Solaris systems, abbreviated names are used for some of the locales.
2. English messages are always installed.
3. DB2 supports a number of locales. However, DB2 messages are not translated in every supported locale. For a complete list of DB2 supported locales, refer to the *"National Language Support (NLS)"* appendix in the *Administration Guide*.
4. On AIX, DB2 Product Messages and Documentation are translated in several locales. However, not every message catalog or book is translated in every locale. The DB2 Product Messages and Documentation are translated into the following locales: bg_BG, cs_CZ, Da_DK, da_DK, De_DE, de_DE, En_US, en_US, Es_ES, es_ES, Fi_FI, fi_FI, Fr_FR, fr_FR, hu_HU, Ja_JP, ja_JP, ko_KR, No_NO, no_NO, pl_PL, pt_BR, ru_RU, sl_SI, Sv_SE, sv_SE, zh_CN, zh_TW, Zh_TW.

Products and Selectable Components

Table 26 lists the DB2 Universal Database products and selectable components that you can install on a DB2 Server.

Table 26 (Page 1 of 2). Components for DB2 Products

Product / Component Description	DB2 Universal Database Extended Enterprise Edition	DB2 CAE
DB2 Client Application Enabler	√√	√√
Open Database Connectivity (ODBC) support	√	√
Java Database Connectivity (JDBC) support	√	√
Code Page Conversion Tables - Unicode Support	√√	√
Administration Server	√√	

Table 26 (Page 2 of 2). Components for DB2 Products

Product / Component Description	**DB2 Universal Database Extended Enterprise Edition**	**DB2 CAE**
DB2 Run-time Environment	√√	
DB2 Engine	√√	
DB2 Parallel Extensions	√√	
DB2 Connect	√√	
DB2 Sample Database Source	√√	
Communication Support for DRDA Application Server	√√	
Communication Support for IPX/SPX	√√	
Communication Support for SNA	√√	
Communication Support for TCP/IP	√√	
Code Page Conversion Tables - Japanese	√	
Code Page Conversion Tables - Korean	√	
Code Page Conversion Tables - Simplified Chinese	√	
Code Page Conversion Tables - Traditional Chinese	√	
License Support for DB2 Universal Database Extended Enterprise Edition	√√	
DB2 Replication	√	
DB2 Product Messages (non-English)[1]	o	o
DB2 Product Library (HTML)[1]	o	o

√√ This is a required component that must be installed.

√ This is a recommended component that is installed by default. You can choose not to install this component.

o This is an optional component that is not installed by default. If you want to install it, you must select it.

Note: There is a separate component of the DB2 Product Messages and the DB2 Product Library for each locale.

Prerequisites

The DB2 filesets are listed below with their corresponding prerequisites.

Table 27. Prerequisites for DB2 Filesets

Fileset Description	Prerequisite Fileset
DB2 Client Application Enabler	Appropriate version of the operating system
Open Database Connectivity (ODBC) support	DB2 Client Application Enabler
Java Database Connectivity (JDBC) support	DB2 Client Application Enabler
Administration Server	DB2 Run-time Environment
DB2 Run-time Environment	DB2 Client Application Enabler
DB2 Sample Database Source	DB2 Engine
DB2 Parallel Extension	DB2 Engine
DB2 Engine	DB2 Run-time Environment
DB2 Replication	DB2 Run-time Environment
DB2 Connect	DB2 Run-time Environment
Communication Support for TCP/IP	DB2 Run-time Environment
Communication Support for IPX/SPX	DB2 Run-time Environment
Communication Support for SNA	Communication Support for TCP/IP
Communication Support for DRDA Application Server	Communication Support for SNA
Licensed Support for DB2 Universal Database: Extended Enterprise Edition	DB2 Engine
DB2 Application Development Tools (ADT)	DB2 Client Application Enabler
DB2 ADT Sample Programs	DB2 Application Development Tools
DB2 Product Messages	DB2 Client Application Enabler
DB2 Product Library (HTML)	DB2 Client Application Enabler

Appendix E. National Language Support (NLS)

This section contains information about the National Language Support (NLS) provided by DB2, including information about supported locales and code sets. For more detailed information on developing applications, refer to the *Embedded SQL Programming Guide*.

Language and Codeset Support for UNIX Operating Systems

DB2 supports many code sets and locales without translating the messages for the corresponding languages. Supporting a locale means that you can create and use a database in that locale, but you may have to view all panels and messages in a different language, if translated messages are not available in DB2. For a complete list of locales supported, refer to the *Administration Guide*.

If you want to operate in a different language environment, do the following:

1 Ensure that the appropriate message option for the desired language has been installed.

2 Set the *LANG* environment variable to the desired locale.

For example, to use `fr_FR` messages on DB2 for AIX, you must have the `fr_FR` message option installed and must set *LANG* to `fr_FR`.

The selected message catalog filesets are placed in the */usr/lpp/db2_05_00/msg/%L* directory on the target workstation, where *%L* is equal to the locale name of the message catalog.

Code Page and Language Support for OS/2 and Windows Operating Environments

During installation of DB2, the country, codepage and regional settings are established. You can, however, change these settings after installing DB2. This includes changing regional settings such as code page, country language (for monetary, date, and numeric formatting) and time zone. When a new connection to a database is made, the database manager uses these new values.

> You must ensure that you install the DBCS (double-byte character set) version of DB2 on the corresponding DBCS version of the operating system. Also, note that DB2 national language support may not produce the expected results if the country, code page and regional settings are incorrect for the intended language.

Table 28 shows the languages into which the DB2 messages are translated.

Note: The code page values in the table that follows are also used as directory names on DB2 CD-ROMs. For example, a reference to *x:\language\win16\install* would be *x:\EN\win16\install* for English.

Table 28. Languages and Code Pages

Country Code	Language	Country Code	Language
BG	Bulgarian	HU	Hungarian
BR	Brazilian Portuguese	IL	Hebrew
CN	Simplified Chinese (PRC)	IT	Italian
CZ	Czech	JP	Japanese
DE	German	KR	Korean
DK	Danish	NO	Norwegian
EN	English	PL	Polish
ES	Spanish	RU	Russian
FI	Finnish	SE	Swedish
FR	French	SI	Slovenian
GR	Greek	TW	Traditional Chinese (Taiwan)

Appendix F. Naming Rules

Go to the section that describes the naming rules that you require information on:

- “General Naming Rules”
- “Database, Database Alias, and Catalog Node Names” on page 316
- “Object Names” on page 316
- “Usernames, Group Names, and Instance Names” on page 317
- “DB2SYSTEM Naming Rules” on page 319
- “Password Rules” on page 319

General Naming Rules

Unless otherwise specified, all names can include the following characters:

- A through Z

  Note: When used in most names, characters A through Z are converted from lowercase to uppercase.

- 0 through 9
- @
- #
- $
- _ (underscore)

Unless otherwise specified, all names must begin with one of the following characters:

- A through Z
- @
- #
- $

Do not use SQL reserved words to name tables, views, columns, indexes, or authorization IDs. A list of SQL reserved words is included in the *SQL Reference*.

Database, Database Alias, and Catalog Node Names

Database names are the identifying names assigned to databases in the database manager.

Database alias names are synonyms given to remote databases. Database aliases must be unique within the System Database Directory in which all aliases are stored.

When naming a database or database alias, see “General Naming Rules” on page 315.

In addition, the name you specify can contain 1 to 8 characters.

Notes:

1. To avoid potential problems, do not use the special characters @, #, and $ in a database name if you intend to have a client remotely connect to a host database. Also, because these characters are not common to all keyboards, do not use them if you plan to use the database in another country.

Object Names

Database objects include:

- Tables
- Views
- Columns
- Indexes
- User-defined functions (UDFs)
- User-defined types (UDTs)
- Triggers
- Aliases
- Table spaces
- Schemas

When naming database objects, see “General Naming Rules” on page 315.

In addition, the name you specify:

- Can contain 1 to 18 characters
- Cannot be any of the SQL reserved words that are listed in the *SQL Reference*.

Note: Using delimited identifiers, it is possible to create an object that violates these naming rules; however, subsequent use of the object could result in errors.

For example, if you create a column with a + or − sign included in the name and you subsequently use that column in an index, you will experience problems when you attempt to reorganize the table. To avoid potential problems with the use and operation of your database, *do not* violate the above rules.

Usernames, Group Names, and Instance Names

Usernames are the identifiers assigned to individual users. When naming users, groups, or instances, see “General Naming Rules” on page 315.

In addition, the name you specify:

- Can contain 1 to 8 characters
- Cannot be any of the following:
  - USERS
  - ADMINS
  - GUESTS
  - PUBLIC
  - LOCAL
- Cannot begin with:
  - IBM
  - SQL
  - SYS
- Cannot include accented characters
- In general, when naming users, groups, or instances:
  - On OS/2, use uppercase names
  - On Windows 95 and Windows NT, use any case
  - On UNIX, use lowercase names

Workstation Names (nname)

A workstation name are used to specify the NetBIOS name for a database server or database client that resides on the local workstation. This name is stored in the database manager configuration file. The workstation name is known as the *workstation nname*. When naming workstations, see "General Naming Rules" on page 315.

In addition, the name you specify:

- Can contain 1 to 8 characters
- Cannot include &, #, and @
- Must unique within the network

In an partitioned database system, there is still only one workstation *nname* that represents the entire partitioned database system, but each node has its own derived unique NetBIOS *nname*.

The workstation *nname* that represents the partitioned database system is stored in the instance-owning database partition server's database manager configuration file.

Each node's unique *nname* is a derived combination of the workstation *nname* and the node number.

For a node that does not own an instance, its NetBIOS *nname* is derived as follows:

1 The first character of the instance-owning machine's workstation *nname* is used as the first character of the node's NetBIOS *nname*.

2 The next 1 to 3 characters represent the node number. The range is from 1 to 999.

3 The remaining characters are taken from instance-owning machine's workstation *nname*. The number of remaining characters depend on the length of the instance-owning machine's workstation *nname*. This number can be from 0 to 4.

For example:

Instance-Owning Machine's *Workstation nname*	**Node Number**	**Derived Node NetBIOS** *nname*
GEORGE	3	G3ORGE
A	7	A7
B2	94	B942
N0076543	21	N216543
GEORGE5	1	G1RGE5

Note: If you have changed the default workstation *nname* during the installation, the workstation *nname*'s last 4 characters should be unique across the NetBIOS network to minimize the chance of deriving a conflicting NetBIOS *nname*.

DB2SYSTEM Naming Rules

DB2SYSTEM names are used by DB2 to identify a physical DB2 machine, system, or workstation within a network. On UNIX, the DB2SYSTEM name defaults to the TCP/IP hostname. On OS/2, you must specify the *DB2SYSTEM* name during install. On Windows 95 and Windows NT, you do not need to specify a *DB2SYSTEM* name; the DB2 setup program detects the NT Computer name and assigns it to *DB2SYSTEM*.

When creating a *DB2SYSTEM* name, see “General Naming Rules” on page 315.

In addition, the name you specify:

- Must be unique within a network
- Can contain a maximum of 21 characters

Password Rules

When determining passwords on OS/2 and Windows 3.x, the password you specify can consist of a maximum of 8 characters. On all other operating systems, the word or name you specify can consist of a maximum of 18 characters.

Appendix G. Notices

Any reference to an IBM licensed program in this publication is not intended to state or imply that only IBM's licensed program may be used. Any functionally equivalent product, program or service that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program, or service. Evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, is the user's responsibility.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the

IBM Director of Licensing,
IBM Corporation,
500 Columbus Avenue,
Thornwood, NY, 10594
USA.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Canada Limited
Department 071
1150 Eglinton Ave. East
North York, Ontario
M3C 1H7
CANADA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

This publication may contain examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

Trademarks

The following terms are trademarks or registered trademarks of the IBM Corporation in the United States and/or other countries:

ACF/VTAM
ADSTAR
AISPO
AIX
AIXwindows
AnyNet
APPN
AS/400
CICS
C Set++
C/370
DATABASE 2
DatagLANce
DataHub
DataJoiner
DataPropagator
DataRefresher
DB2
Distributed Relational Database Architecture
DRDA
Extended Services
FFST
First Failure Support Technology
IBM
IMS
Lan Distance
MVS/ESA
MVS/XA
NetView
OS/400
OS/390
OS/2
PowerPC
QMF
RACF
RISC System/6000
SAA
SP
SQL/DS
SQL/400
S/370
System/370
System/390
SystemView
VisualAge
VM/ESA
VSE/ESA
VTAM
WIN-OS/2

Trademarks of Other Companies

The following terms are trademarks or registered trademarks of the companies listed:

C-bus is a trademark of Corollary, Inc.

HP-UX is a trademark of Hewlett-Packard.

Java, HotJava, Solaris, Solstice, and Sun are trademarks of Sun Microsystems, Inc.

Microsoft, Windows, Windows NT, and the Windows 95 logo are trademarks or registered trademarks of Microsoft Corporation.

PC Direct is a trademark of Ziff Communications Company and is used by IBM Corporation under license.

SCO is a trademark of The Santa Cruz Operations.

SINIX is a trademark of Siemens Nixdorf.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Other company, product, or service names, which may be denoted by a double asterisk (**), may be trademarks or service marks of others.

Index

Contacting IBM

This section lists ways you can get more information from IBM.

If you have a technical problem, please take the time to review and carry out the actions suggested by the *Troubleshooting Guide* before contacting DB2 Customer Support. Depending on the nature of your problem or concern, this guide will suggest information you can gather to help us to serve you better.

For information or to order any of the DB2 Universal Database products contact an IBM representative at a local branch office or contact any authorized IBM software remarketer.

Telephone

If you live in the U.S.A., call one of the following numbers:

- 1-800-237-5511 to learn about available service options.
- 1-800-IBM-CALL (1-800-426-2255) or 1-800-3IBM-OS2 (1-800-342-6672) to order products or get general information.
- 1-800-879-2755 to order publications.

For information on how to contact IBM outside of the United States, see Appendix A of the IBM Software Support Handbook. You can access this document by selecting the "Roadmap to IBM Support" item at: http://www.ibm.com/support/.

Note that in some countries, IBM-authorized dealers should contact their dealer support structure instead of the IBM Support Center.

World Wide Web
http://www.software.ibm.com/data/
http://www.software.ibm.com/data/db2/library/

The DB2 World Wide Web pages provide current DB2 information about news, product descriptions, education schedules, and more. The DB2 Product and Service Technical Library provides access to frequently asked questions, fixes, books, and up-to-date DB2 technical information. (Note that this information may be in English only.)

Anonymous FTP Sites
ftp.software.ibm.com

Log on as anonymous. In the directory /ps/products/db2, you can find demos, fixes, information, and tools concerning DB2 and many related products.

Internet Newsgroups
comp.databases.ibm-db2, bit.listserv.db2-l

These newsgroups are available for users to discuss their experiences with DB2 products.

CompuServe
GO IBMDB2 to access the IBM DB2 Family forums

All DB2 products are supported through these forums.

To find out about the IBM Professional Certification Program for DB2 Universal Database, go to http://www.software.ibm.com/data/db2/db2tech/db2cert.html

IBM®

Part Number: 99H8314

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

99H8314

S99H-8314-00