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

Document Number S09L-6713-00

IBM DB2 Universal Database
Extended Enterprise Edition
for Windows NT**

Quick Beginnings

Version 5

S09L-6713-00

IBM DB2 Universal Database
Extended Enterprise Edition
for Windows NT**

Quick Beginnings

Version 5

S09L-6713-00

Before using this information and the product it supports, be sure to read the general information under Appendix G, "Notices" on page 245.

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 ix
Conventions x
Road Map x

Part 1. Introduction to DB2 Universal Database 1

Chapter 1. About DB2 Universal Database 3
Working with Data Using DB2 Universal Database 5
Accessing DB2 Data from Remote Clients 5
Accessing Multiple DB2 Servers 7
Administering Databases with DB2 Tools 8
Managing Databases Using the Control Center 8
Understanding the Administration Server 10
Monitoring Databases Using DB2 Performance Monitor 11
Viewing SQL Access Plans Using Visual Explain 12
Managing Communications on the Server 12
Managing Connections to Databases Using the Client Configuration Assistant . . 12
Developing Applications Using the DB2 Software Developer's Kit 13
Accessing Host Data from the Desktop Using DB2 Connect Enterprise Edition . 14
Accessing DB2 Data from the Web 16

Chapter 2. What's New in DB2 Version 5? 19

Chapter 3. Introduction to DB2 Extended Enterprise Edition 21
Processing in the Partitioned Database Environment 22
Cost-Based Query Optimization 22
Configuration 24
Machines and Storage 25
Nodegroups and Data Partitioning 25
Multiple Logical Nodes 28
Instances 29
Fast Communication Manager 30
Agents 31

Part 2. Installing DB2 Server 33

Chapter 4. Planning for Installation 35
Memory Requirements 35
Configuration for Accessing DB2 Servers on the LAN 35
Configuration for Accessing LAN and Host Data 37
Configuration for Accessing from the Web 39
Disk Requirements 39

Estimating Fixed Disk Requirements 39
Software Requirements 41
Server Product Requirements 41
Client Product Requirements 42
Possible Client-to-DB2 Server Connectivity Scenarios 43
Understanding System Administrative Authority 44

Chapter 5. Installing and Configuring DB2 Extended Enterprise Edition . . 45
Before You Begin 45
Pre-installation Requirements 45
Determining Usernames for DB2 Installation and Configuration 46
Performing the Installation 48
Installing DB2 Extended Enterprise Edition 48
Verifying the Installation 51
Creating a Server Access Profile 53

Part 3. Installing Clients 55

Chapter 6. Installing DB2 Clients 57

Chapter 7. Installing DB2 Clients on Windows 95 or Windows NT Workstations 59
Before You Begin 59
Performing the Installation 59

Chapter 8. Installing DB2 Clients on OS/2 Workstations 63

Chapter 9. Installing DB2 Clients on Windows 3.x Workstations 65

Part 4. Configuring DB2 Communications 69

Chapter 10. Configuring Client-to-Server Communications Using the CCA . 71
Verifying the Connection 73

Chapter 11. Configuring Server Communications Using the Control Center 75
Modifying the DB2 Communication Configuration of Server Instances 76

Part 5. Using DB2 Universal Database 77

Chapter 12. Getting Started with DB2 Universal Database 79
Logging on to the System 79
Creating the Sample Database 80
Connecting to a Database 80
Working with the System Administrative Group 82
Software Registration 82
Upgrading from a Try-and-Buy License 82

Chapter 13. Working with Instances 85
Instance Directory 85
Creating Instances 86
Adding a Database Partition Server to an Instance 87
Dropping a Database Partition Server from an Instance 88
Listing Database Partition Servers in an Instance 89
Listing Instances 90
Setting the Current Instance 90
Starting and Stopping an Instance 91
Running Multiple Instances Concurrently 92
Removing Instances 92

Chapter 14. Using the Administration Server 93
Creating an Administration Server 93
Setting Up the Administration Server to Use the CCA and the Control Center . . 94
Update Instance Lists 94
Update the Administration Server Configuration 95
Starting the Administration Server 96
Stopping the Administration Server 96
Setting the Logon Account for the Administration Server 96
Removing the Administration Server 97

Chapter 15. Running Your Own Applications 99
Binding Database Utilities 99
Running CLI/ODBC Programs 101
Platform Specific Details for CLI/ODBC Access 102
Detailed Configuration Information 107
Running Java Programs 107
Configuring the Environment 107
Java Applications 108
Java Applets 108

Chapter 16. Entering DB2 Commands and SQL Statements 109
Using the Command Center 109
Using the Command Line Processor 110
Interactive Input Mode 111
Command Line Mode 112
Connecting to a Logical Node 113

Chapter 17. Controlling Your DB2 Environment 115
Controlling the DB2 Profile Registry 116
Setting Your Environment on OS/2 117
Setting Your Environment on Windows 95 and Windows NT 118
Setting Your Environment on Windows 3.x 119
DB2 Registry Values and Environment Variables 120
Configuration Parameters 127
Setting Database Manager Configuration Parameters on a Client Instance . . 127

Chapter 18. Using the Windows NT Performance Monitor 129
Registering DB2 with the Windows NT Performance Monitor 129
Enabling Remote Access to DB2 Performance Information 130
Displaying DB2 Performance Values 131
Accessing Remote DB2 Performance Information 132
Resetting DB2 Performance Values 132

Part 6. Advanced Configuration 133

Chapter 19. Administering Clients Using the CCA 135
Configuring Database Connections 136
Using an Access Profile 136
Searching the Network for Databases 139
Manually Configure a Connection to a DB2 Database 144

Chapter 20. Setting up Communications on the Server Using the Command Line Processor 147
Setting the DB2COMM Registry Value 147
Configuring Named Pipes on the Server 149
Configuring TCP/IP on the Server 150
Step 1. Identify and Record Parameter Values 150
Step 2. Configure the Server 151
Configuring NetBIOS on the Server 153
Step 1. Identify and Record Parameter Values 153
Step 2. Configure the Server 154
Autostarting DB2 with NetBIOS 156
Configuring IPX/SPX on the Server 157
Step 1. Identify and Record Parameter Values 158
Step 2. Configure the Server 159
Configuring APPC on the Server 160
Step 1. Identify and Record Parameter Values 160
Step 2. Configure the Server 160

Chapter 21. Using the DB2 Workstation as a Server for Host Applications . 163
Application Requesters Supported 163
PTFs Required for DRDA Application Servers 164
DRDA-AS Setup for the DRDA Application Requester 164
Setting Up DRDA-AS Functionality 165
Using the Application Server 166
Authentication 166
Troubleshooting 167
DRDA Functions Supported 167
Security and Auditability 174
Configuration Considerations 174

Part 7. Appendixes 175

Appendix A. Obtaining Information . . . 177
How the DB2 Library Is Structured . . . 177
Online Help . . . 177
DB2 Books . . . 179
About the Information Center . . . 183
Ordering the Printed DB2 Books . . . 184
Printing the PostScript Books . . . 185
Viewing Online Books . . . 185
Installing the Netscape Browser . . . 186
Setting up a Document Server . . . 186

Appendix B. Installation Troubleshooting . . . 189

Appendix C. Issuing Commands to Multiple Database Partition Servers . . 191
Commands . . . 191
Command Descriptions . . . 192
Specifying the Command to Run . . . 192
Prefix Sequences . . . 193
Specifying the List of Machines . . . 195
Eliminating Duplicate Entries from the List of Machines . . . 195
Controlling the rah Command . . . 196
Setting the Default Environment Profile . . . 196

Appendix D. Implementing Failover Support in the Windows NT Environment . . . 197
Failover Configurations . . . 199
Hot Standby Configuration . . . 199
Mutual Takeover Configuration . . . 200
Setting up DB2 Groups and Resources from the Cluster Administrator . . . 201
Registering a Resource Type of DB2 . . . 201
Setting up a Hot Standby Configuration . . . 202
Setting up a Mutual Takeover Configuration . . . 206
Setting up the DB2 Instance for Failover Support . . . 211
Creating the DB2 Instance . . . 212
Migrating an Instance to Use MSCS Support . . . 212
Enabling the DB2 Resource to Fail Over to the Second MSCS Node . . . 214
Adding an Additional Database Partition Server to the Instance . . . 214
Disabling DB2 Resource Failover to the Second MSCS Node . . . 216
Failback Considerations . . . 216
Registering Database Drive Mapping for Mutual Takeover Configurations . . . 217
Reconciling Database Drive Mapping . . . 218
Example - Setting up DB2 Failover Support in a Mutual Takeover Configuration . 219
Preliminary Tasks . . . 220
Setting up the db2test1 Machine . . . 221
Enable the DB2 Resource to Fail Over to the db2test2 Machine . . . 223
Register the Database Drive Mapping for ClusterA . . . 224
Bring the DB2 Resources for ClusterA Online . . . 224
Setting up the db2test3 Machine . . . 224

Enable the DB2 Resource to Fail Over to the db2test4 Machine 225
Register the Database Drive Mapping for ClusterB 226
Bring the DB2 Resources for ClusterB Online 226
Administering DB2 in an MSCS Environment 226
Starting and Stopping DB2 Resources 227
Running Scripts 227
Database Considerations 232
User and Group Support 232
Communications Considerations 233
System Time Considerations 234
Administration Server and Control Center Considerations 234
Limitations and Restrictions 236

Appendix E. National Language Support (NLS) 237
Code Page and Language Support for OS/2 and Windows Operating Environments 237

Appendix F. Naming Rules 239
General Naming Rules 239
Database, Database Alias, and Catalog Node Names 240
Object Names 241
Usernames, Group Names, and Instance Names 241
Workstation Names (nname) 242
DB2SYSTEM Naming Rules 243
Password Rules 244

Appendix G. Notices 245
Trademarks 246
Trademarks of Other Companies 246

Index 249

Contacting IBM 251

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 OS/2 and Windows operating systems.

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, using the command line processor for setting up server communications, and using the DB2 workstation as a server for host applications.

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.
Install DB2 Extended Enterprise Edition from a CD-ROM,	Chapter 5, “Installing and Configuring DB2 Extended Enterprise Edition” on page 45.
Install the client products,	Chapter 6, “Installing DB2 Clients” on page 57.
Set up communications on the client to access databases,	Chapter 10, “Configuring Client-to-Server Communications Using the CCA” on page 71.

If you want to...	Refer to...
Use the DB2 components,	Chapter 12, “Getting Started with DB2 Universal Database” on page 79.
Create the sample database,	“Creating the Sample Database” on page 80.
Connect to a database,	“Connecting to a Database” on page 80.
Print or view online documentation,	“Printing the PostScript Books” on page 185 or “Viewing Online Books” on page 185.
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 101.
Work with the System Administrative (SYSADM) group,	“Working with the System Administrative Group” on page 82.
Use the DB2 Command Center or the DB2 command line processor to enter SQL statements or DB2 commands interactively,	Chapter 16, “Entering DB2 Commands and SQL Statements” on page 109.
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 177 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

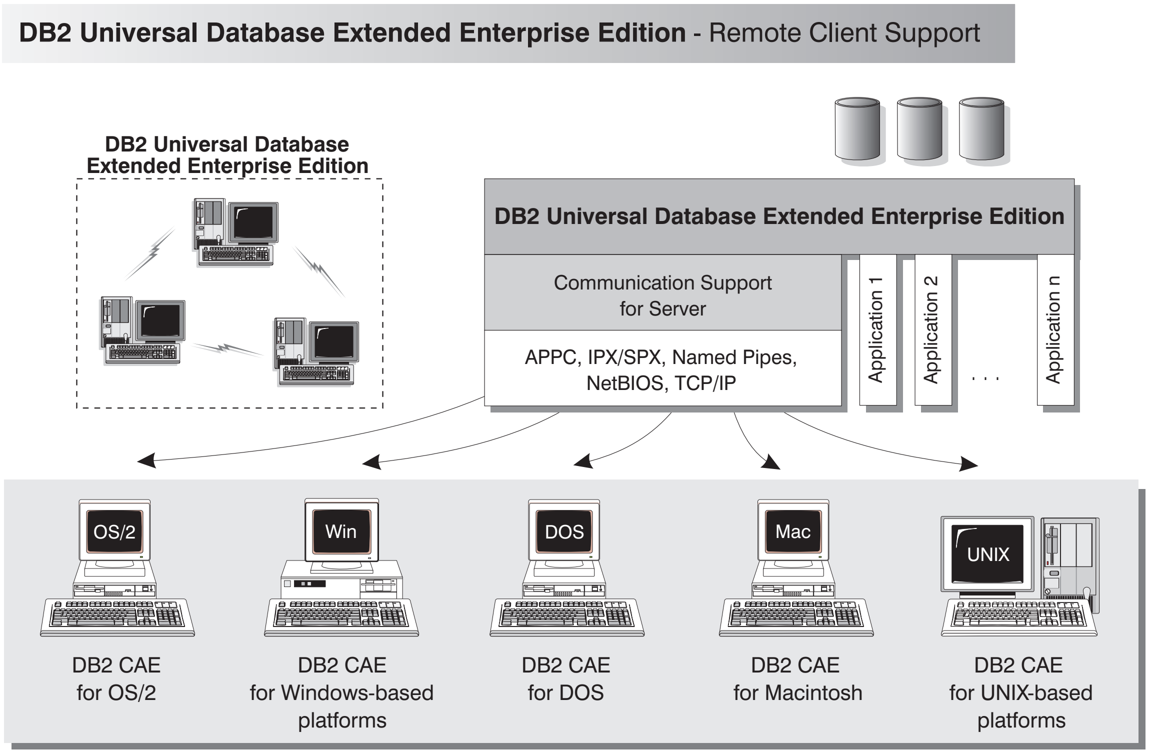
The DB2 Client Application Enabler provides a run-time environment that enables client applications to access one or more remote databases. Local applications and all Java applications (either local or remote) access a database through the DB2 Client Application Enabler component on the server. All remote 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 6, "Installing DB2 Clients" on page 57 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
DB2 Universal Database Extended Enterprise Edition
Communication Support for Server
APPC, IPX/SPX, Named Pipes, NetBIOS, 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 21, “Using the DB2 Workstation as a Server for Host Applications” on page 163, 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.

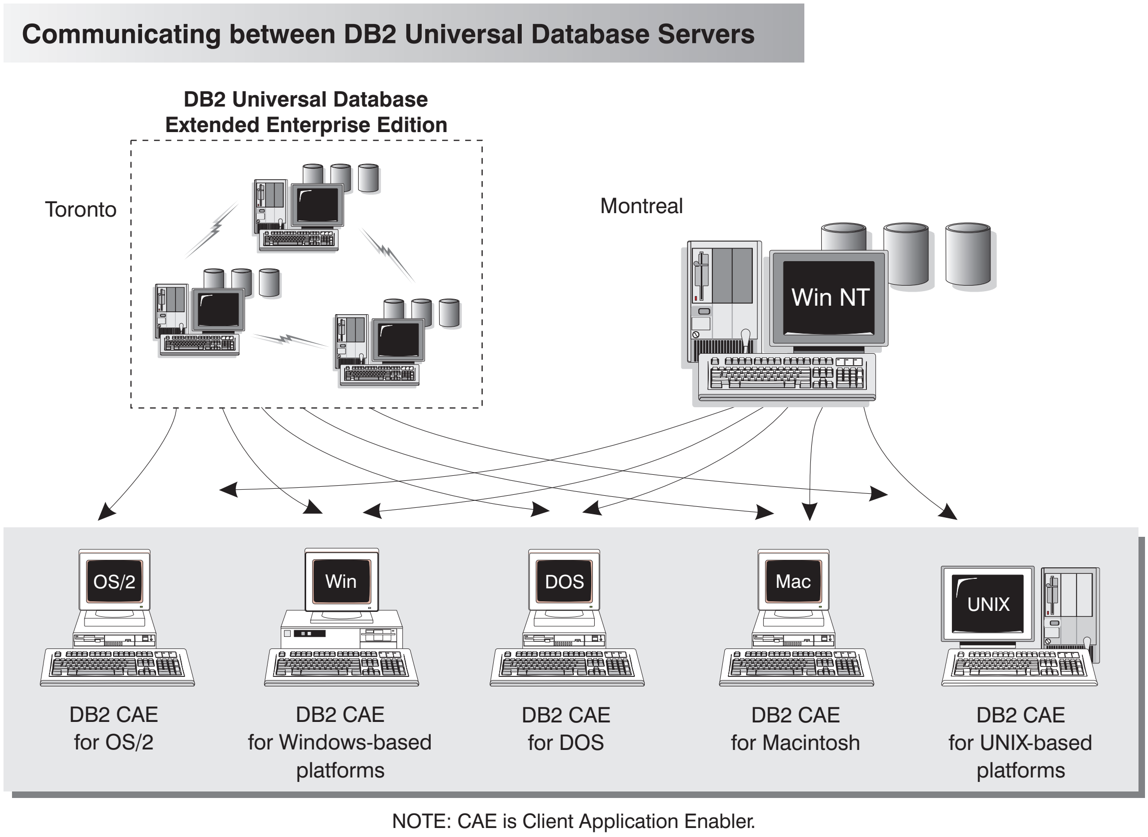


Figure 2. Accessing Data on Multiple Servers

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

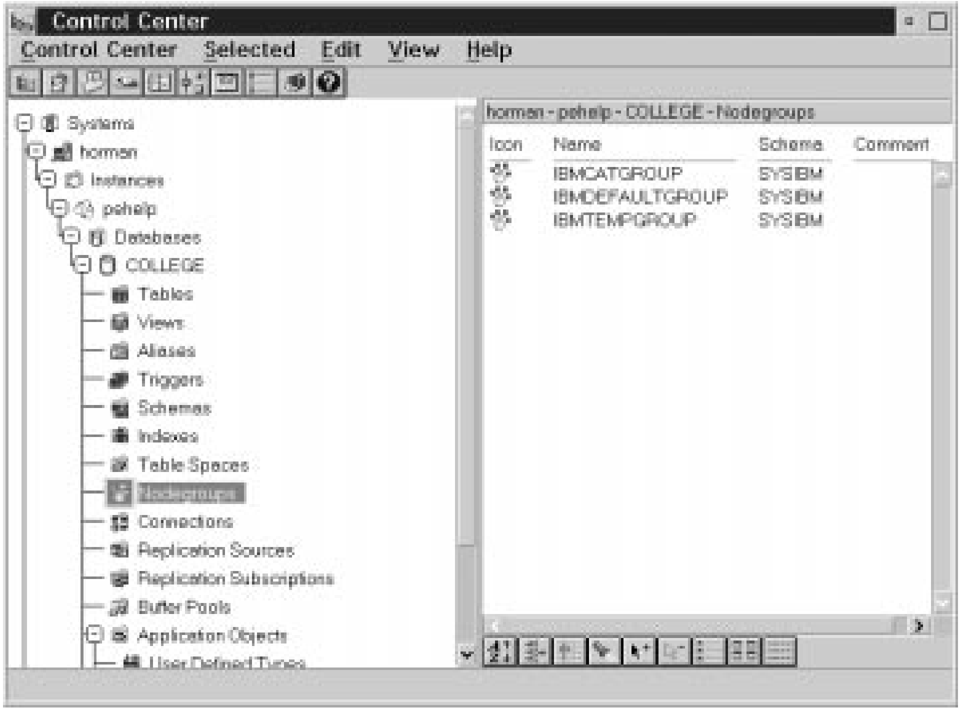
Use the *Control Center* to perform administration tasks such as configuring, backing up and recovering data, managing directories, scheduling jobs, and managing media, all from a graphical interface.

The Control Center uses TCP/IP to retrieve status, connection, and snapshot information from each database partition server (node). You assign a TCP/IP port number for the Performance Monitor component of the Control Center during installation.

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 on page 9 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 17, “Controlling Your DB2 Environment” on page 115.



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. The setup program creates the Administration Server on the instance-owning machine and automatically starts it at boot time. The default name of the Administration Server is DB2DAS00. For more information, see Chapter 14, "Using the Administration Server" on page 93.

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.

> You are also able to use the Windows NT Performance Monitor to monitor both database and system performance. See Chapter 18, “Using the Windows NT Performance Monitor” on page 129 for details on how to register DB2 resources and to use the Windows NT Performance Monitor.

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.

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

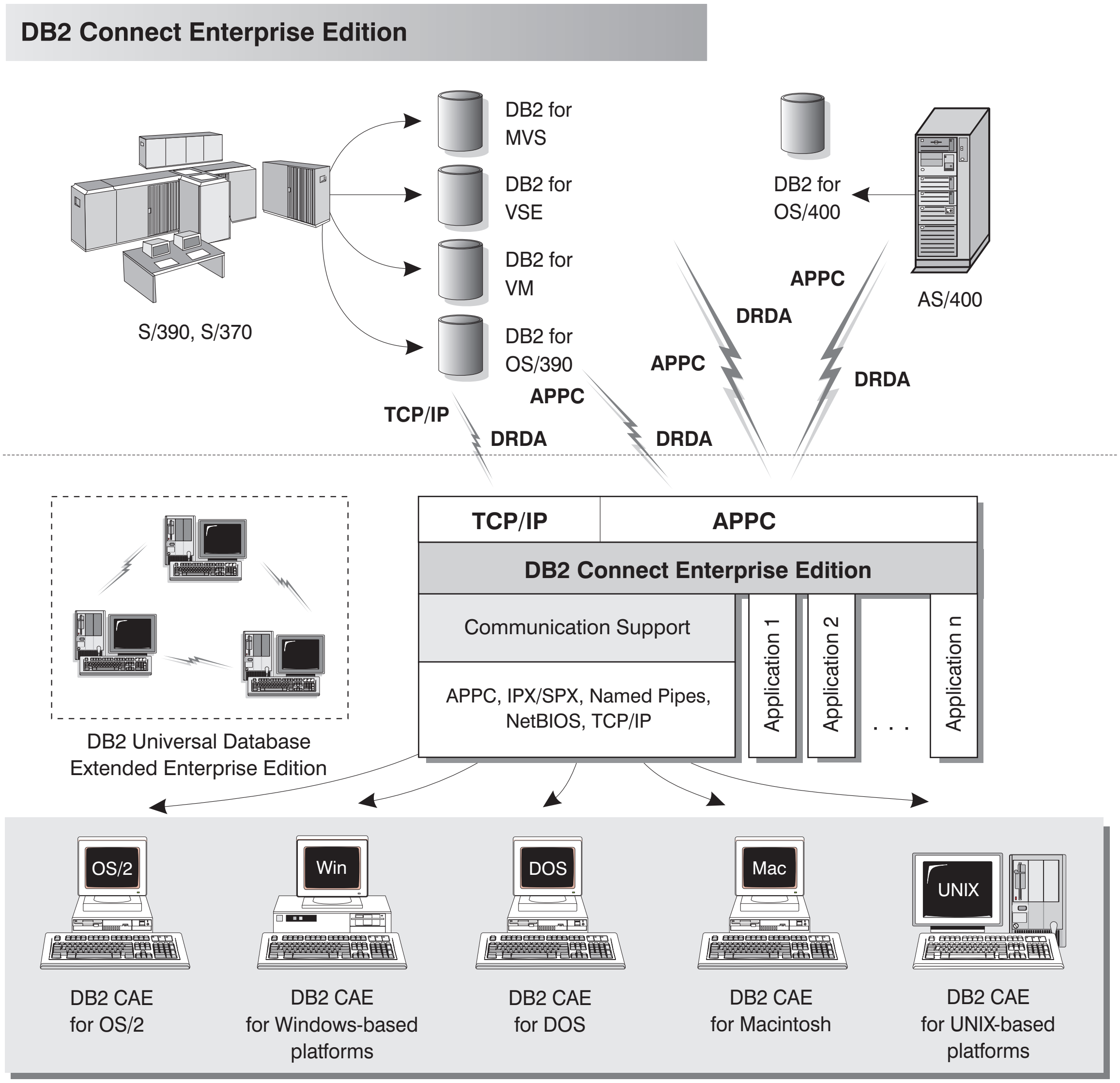
DB2 Connect Enterprise Edition gives clients on your LAN access to data that is stored on host systems. It provides applications with transparent access to host data through a standard architecture for managing distributed data. This standard is known as Distributed Relational Database Architecture (DRDA). 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.



NOTE: CAE is Client Application Enabler. Not all protocols are supported for all the clients.

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.

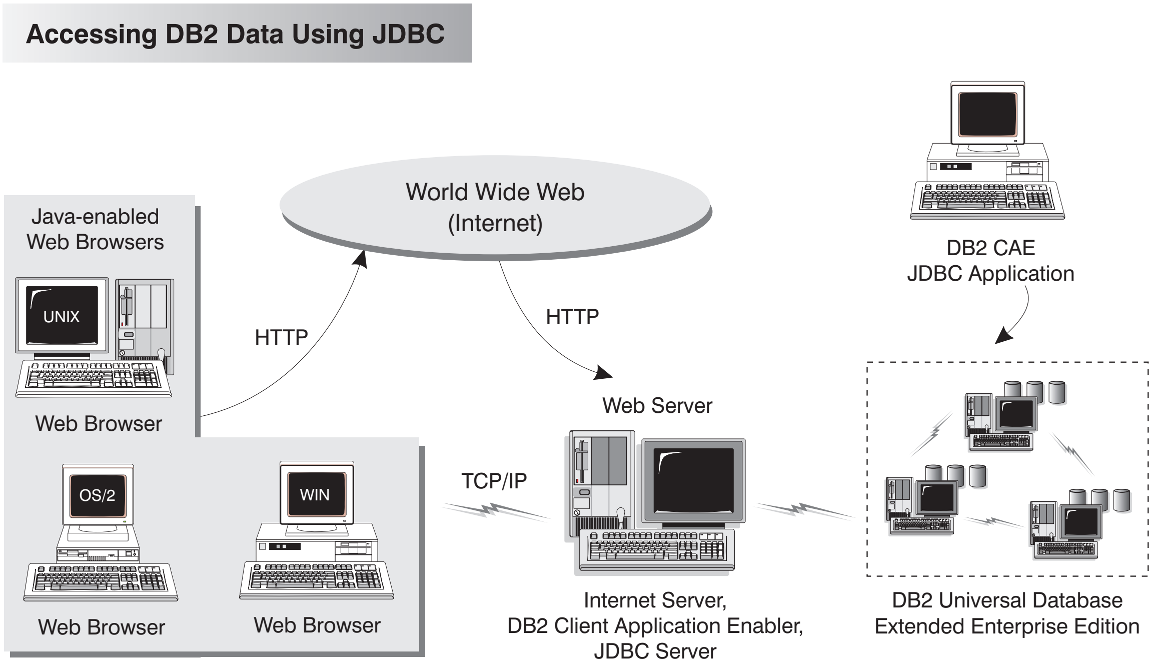


Figure 5. Accessing Internet Data Stored on DB2 Using JDBC

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.

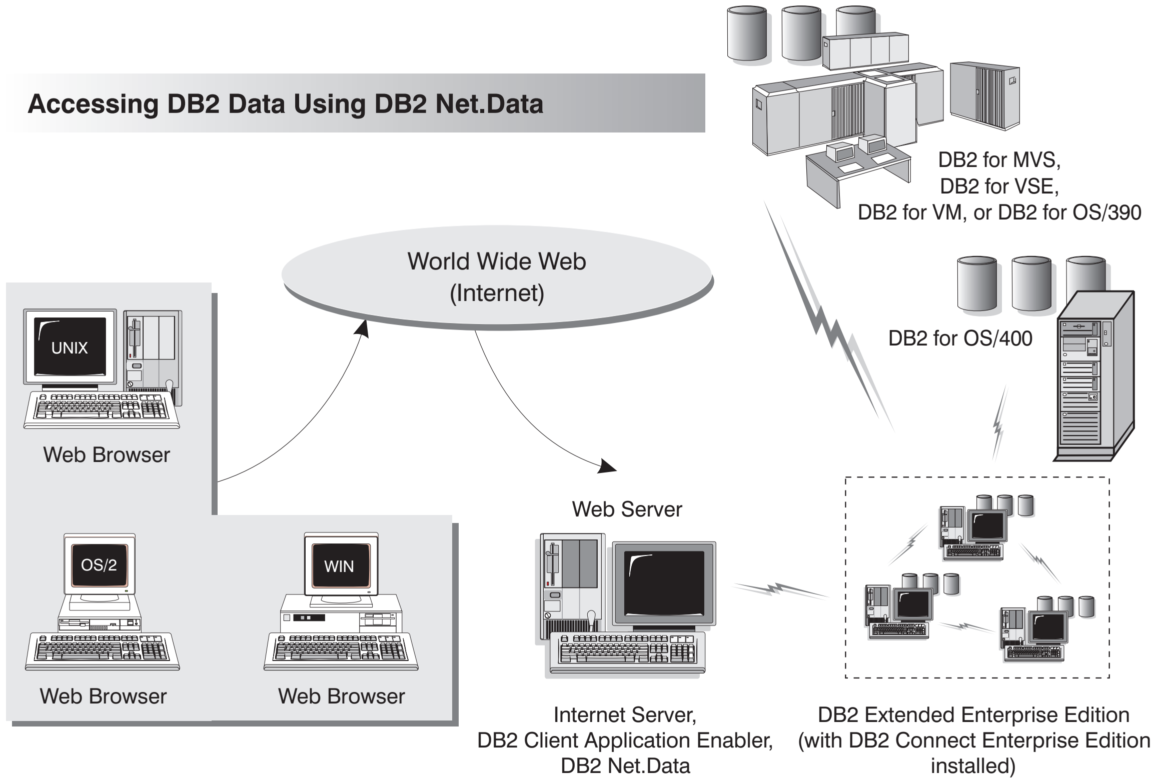

Accessing DB2 Data Using DB2 Net.Data
DB2 for MVS,
DB2 for VSE,
DB2 for VM, or DB2 for OS/390
World Wide Web
(Internet)
DB2 for OS/400
UNIX
Web Browser
Web Server
OS/2
WIN
Web Browser
Web Browser
Internet Server,
DB2 Client Application Enabler,
DB2 Net.Data
DB2 Extended Enterprise Edition
(with DB2 Connect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

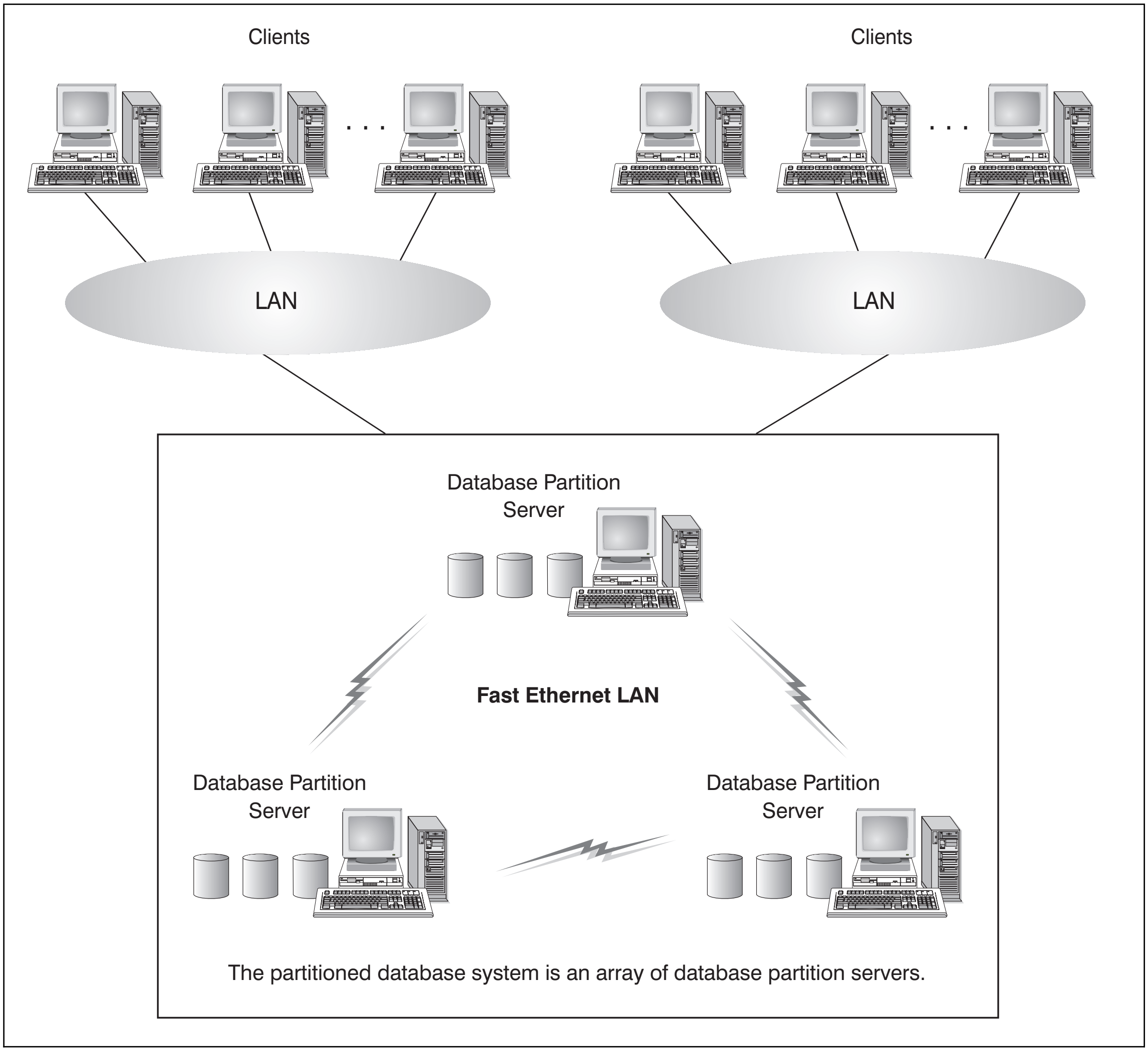

Clients
Clients
LAN
LAN
Database Partition Server
Fast Ethernet LAN
Database Partition Server
Database Partition Server
The partitioned database system is an array of database partition servers.

Figure 7. DB2 Extended Enterprise Edition Hardware Configuration

DB2 Extended Enterprise Edition can run on a cluster of individual machines interconnected by a LAN.

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) in the instance.

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.

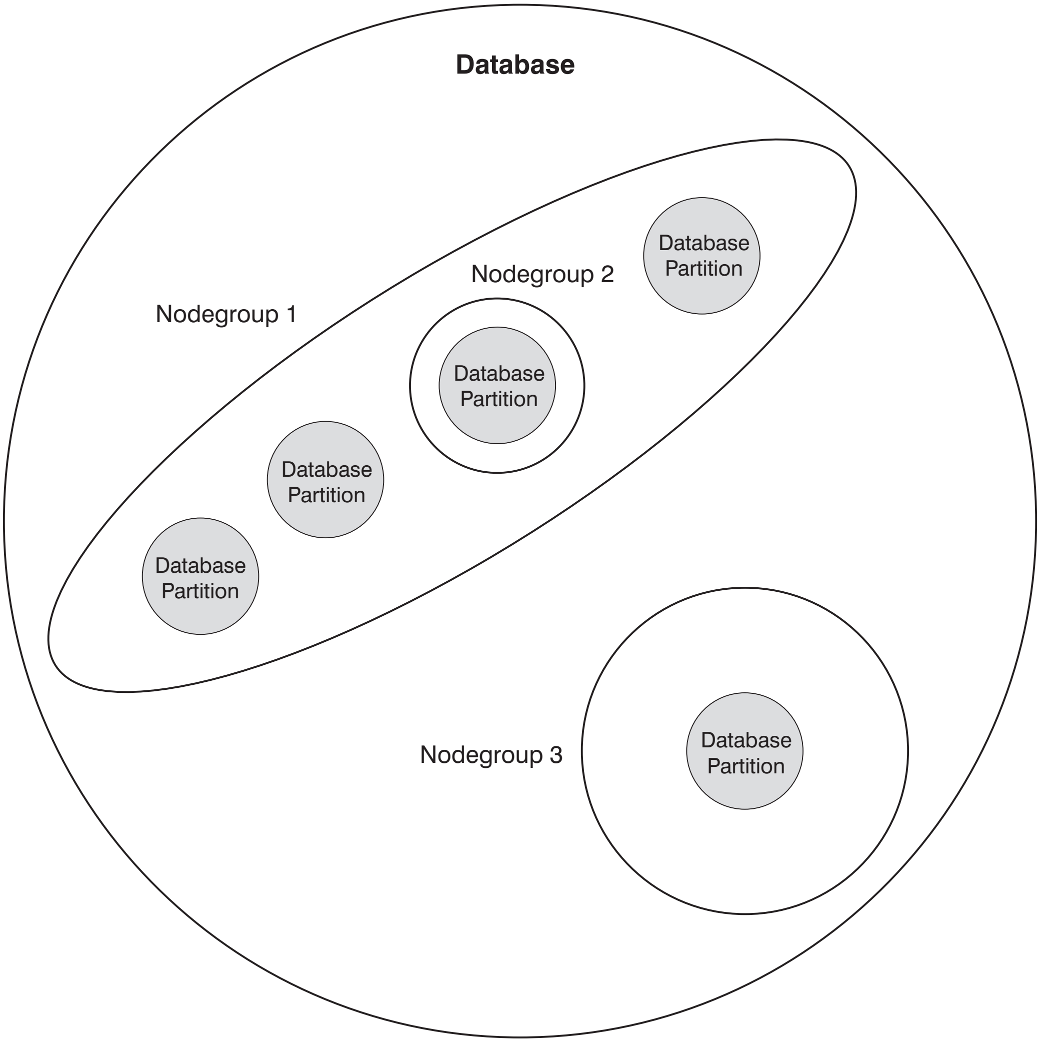


Figure 8. Nodegroups in a Database

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, you can add them to the 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 populated database system, refer to the *Administration Guide*. For information about scaling an empty database system, see “Adding a Database Partition Server to an Instance” on page 87.

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 113.

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. 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 85 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.) Instance 1 does not contain database partition servers that belong to Instance 2, and the converse.

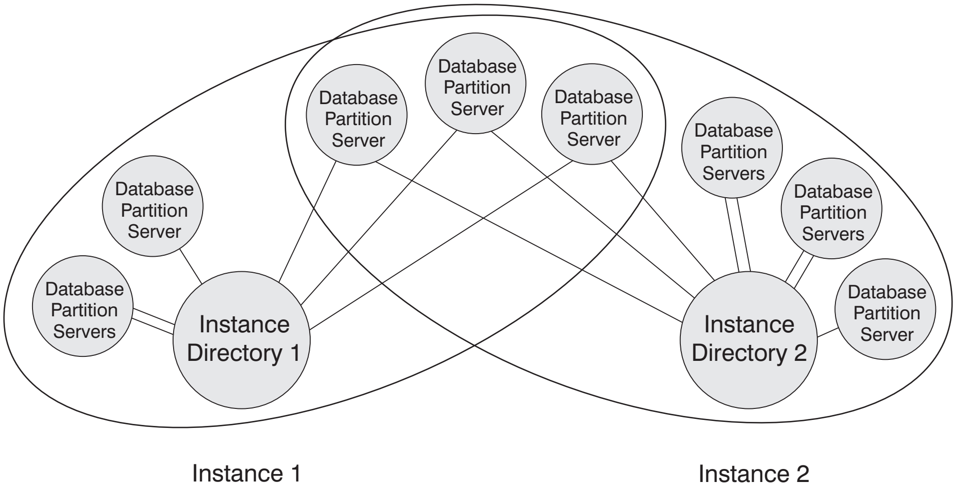

Database Partition Server
Database Partition Server
Database Partition Server
Database Partition Servers
Database Partition Server
Database Partition Servers
Database Partition Servers
Instance Directory 1
Instance Directory 2
Database Partition Servers
Database Partition Server
Instance 1
Instance 2

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 restrict access to specific databases
- To exploit different database configurations

Each instance is owned by the machine where the instance was first created (known as the instance-owning machine). The instance-owning machine stores information that is common to all of the database partition servers. For information about creating instances, see Chapter 13, “Working with Instances” on page 85.

Fast Communication Manager

The fast communication manager (FCM) provides communication support for DB2 Extended Enterprise Edition.

Each database partition server has one FCM thread to provide communications between database partition servers to handle agent requests, and to deliver message buffers. The FCM thread is started when you issue the **db2start** command.

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*.

If communications fail between database partition servers or if they re-establish communications, the FCM thread 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 , 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.
- 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. Installing DB2 Server

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 Extended Enterprise Edition" on page 45.

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, Named Pipes, NetBIOS, or TCP/IP.

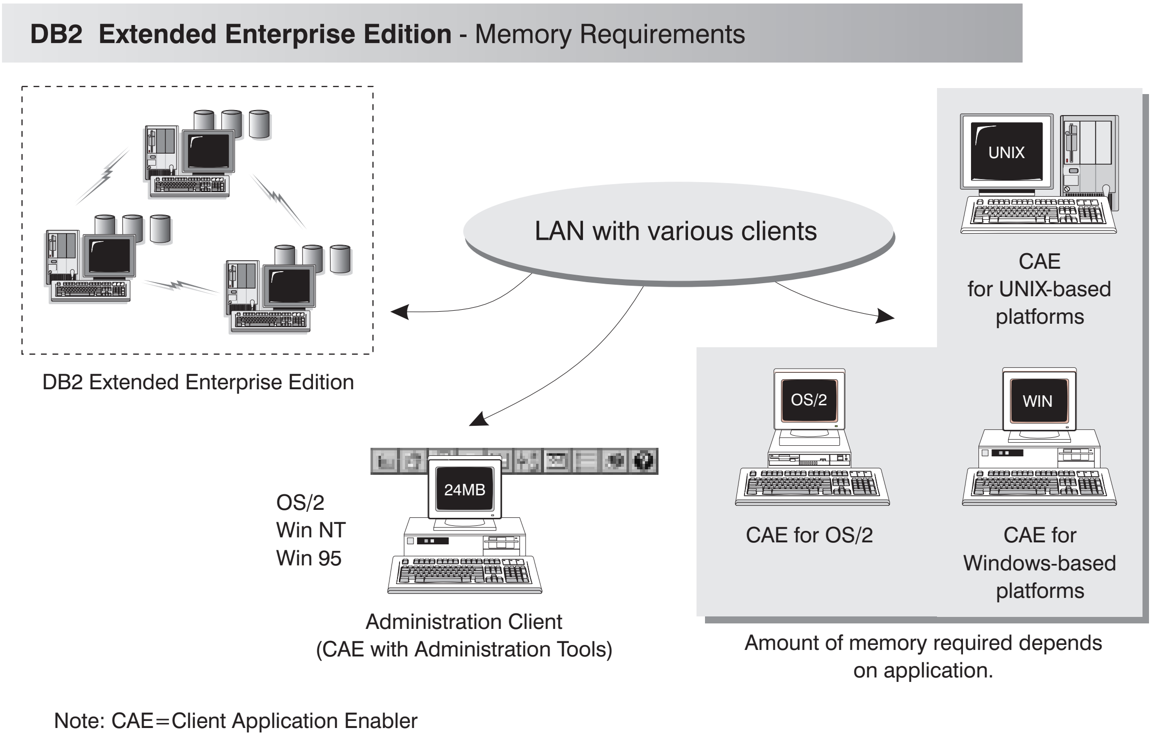

DB2 Extended Enterprise Edition - Memory Requirements
LAN with various clients
UNIX
CAE
for UNIX-based
platforms
DB2 Extended Enterprise Edition
OS/2
WIN
CAE for OS/2
CAE for
Windows-based
platforms
24MB
OS/2
Win NT
Win 95
Administration Client
(CAE with Administration Tools)
Amount of memory required 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. For the server, it is recommended that a minimum of 32 MB of RAM be installed to accommodate five concurrent clients. We suggest that you have a minimum of 48 MB of RAM to accommodate 25 concurrent clients and 64 MB of RAM for 50 concurrent clients. 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, Named Pipes, NetBIOS, 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.)

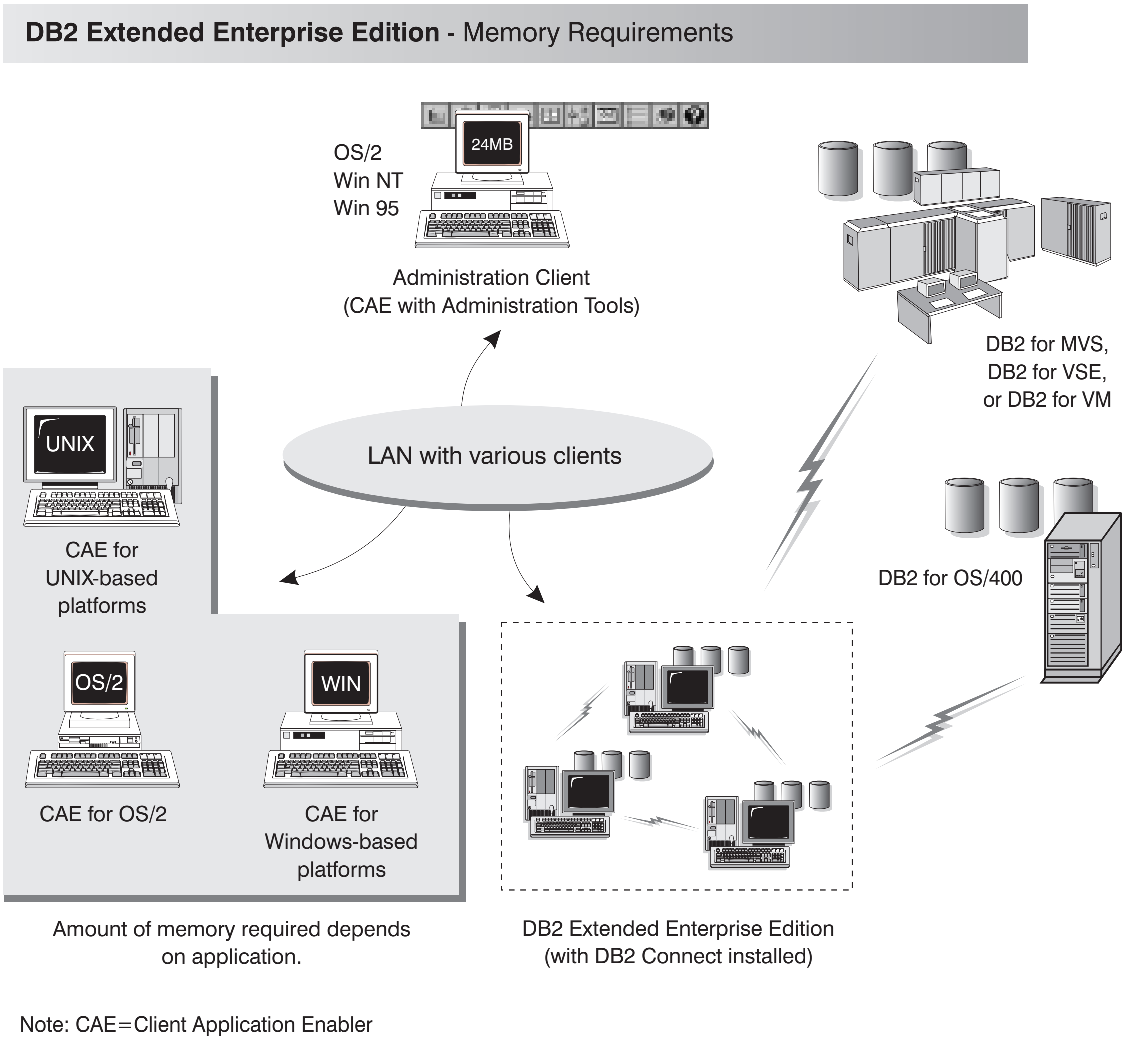

DB2 Extended Enterprise Edition - Memory Requirements
24MB
OS/2
Win NT
Win 95
Administration Client
(CAE with Administration Tools)
DB2 for MVS,
DB2 for VSE,
or DB2 for VM
UNIX
LAN with various clients
CAE for
UNIX-based
platforms
DB2 for OS/400
OS/2
WIN
CAE for OS/2
CAE for
Windows-based
platforms
Amount 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. For the server, it is recommended that a minimum of 32 MB of RAM be installed to accommodate five concurrent clients. We suggest that you have a minimum of 48 MB of RAM to accommodate 25 concurrent clients and 64 MB of RAM for 50 concurrent clients. 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 on page 40 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	
DB2 Universal Database Extended Enterprise Edition — this amount includes the following components:	90 MB
• DB2 Connect Support	2 MB
• Graphical Tools (Client Configuration Assistant, Control Center, Performance Monitor, and Visual Explain)	35 MB
• East Asian Conversion Support	3 MB
Online documentation in HTML format	35 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. Estimating Disk Requirements for Client Components

	Recommended Minimum Disk (MB)
DB2 Client Application Enabler for OS/2	
DB2 Client Application Enabler for OS/2 — this amount includes the following components:	55 MB
• Client Configuration Assistant	5 MB
• Graphical Tools	5 MB
• Control Center	30 MB
• Performance Monitor	10 MB
• Visual Explain	5 MB
Online documentation in HTML format	25 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:	54 MB
• Graphical Tools (Client Configuration Assistant, Control Center, Performance Monitor, and Visual Explain)	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. Software Requirements for DB2 Servers

Product	Operating System	Communications
• DB2 Universal Database Extended Enterprise Edition	• Windows NT Version 4.0 or later	APPC, IPX/SPX, NetBIOS, TCP/IP, or Named Pipes • For APPC connectivity, you require one of: – IBM Communications Server for NT Version 5.0 – Microsoft SNA Server Version 2.11 or later • The Windows NT base operating system provides NetBIOS, IPX/SPX, Named Pipes, and TCP/IP connectivity. **Notes:** 1. DB2 Connect for Windows NT supports APPC, NetBIOS, IPX/SPX, TCP/IP and Named Pipes for LAN connections. 2. For host connectivity, you require IBM Communications Server for Windows NT Version 5.0. **Note:** Windows 95 cannot be used as a server or gateway. 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 with IBM DCE for Windows NT Version 2.0 Release. In addition, DB2 also supports Gradient PC-DCE for Windows NT or 95 Version 2.0 Release Runtime Media Kit. For DB2 Connect support, you require DB2/MVS Version 5.1 plus its prerequisite, OS/390 DCE Base Services version 3 for DCE support. **Note:** 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 have the IBM Antivirus program installed on your operating system, it must be Version 3.0 or later. 6. If you want to use the Simple Network Management Protocol (SNMP) subagent, you require DPI 2.0 provided by IBM SystemView Agent.

Client Product Requirements

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

Table 4 (Page 1 of 2). Software Requirements for Clients

Component	Operating System	Communications
• 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.
• 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.
• 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.

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

Component	Operating System	Communications
• 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 (Page 1 of 2). Possible Client-to-Server Connectivity Scenarios

Client	DB2 for Windows NT Server
OS/2	APPC IPX/SPX[1] NetBIOS TCP/IP
Windows 3.1x	IPX/SPX[1] NetBIOS TCP/IP

Table 5 (Page 2 of 2). Possible Client-to-Server Connectivity Scenarios

Client	**DB2 for Windows NT Server**
Windows 95	IPX/SPX[1] Named Pipes NetBIOS TCP/IP
Windows NT	APPC IPX/SPX[1] Named Pipes NetBIOS TCP/IP
1 Direct 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 Administrators group 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 82.

Chapter 5. Installing and Configuring DB2 Extended Enterprise Edition

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

If you want to install DB2 clients, see Chapter 6, “Installing DB2 Clients” on page 57.

When you are creating a partitioned database system, the installation of DB2 Universal Database Extended Enterprise Edition differs between the first machine and subsequent machines. When you install on the first machine, you will choose to create the database partition server that will become the instance-owning machine. When you perform each subsequent install, you will choose to add a new node to an existing partitioned database system.

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.

Pre-installation Requirements

Before you install DB2 Extended Enterprise Edition, for each of the machines that will make up your partitioned database system, you need to:

- Determine how each of the machines will be used.
- Ensure that all machines are connected to a LAN with TCP/IP configured, so that the machines can connect to each other using TCP/IP.
- Ensure that all machines belong to the same Windows NT domain.
- Ensure that all machines are set to a consistent date and time. In order to be considered consistent, the difference in GMT time between the machines must not exceed the value configured for the *max_time_diff* parameter in the database manager configuration file. The default time for the *max_time_diff* configuration parameter is 1 hour. For information on setting database manager configuration parameters, refer to the *Administration Guide*.

- Determine the usernames required to install DB2 and its components. (See “Determining Usernames for DB2 Installation and Configuration” on page 46.)
- Reserve a TCP/IP port range on each machine, to be used by the Fast Communication Manager (FCM). FCM is a feature of DB2 that handles communications between database partition servers.

  To determine the port range, check the services file on each of the machines on which the database will reside. The services file is located in the *x:/winnt/system32/drivers/etc* directory, where *x:* is the drive on which you installed Windows NT. This file lists the port numbers that are already in use. Determine a port range that is available on all of the machines. You will need to specify this port range during installation.

  Note: If any of the machines that will make up your partitioned database system will be running multiple logical nodes (MLNs), then the port range you set must be equal to or greater than the number of logical nodes on the machine running the most MLNs.

  If you are not planning to run multiple logical nodes, the port range you specify only needs to span two ports.
- Reserve a TCP/IP port on each machine, to be used by the DB2 Performance Monitor component of the Control Center. Use the instructions above to determine a port that is available on all machines. The same port must be reserved on all machines that will make up the partitioned database system.

Determining Usernames for DB2 Installation and Configuration

This section describes the usernames required to install DB2 and its components.

Determining a Username for Installing DB2

You need to have a username that will be used to install DB2. This username must meet the following requirements:

- It must be a domain user.
- It must belong to the local Administrators group on all of the machines that will make up the partitioned database system. The local Administrators group must have the *"Act as part of the operating system"* advanced user right.

All of the machines that will make up your partitioned database system must belong to the same local Administrators group.

Determining the Domain Username

You need to have a domain username that will be used to create a database partition server. DB2 will use this domain username to log on when DB2 is started as a Windows NT service. This username must meet the following requirements:

- It must be an existing domain username.
- It must be a valid DB2 username. A valid DB2 username is eight characters or less, and complies with DB2's naming rules. For more information on DB2's naming rules, see Appendix F, “Naming Rules” on page 239.

Determining the Username for the Administration Server

When you create the database partition server that will become the instance-owning machine, the setup program automatically creates the Administration Server. During installation, you will be asked to provide a username and password that will be used by the Administration Server to log on to the system and to start itself as a service.

The username must belong to the local Administrators group. By default, the setup program will default both the username and the password to *db2admin*.

The setup program will check to see if the username specified for the Administration Server exists. If it does not exist, it will be created. If it does exist, the setup program will verify that the username is a member of the Administrators group and that the password supplied is valid.

Notes:

1. If the *db2admin* username already exists on your system, you must use the same password specified for this username.

   If the *db2admin* username uses the password *db2admin*, for security reasons (since this password is the default for any DB2 install and therefore is well known) you should change this password.

2. If the *db2admin* username *does not* already exist, for security reasons (since this password is the default for any DB2 install and therefore is well known) we recommend that you provide your own password. Enter a password in the **Password** and **Confirm Password** fields. You must ensure that it is eight characters or less, and complies with DB2's naming rules.

Performing the Installation

To install DB2 Extended Enterprise Edition, you need to perform steps to set up your system environment. You also need to determine certain information about your system in order to configure your partitioned database system. Before you install, make sure that you have all of the information that you require. See “Before You Begin” on page 45 for a list of pre-installation requirements.

Installing DB2 Extended Enterprise Edition

To install DB2 Extended Enterprise Edition on a Windows NT workstation:

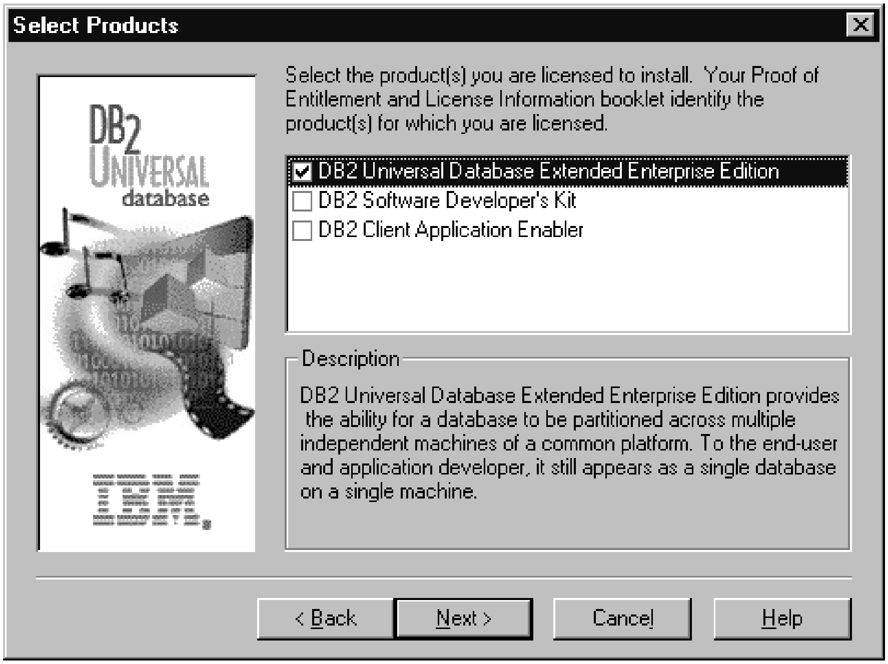
1. Log on as a user that meets the requirements for installing DB2. For more information, see “Determining Usernames for DB2 Installation and Configuration” on page 46.

2. Shut down any other programs so that the setup program can update files as required.

3. Insert the CD-ROM 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 14 on page 238 lists the code for each available language.
> 3. Select **OK**.

4. The Welcome window opens. Click on the **Next** push button to open the Select Products window, as shown below. Select the product that you want to install and click on the **Next** push button.



Select Products

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.

☑ DB2 Universal Database Extended Enterprise Edition
□ DB2 Software Developer's Kit
□ DB2 Client Application Enabler

Description

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.

5 The Select Installation Option window opens.

The setup program allows you to choose different installation options. All of the options cause the setup program to install DB2 Extended Enterprise Edition.

Use the following information to determine the option you want to select.

- If there is no previous version of DB2 on the machine, you have two installation options:

  This machine will be the instance-owning database partition server
  The setup program creates the default instance on what will become the instance-owning database partition server. The default instance is named DB2MPP.

  This machine will be a new node on an existing partitioned database system
  The setup program adds this machine as a node in an existing partitioned database system.

- If there is a previous version of DB2 on the machine, you have three installation options:

  This machine will be the instance-owning database partition server
  If you choose this option, the setup program will ask you if you want to use the existing default instance as the default instance in the partitioned database system. Choose one of the following options:

Yes, use the instance
If you choose to use the existing default instance, the setup program migrates the existing instance to the DB2 Version 5 multipartition format (the default instance retains its name). This machine becomes the instance-owning database partition server. All other existing DB2 instances will be migrated to the DB2 Version 5 single-partition format.

No, create a new default instance
If you choose to create a new default instance, the setup program creates a default instance, called DB2MPP, on what will become the instance-owning database partition server. All existing DB2 instances will be migrated to the DB2 Version 5 single-partition format.

This machine will be a new node on an existing partitioned database system
The setup program adds the machine as a node in an existing partitioned database system. All existing DB2 instances will be migrated to the DB2 Version 5 single-partition format.

This machine will be a single-partition database server
This option does not allow you to exploit the scalability and power of a multipartition database system. All existing DB2 instances are migrated to the DB2 Version 5 single-partition format.

6 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.

7 After you install DB2, you must restart your workstation before you can begin to use it. Select a restart option and click on the **Finish** push button.

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.

Note: The instance directory is located in the *\sqllib* directory. During the installation, the DB2 setup program gives the instance directory the shared name DB2-*INSTANCE_NAME*. Read and write permissions are automatically granted to everyone in the domain. After you install you can change the permissions to restrict access to the directory.

If you want to add another node to this partitioned database system, return to “Before You Begin” on page 45.

The setup program has:

- Created DB2 program groups and items (or shortcuts)
- Registered a security service
- Registered the DB2 Remote Command service
- Registered the DB2 Governor service
- Updated the Windows registry
- Created the DB2 Administration Server on the instance-owning machine, added it as a service, and configured it so that DB2 tools can administer the server. The service's start type was set to **Automatic**.

You have completed all the installation steps. Go to “Verifying the Installation” to verify the installation.

By default, any user belonging to the Administrators group has System Administrative (SYSADM) authority on the instance created or migrated during the installation. To change the group that has SYSADM authority on the default instance, go to “Working with the System Administrative Group” on page 82.

Verifying the Installation

This section describes how you can create a sample database to verify that DB2 is properly installed. If you want, you can verify installation immediately after you create the instance on the instance-owning machine (node 0). However, it is recommended that you wait to create the sample database until after you have added all of the machines that will make up your partitioned database system.

You can verify that DB2 is installed correctly by creating the sample database on your system, and accessing data from the database.

If you installed the DB2 tools, you can verify the installation as follows:

1 Log on to the instance-owning machine with a valid DB2 username that has System Administrative (SYSADM) authority.

2 To invoke First Steps, click on **Start** and select **Programs**->**DB2 for Windows NT**->**First Steps**

3 Click on the **Create the SAMPLE Database** graphic button on the main panel of First Steps.

4. Once the database is created, click on the **View the SAMPLE Database** graphic button on the main panel of First Steps to select data from the SAMPLE database. This starts the Command Center allowing you to use the supplied script that shows some of the data from the database. Click on the execute icon to begin the query.

5. Click on the **Work with the SAMPLE Database** graphic button on the main panel of First Steps to start the Control Center. This allows you to see the tables that are in the SAMPLE database and enables you to perform actions on them.

If you did not install the DB2 tools, you can verify the installation as follows:

1. Log on to the instance-owning machine with a valid DB2 username that has System Administrative (SYSADM) authority.

2. Create the sample database using the **db2sampl** command at the operating system command prompt. Type:

   ```
   db2sampl
   ```

3. Start the database manager by issuing the **db2start** command in the command line processor. Type:

   ```
   db2start
   ```

4. Once the database is created, enter the following commands in the command line processor to connect to the sample database and retrieve a list of all the employees that work in department 20.

   ```
   db2 connect to sample
   db2 "select * from staff where dept = 20"
   ```

After you have verified the installation, you can remove the sample database to free up disk space. You can use the Command Center to drop the sample database. Alternatively, you can issue the **drop database** *sample* command from the command line processor.

Creating a Server Access Profile

> You should have at least one database on your server before creating a server access profile.

To make it easy to set up a client, you can create a profile of your server that contains detailed information that the client needs to make a connection. This information includes the names and locations of all databases and instances on your server system. Creating a profile makes it easy to set up a single client as well as hundreds of clients. Follow these steps to create a profile that you can use to configure your clients:

1. Click on **Start** and select **Programs**->**DB2 for Windows NT**->**Administration Tools**->**Control Center**.
2. Expand the **System** directory tree by clicking on the plus sign (**+**).
3. Select the system that you want to profile and click with the right mouse button.
4. Select the **Generate Access Profile** option from the pop-up menu.

   Follow the instructions in the online help to write the profile to a diskette or to a file on the network that is accessible to the client workstations.

Now that you have successfully installed DB2, go to Chapter 6, “Installing DB2 Clients” on page 57 to install remote clients to access DB2 databases. If you are migrating from a single-partition database system, refer to the *Release Notes* for information on how to complete the migration process.

Part 3. Installing Clients

Chapter 6. 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 10, “Configuring Client-to-Server Communications Using the CCA” on page 71.

Use the instructions in this section to install a DB2 client.

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 7, “Installing DB2 Clients on Windows 95 or Windows NT Workstations” on page 59.
- Chapter 8, “Installing DB2 Clients on OS/2 Workstations” on page 63.
- Chapter 9, “Installing DB2 Clients on Windows 3.x Workstations” on page 65.

To install and configure UNIX or Macintosh clients, refer to the *Installing and Configuring DB2 Clients* manual, available on the DB2 Client Application Enablers CD-ROM.

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 7. 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:

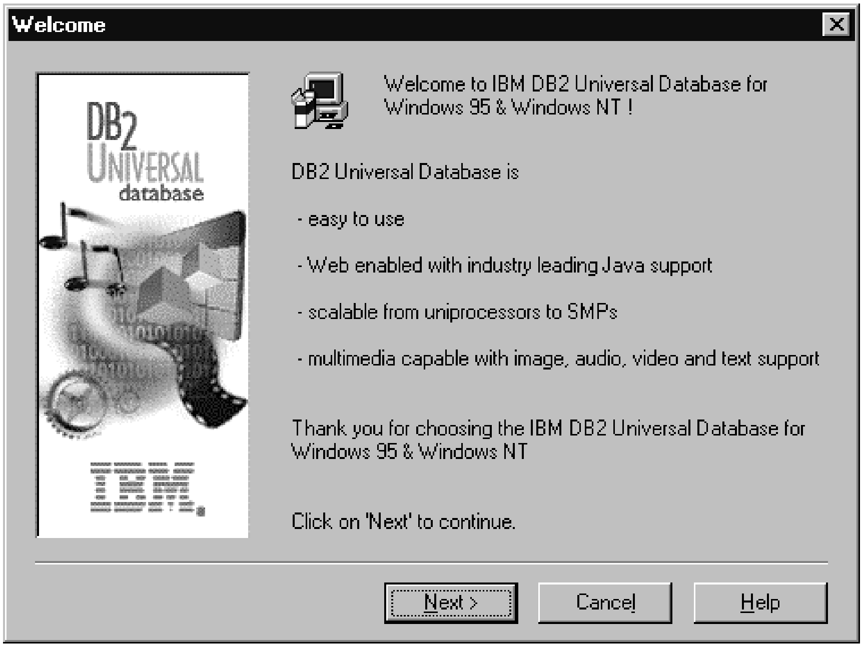
```
x:\setup /i language
```

where:

- *x*: represents your CD-ROM drive
- *language* represents the two-character country code for your language (for example, EN for English). Table 14 on page 238 lists the code for each available language.

3 Select **OK**.

4 The Welcome window opens.



Welcome

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

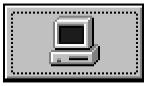
Thank you for choosing the IBM DB2 Universal Database for Windows 95 & Windows NT

Click on 'Next' to continue.

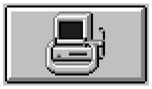
Click on the **Next** push button to continue.

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.

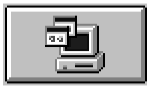
6 Select the installation type you prefer:



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.

Refer to the online *Installing and Configuring DB2 Clients* manual for information on setting up clients for communications 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 10, “Configuring Client-to-Server Communications Using the CCA” on page 71.

Chapter 8. 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.

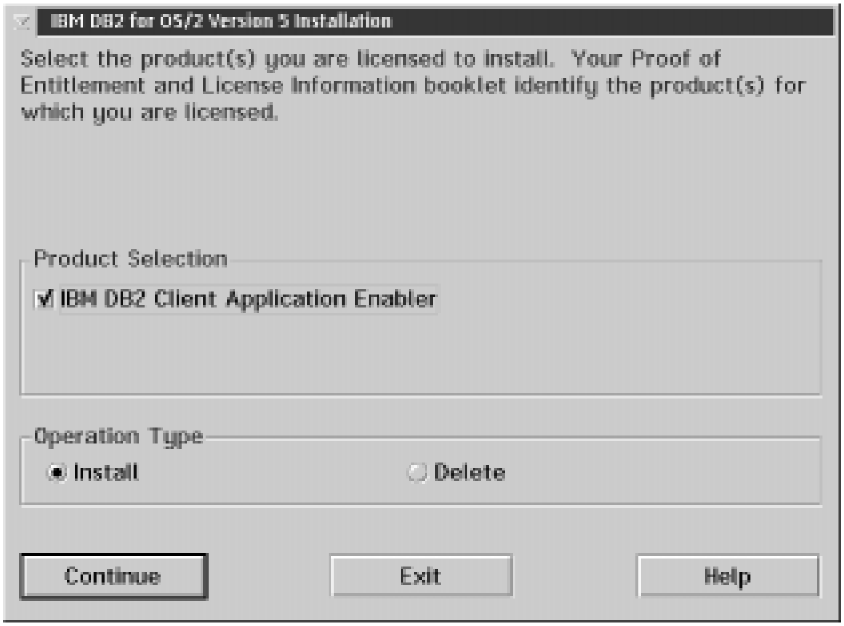
2 Begin the installation program as follows:

  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 14 on page 238 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 10, “Configuring Client-to-Server Communications Using the CCA” on page 71.

Chapter 9. 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:

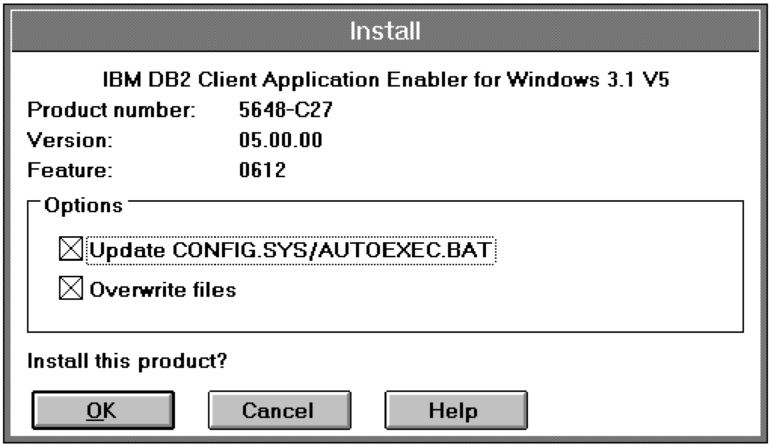
1 Insert the DB2 Client Application Enablers CD-ROM that contains the Windows 3.x client into the drive.

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 14 on page 238 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

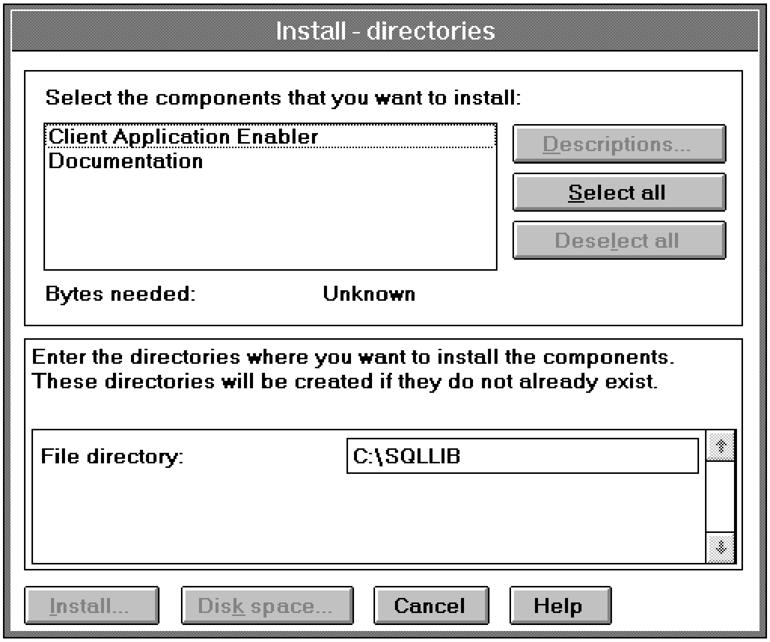
Install this product?

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 10, “Configuring Client-to-Server Communications Using the CCA” on page 71.

Part 4. Configuring DB2 Communications

Chapter 10. 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.

Before you can use the Client Configuration Assistant, be sure that you have completed the instructions in “Before You Begin” on page 45. You should also refer to “Setting Up the Administration Server to Use the CCA and the Control Center” on page 94.

To configure all other clients, refer to the online *Installing and Configuring DB2 Clients* manual.

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 80 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 12, “Getting Started with DB2 Universal Database” on page 79 for details.

Chapter 11. 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. Before you can use the Control Center, be sure that you have completed the instructions in “Before You Begin” on page 45. You should also refer to “Setting Up the Administration Server to Use the CCA and the Control Center” on page 94.

To configure an Administration Server for communications, go to Chapter 20, “Setting up Communications on the Server Using the Command Line Processor” on page 147.

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 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.

Part 5. Using DB2 Universal Database

Chapter 12. 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 80.
- "Connecting to a Database" on page 80.
- "Working with the System Administrative Group" on page 82.
- "Software Registration" on page 82.
- "Upgrading from a Try-and-Buy License" on page 82.

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.

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 82 for more information

See Appendix F, "Naming Rules" on page 239 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*.

To create the sample database, use one of the following methods:

- Click on the **Create the SAMPLE Database** graphic button in the First Steps main panel. (To start First Steps, click on **Start** and select **Programs**->**DB2 for Windows NT**->**First Steps**.)
- Enter the **db2sampl** command from the DB2 Command Window or in the operating system's command prompt. By default, the sample database will be created on the drive where DB2 was installed (you can optionally create the database on a different drive). For example, to create it on the F: drive, type:

```
db2sampl F:
```

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 in the Control Center or 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 Center or 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 Center or 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/NT 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.

Windows 95 Any Windows 95 user.

Windows NT Any valid DB2 username which belongs to the local Administrators group.

You can change the users who have administrator privileges for each DB2 instance by changing the *sysadm_group* parameter; but before you do, ensure that the group exists. Using the Windows NT User Manager Administrative Tool, check to see if this group exists.

Software Registration

The **Registration** icon, located in the **DB2** folder, is used to access and update the IBM Software Customer Profile and to register IBM Software with IBM. The customer profile contains identification and demographic information about the user of each IBM software product. The software registration tools contains information and instructions on how to register IBM software by modem, telephone, Internet, FAX, or postal service.

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. Put the product CD-ROM in the CD-ROM drive.
2. Double-click on the **Nodelock Administration Tool** icon in the **License Use Runtime - Client** folder to start the Nodelock Administration Tool.
3. Select **Products**->**New** from the menu bar.
4. Click on the **Import** push button.
5. In the Import window, locate the *db2\license* directory on your CD-ROM. A list of files is shown.
6. Select the **db2eent.lic** license file, this is the license file for DB2 Extended Enterprise Edition for Windows NT.
7. Click on **OK**.

Chapter 13. 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 network share that all machines in the instance can access.

Creating Instances

To create additional instances, use the **db2icrt** command. The machine that you run the **db2icrt** command on will become the instance-owning machine (node 0). We recommend that you always create instances on a machine where an Administration Server resides.

> You can choose to update an existing single-partition instance to the multipartition format using the **db2iupdt** command. For more information, refer to the *Release Notes*.

To create an instance, perform the following steps:

1 Log on to an account that belongs to the local Administrators group.

2 From a command prompt, run the **db2icrt** command as follows:

```
►►──db2icrt──── instance_name ──── /mpp ──┬─────────────────────┬────────────►
                                          └─/p:instprof_path ───┘

►── /u:username, password ──────┬──────────────────────────────┬────────────►◄
                                └─/r:base_port, end_port ──────┘
```

where:

instance_name Is the name of the instance.

/mpp Specifies that this instance is a partitioned database system.

/p: *instprof_path*

To specify a different instance profile path, you must create a shared drive or directory. For example: */p:\\machineA\DB2-DB2MPP*

This parameter is optional. If you do not specify the instance profile path, the instance directory is created under the *\sqllib* directory, and given the shared name DB2-*INSTANCE_NAME*. Read and write permissions will automatically be granted to everyone in the domain. You can change the permissions to restrict access to that directory.

/u: *username*, *password* Is the logon account name and password of the DB2 Service. For example: `/u:domainA\db2nt,password`

/r: *base_port***,** *end_port* Specifies the TCP/IP port range for the FCM; for example: /r:9010,9015

3 Optionally, create an Administration Server. For more information, see Chapter 14, “Using the Administration Server” on page 93.

> The **db2icrt** command grants the username used to create the instance the following Windows NT user rights:
>
> - Act as a part of the operating system
> - Create a token object
> - Increase quota
> - Logon as a service
> - Replace a process level token.
>
> The instance requires these user rights to access the shared drive, authenticate the user account, and run DB2 as a Windows NT service.

Adding a Database Partition Server to an Instance

Use the **db2ncrt** command to add a database partition server (node) to an instance.

> Do not use the **db2ncrt** command if there are databases in this instance.
>
> Instead, you should use the **db2start addnode** command. This ensures that the database is correctly added to the new database partition server.
>
> **Do not edit** the *db2nodes.cfg* file, since changing the file may cause inconsistencies in the partitioned database system.
>
> Refer to the *Administration Guide* for further information.

The syntax of the command is as follows:

```
►►──db2ncrt ── /n:node_number── /u:username, password ──────────►

►──┬───────────────────┬──┬──────────────────┬──┬────────────────┬──►
   └─/i:instance_name ─┘  └─/m:machine_name ─┘  └─/p:logical_port ─┘

►──┬───────────────┬──┬───────────────────────────────┬──────────►◄
   └─/h:host_name ─┘  └─/o:instance_owning_machine ─┘
```

where:

/n:*node_number* Is the unique node number to identify the database partition server. The number can be from 1 to 999 in ascending sequence.

/u:*username***,** *password* Is the logon account name and password of the DB2 service.

/i :*instance_name* Is the instance name; the default is the current instance.

/m:*machine_name* Is the computer name of the Windows NT workstation on which the node resides; the default name is the computer name of the local machine.

/p:*logical_port* Specifies the logical port number used for the database partition server, if the logical port number is not 0.

/h:*host_name* Is the TCP/IP host name that is used by FCM for internal communications if the host name is not the local host name.

/o:*instance_owning_machine* Is the computer name of the machine that is the instance-owning machine; the default is the local machine. This parameter is required when the **db2ncrt** command is invoked on any machine that is not the instance-owning machine.

Dropping a Database Partition Server from an Instance

Use the **db2ndrop** command to drop a database partition server from an instance that has no databases. If you drop a database partition server, its node number can be reused for a new database partition server.

Exercise caution when you are dropping database partition servers from an instance. If you drop the instance-owning database partition server (node *0*) from the instance, the instance will become unusable. If you want to drop the instance, use the **db2idrop** command. See “Removing Instances” on page 92 for more information.

Do not use the **db2ndrop** command if there are databases in this instance.

Instead, you should use the **db2stop drop** command. This ensures that the database is correctly removed from the database partition server.

Do not edit the *db2nodes.cfg* file, since changing the file may cause inconsistencies in the partitioned database system.

Refer to the *Administration Guide* for further information.

The syntax of the command is as follows:

```
►►──db2ndrop──/n:node_number──┬──────────────────┬──────────────────►◄
                              └─/i:instance_name─┘
```

where:

/n:*node_number* Is the unique node number to identify the database partition server. The number can be from 0 to 999 in ascending sequence. Not that Node 0 represents the instance-owning machine.

/i:*instance_name* Is the instance name; the default is the current instance (set by the *DB2INSTANCE* environment variable.

Listing Database Partition Servers in an Instance

Use the **db2nlist** command to obtain a list of database partition servers that participate in an instance.

The syntax of the command is as follows:

```
►►──db2nlist──┬──────────────────┬──┬────┬──────────────────────────►◄
              └─/i:instance_name─┘  └─/s─┘
```

where:

/i:*instance_name* Is the instance name; the default is the current instance (set by the *DB2INSTANCE* environment variable.

/s Displays the status of each node. The status of each node can be one of the following:

- Starting
- Running
- Stopping
- Stopped.

Listing Instances

To get a list of all the instances that are available on a system, enter:

```
db2ilist
```

To determine which instance applies to the current session, enter:

```
set db2instance
```

Setting the Current Instance

When you run commands to start or stop an instance's database manager, DB2 applies the command to the current instance. DB2 determines the current instance as follows:

- If the *DB2INSTANCE* environment variable is set for the current session, its value is the current instance. To set the *DB2INSTANCE* environment variable, enter:

  ```
  set db2instance=new_instance_name
  ```

- If the *DB2INSTANCE* environment variable is not set for the current session, DB2 uses the setting for the *DB2INSTANCE* environment variable from the system environment variables:

  – On Windows NT, system environment variables are set in the System Environment. See "Setting Your Environment on Windows 95 and Windows NT" on page 118 for instructions on setting *DB2INSTANCE*.

- If the *DB2INSTANCE* environment variable is not set at all, DB2 uses the registry value, *db2instdef*.

  To set the *db2instdef* registry value at the global level of the registry, enter:

  ```
  db2set db2instdef=new_instance_name -g
  ```

  For more information on setting registry values, refer to Chapter 17, "Controlling Your DB2 Environment" on page 115.

To determine which instance applies in the current session, enter:

```
set 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 with a username that has SYSADM authority on the instance.
2. Use one of these methods to start the instance:
   - From the Control Center, click with the right mouse button on the instance that you want to start and select the **Start** option from the pop-up menu.
   - From a command line, enter the **db2start** command.

   Note: The **db2start** command will start the instance as defined by the *DB2INSTANCE* environment variable. For more information, see “Setting the Current Instance” on page 90.

To stop an instance use one of these methods:

- From the Control Center, click with the right mouse button on the instance that you want to stop and select the **Stop** option from the pop-up menu.
- From a command line, enter the **db2stop** command.

  Note: If command line processor sessions are attached to an instance, you must run the **terminate** command to end each session before running the **db2stop** command. The **db2stop** command will stop the instance defined by the *DB2INSTANCE* environment variable. For more information, see “Setting the Current Instance” on page 90.

Running Multiple Instances Concurrently

To run multiple instances concurrently, use one of the following methods:

- From the Control Center, click with the right mouse button on another instance that you want to start and select the **Start** option.
- From the command line do the following:

  1 Set the *DB2INSTANCE* variable to the name of the other instance that you want to start, enter:

  ```
  set db2instance=another_InstName
  ```

  2 Start the instance by entering the **db2start** command.

See Chapter 17, "Controlling Your DB2 Environment" on page 115 for information on configuring each instance with different environment variables.

Removing Instances

To remove an instance, perform the following steps:

1 End all applications that are currently using the instance.

2 Stop the Command Line Processor by running **db2 terminate** commands in each DB2 command window.

3 Stop the instance by running the **db2stop** command.

4 Back up the instance directory indicated by the *DB2INSTPROF* registry value. For example, you might want to save the database manager configuration file, *db2systm*.

5 Issue the **db2idrop** command:

```
db2idrop InstName
```

where *InstName* is the name of the instance being dropped.

This command removes the instance entry from the list of instances and removes the instance directory.

Chapter 14. 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 an Administration Server

The setup program creates an Administration Server on the instance-owning machine during install. If, however, the setup program failed to create it, you can manually create an Administration Server using the **db2admin** command.

If you are using the CCA or the Control Center to automate connection configuration to a DB2 server, the database partition server that is on the same machine as the Administration Server will be the coordinator node. This means that all physical connections from the client to the database will be directed to the database partition server on the instance-owning machine before being routed to other database partition servers.

Creating additional Administration Servers on other machines allows the CCA or Control Center to configure other systems as coordinator nodes using DB2 Discovery (further information is provided below; also see Chapter 19, “Administering Clients Using the CCA” on page 135). To do this, perform the following steps:

1 Log on to the machine using an account that has local Administrator authority.

2 Create a Windows NT account that has local Administrator authority that will be used by the Administration Server. Ensure that the user name of the account adheres to the DB2 naming convention, for example, db2admin. See Appendix F, "Naming Rules" on page 239 for further information. When creating the account for the Administration Server, note the following:

- Do not use the account for the Administration Server as a User Account.
- Set the password for the account to `Password Never Expires`.

3 Run the following command:

```
db2admin create /user:username / password:password
```

where *username* and *password* are the username and password for the Administration Server.

Setting Up the Administration Server to Use the CCA and the Control Center

You must configure DB2 Discovery, a feature that is used by the CCA and Control Center, to retrieve information about systems on your network. Specifically, you may need to update instance lists and the Administration Server configuration to ensure that DB2 Discovery retrieves the correct information.

Update Instance Lists

An Administration Server may not be aware of all the instances in a partitioned database system because initially when an instance is created, only the Administration Server on the instance-owning machine is aware of the instance.

If you created an instance on a machine that does not have an Administration Server, you can create an Administration Server on this machine to make the instance known.

Perform the following steps if you created more than one Administration Server, and you want each Administration Server to be aware of all the instances in your partitioned database system.

1 **For each Administration Server**

Run the **db2ilist** command on the Administration Server machine to display a list of instances known to this Administration Server.

If this instance list includes the instances that you want the Administration Server to recognize, go to "Update the Administration Server Configuration" on page 95 for detailed information on how to configure Discovery.

2 **For each instance that is missing from the instance list in the step above**

On the instance-owning machine, run the **db2nlist** command to see if there is an entry for the machine that has the Administration Server. If there isn't an entry for the Administration Server machine, you must run the **db2ncrt** command to add this machine to the instance. For more information on the **db2ncrt** command, see "Adding a Database Partition Server to an Instance" on page 87.

Note: The network shared drive for the instance must be available on the Administration Server machine.

Update the Administration Server Configuration

By default, the setup program sets the *DB2SYSTEM* registry value to the Windows NT computer name. The system names that are retrieved by Discovery are the systems on which an Administration Server resides. Discovery uses these systems as coordinator nodes 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* (which is the default) 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.

Starting the Administration Server

To manually start the Administration Server, use the **db2admin** command as follows:

1 Log on to the machine using an account that has local Administrator authority.

2 From a command window, type:

```
db2admin start
```

Stopping the Administration Server

To manually stop the Administration Server, use the **db2admin** command as follows:

1 Log on to the machine using an account that has local Administrator authority.

2 From a command window, type:

```
db2admin stop
```

Setting the Logon Account for the Administration Server

> You should not use the **Services** utility in the **Control Panel** to change the logon account for the Administrator Server, since some of the required access rights will not be set for the logon account. Always use the **db2admin** command to set or change the logon account for the Administration Server.

After creating the Administration Server, you can set or change the logon account using the **db2admin** command as follows:

1 Log on to the machine using an account that has local Administrator authority.

2 From a command window, type:

```
db2admin setid username password
```

where *username* and *password* are the user name and password of an account that has local Administrator authority.

Removing the Administration Server

To remove the Administration Server, use the **db2admin** command as follows:

1 Log on to the machine using an account that has local Administrator authority.

2 Stop the Administration Server by entering the following command:

```
db2admin stop
```

3 Backup all the files in the *\sqllib\DB2DAS00* directory (if needed), where *DB2DAS00* is the Administration Server directory.

4 Drop the Administration Server by entering the following command:

```
db2admin drop
```

Chapter 15. 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 (typically *sqllib*). 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: *x:\sqllib\bnd*, where x: is the drive where you installed DB2

  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 99 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.

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 103
- “OS/2 Client Access to DB2 using CLI/ODBC” on page 105

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 136 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.

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.

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 136.

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.

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.

Detailed Configuration Information

The section “Platform Specific Details for CLI/ODBC Access” on page 102 should provide you with all of the information you require. *Installing and Configuring DB2 Clients* includes additional information on setting up and using DB2 CLI and ODBC applications. This information is useful where DB2 tool support is not available, and for administrators who require more detailed information.

The following topics are covered in the *Installing and Configuring DB2 Clients* manual:

- How to Bind the DB2 CLI/ODBC Driver to the Database
- How to Set CLI/ODBC Configuration Keywords
- Configuring db2cli.ini

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)

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 16. 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**
OS/2	• Command Center • command line processor • command 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.

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 OS/2, 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*. 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. You can find this node number by running the **db2nlist** command. 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*.

> Use the **db2nlist** command to see the list of database partition servers that participate in an instance.

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 Set *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 Set 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 17. 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 116.
- To set DB2 environment variables, see:
  - "Setting Your Environment on OS/2" on page 117.
  - "Setting Your Environment on Windows 95 and Windows NT" on page 118.
  - "Setting Your Environment on Windows 3.x" on page 119.
- To set database manager configuration parameters on a client instance, see "Configuration Parameters" on page 127.

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.

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.

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 119 for more information.

DB2 configures its operating parameters by checking for variable values according to the following search order:

- The environment variable settings.
- 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 120 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*.

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
  ```

  Note: 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*.

- 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 has 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.

DB2 Extended Enterprise Edition servers on Windows NT have two system environment variables, *DB2INSTANCE* and *DB2NODE*, that can only be set outside the profile registry. 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.

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 113.

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 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 119 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.

Parameter	Operating System	Values	Description
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.
DB2DBMSADDR	Windows 95, and Windows NT	Default= (not set), value for Windows 95: (90000000), value for Windows 95: (20000000)	Sets the memory location to which the Windows 95 or Windows NT system allocates shared memory for an instance. On Windows 95, the memory location that is assigned to the default instance, DB2, is 90000000. On Windows NT, location of the default instance is typically at 20000000. You need to override the default value only if you get an error message indicating you could not create database shared memory. Additional instances must be assigned memory values in increments of at least 3000000 bytes.
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.
DB2DMNBCKCTLR	Windows NT	Default: null	If DB2 is installed on a backup domain controller, setting db2dmnbckctlr=YES allows DB2 to use the security database on the backup domain controller, thereby reducing LAN traffic. **Note:** A backup domain controller shadows the security database on the primary domain controller.

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.
DB2INSTOWNER	Windows NT		The environment variable that is created in the DB2 profile registry when the instance is first created. This variable is set to the name of the instance-owning machine.

Parameter	Operating System	Values	Description
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.
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.

Parameter	Operating System	Values	Description
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.
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.

Parameter	Operating System	Values	Description
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*.
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. By default it is set to *x:\sqllib\win* on Windows 95 and Windows NT.
DB2NODE	All		Used to specify the target logical node of a DB2 Extended Enterprise Edition database partition server that you want to connect to. If this variable is not set, the target logical node defaults to the logical node which is defined with port *0* on the machine.
		Communications	
DB2COMM	All, server only	Default= null, values: any combination of APPC,IPXSPX, NETBIOS, NPIPE,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.

Parameter	Operating System	Values	Description
DB2NBADAPTERS	OS/2, Windows 95, and Windows NT	Default=0, Range: 0-15, Multiple values should be separated by commas	Used on the server to specify which local adapters to use for DB2 NetBIOS LAN communications. Each local adapter is specified using its logical adapter number. You can set *db2nbadapters* on the client if you want to perform a NetBIOS discovery on an adapter other than the default adapter 0. **Note:** On Windows 95 *db2nbadapters* is only used for discovery.
DB2NETREQ	Windows 3.1	Default=3, Range: 0-25	Specifies the number of NetBIOS requests that can be run concurrently on Windows 3.1 clients. The default value is 3. The number can range from 0 to 25. The higher you set this value, the more memory below the 1MB level will be used. When the concurrent number of requests to use NetBIOS services reaches the number you have set, subsequent incoming requests for NetBIOS services are held in a queue and become active as the current requests complete. If you enter 0 (zero) for *db2netreq*, the Windows database client issues NetBIOS calls in synchronous mode using the NetBIOS wait option. In this mode, the database client allows only the current NetBIOS request to be active and does not process another one until the current request has completed. This can affect other application programs. The 0 value is provided for backwards compatibility only. It is strongly recommended that 0 not be used.
DB2PORTRANGE	Windows NT		Sets the TCP/IP port range used by the fast communication manager (FCM). This ensures that any additional database partition servers created on other machines will have the same port range.

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.

- For information on using Client Configuration Assistant, see Chapter 19, “Administering Clients Using the CCA” on page 135.

Chapter 18. Using the Windows NT Performance Monitor

There are two performance monitors available to DB2 for Windows NT users:

DB2 Performance Monitor
The DB2 Performance Monitor provides snapshot and event data related to DB2 only. (For more information, click on the **Help** push button in the Control Center and see the Getting Started online help.)

Windows NT Performance Monitor
The Windows NT Performance Monitor enables you to monitor both database and system performance, retrieving information from any of the performance data providers registered with the system. Windows NT also provides performance data on all aspects of machine operation, including:

- CPU usage
- Memory utilization
- Disk activity
- Network activity

To set up and use the Windows NT Performance Monitor with DB2, refer to:

- "Registering DB2 with the Windows NT Performance Monitor"
- "Enabling Remote Access to DB2 Performance Information" on page 130
- "Displaying DB2 Performance Values" on page 131
- "Accessing Remote DB2 Performance Information" on page 132
- "Resetting DB2 Performance Values" on page 132.

Registering DB2 with the Windows NT Performance Monitor

The setup program automatically registers DB2 with the Windows NT Performance Monitor for you.

To make DB2 performance information accessible to the Windows NT Performance Monitor, you must register the DLL for the DB2 for Windows NT Performance Counters. This also enables any other Windows NT application using the Win32 performance APIs to get performance data.

To install and register the DB2 for Windows NT Performance Counters DLL (DB2Perf.DLL) with the Windows NT Performance Monitor, type:

```
db2perfi /i
```

This copies the DLL to the directory *\SYSTEM32* under the system directory. To find the name of the system directory, type

```
echo %systemroot%
```

Registering the DLL also creates a new key in the services section of the registry. One entry gives the name of the DLL, which provides the counter support. Three other entries give names of functions provided within that DLL. These functions are:

Open Called when the DLL is first loaded by the system in a process.

Collect Called to request performance information from the DLL.

Close Called when the DLL is unloaded.

Enabling Remote Access to DB2 Performance Information

In order to see Windows NT performance objects from another DB2 for Windows NT machine, you must register an administrator username and password with DB2. (The default Windows NT Performance Monitor username, *SYSTEM*, is a DB2 reserved word and cannot be used.) To register the name, type:

```
db2perfr -r username password
```

Note: The username you choose must conform to the naming rules in Appendix F, “Naming Rules” on page 239.

The username and password data is held in a key in the registry, with security that allows access only by administrators and the SYSTEM account. The data is encoded, to prevent security concerns about storing an administrator password in the registry.

Notes:

1. Once a username and password combination has been registered with DB2, even local instances of the Performance Monitor will explicitly log on using that username and password. This means that if the username information registered with DB2 does not match, local sessions of the Performance Monitor will not show DB2 performance information.
2. The username and password combination must be maintained to match the username and password values stored in the Windows NT Security database. If the username or password is changed in the Windows NT Security database, the username and password combination used for remote performance monitoring must be reset.
3. To deregister, type:

   ```
   db2perfr -u
   ```

Displaying DB2 Performance Values

To display DB2 performance values using the Performance Monitor, simply choose the performance values you want displayed from the **Add to** box. This box displays a list of performance objects providing performance data. Select an object to see a list of the counters it supplies.

A performance object can also have multiple instances. For example, the LogicalDisk Object provides counters such as '% Disk Read Time' and 'Disk Bytes/sec'; it also has an instance for each logical drive in the machine, including 'C:' and 'D:'.

Windows NT provides the following performance objects:

Windows NT Database Manager
This object provides general server information for a single Windows NT server instance. The DB2 instance being monitored appears as the object instance.

For practical and performance reasons, you can only get performance information from one DB2 server instance at a time. The DB2 instance that the Performance Monitor shows is governed by the *db2instance* registry value in the Performance Monitor process. If you have multiple DB2 server instances running simultaneously and want to see performance information from more than one, you must start a separate session of the Performance Monitor, with *db2instance* set to the relevant value for each DB2 instance to be monitored.

Windows NT Databases
This object provides information for a particular database. Information is available for each currently active database.

Windows NT Applications
This object provides information for a particular DB2 application. Information is available for each currently active DB2 application.

Accessing Remote DB2 Performance Information

Enable remote access to DB2 (see “Enabling Remote Access to DB2 Performance Information” on page 130) if you want the Performance Monitor to remotely access performance information on another machine. In the **Add to** box, select another computer to monitor. This beings up a list of all the available performance counters on that computer.

Resetting DB2 Performance Values

When an application calls the DB2 monitor APIs, the information returned is normally the cumulative values since the DB2 server was started. However, often it is useful to:

- Reset performance values
- Run a test
- Reset the values again
- Re-run the test.

To reset database performance values, use the **db2perfc** program.

Type:

```
db2perfc
```

By default, this resets performance values for all active databases. However, you can also specify a list of databases to reset. The program resets the values for ALL programs currently accessing database performance information for the relevant DB2 server instance (that is, the one held in *db2instance* in the session in which you run **db2perfc**).

Invoking **db2perfc** also resets the values seen by anyone remotely accessing DB2 performance information when the **db2perfc** command is executed.

Note: There is a DB2 API, sqlmrset, that allows an application to reset the values it sees locally, not globally, for particular databases.

Part 6. Advanced Configuration

Chapter 19. Administering Clients Using the CCA

The information in this section describes how to use the Client Configuration Assistant to perform administration tasks on OS/2, Windows 95, and Windows NT workstations. Before you can use the Client Configuration Assistant, be sure that you have completed the instructions in "Before You Begin" on page 45. You should also refer to "Setting Up the Administration Server to Use the CCA and the Control Center" on page 94.

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.”
- Searching the network for databases, see “Searching the Network for Databases” on page 139.
- Adding database connections manually, see “Manually Configure a Connection to a DB2 Database” on page 144.

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 143.

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 any combination of TCP/IP and NetBIOS; the protocols 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 17, “Controlling Your DB2 Environment” on page 115.

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 [ NETBIOS,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 20. 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 11, “Configuring Server Communications Using the Control Center” on page 75 for more information.

For instructions on using the command line processor, see Chapter 16, “Entering DB2 Commands and SQL Statements” on page 109. For a description of database naming rules, see Appendix F, “Naming Rules” on page 239.

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, did not install the Control Center, 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

netbios starts NetBIOS support

npipe starts NAMED PIPE 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 116.

You are now ready to configure the server workstation to use any of the following communication protocols:

- Named Pipes - see “Configuring Named Pipes on the Server”
- TCP/IP - see “Configuring TCP/IP on the Server” on page 150
- NetBIOS - see “Configuring NetBIOS on the Server” on page 153
- IPX/SPX - see “Configuring IPX/SPX on the Server” on page 157
- APPC - see “Configuring APPC on the Server” on page 160

Configuring Named Pipes on the Server

To access a remote server through Named Pipes, 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 43 for the supported communication protocols for your particular client and server.

If you have read and completed the instructions in “Setting the DB2COMM Registry Value” on page 147, there are no further steps required to set up Named Pipe communications on the server or to support the Administration Server.

Now that you have configured the server, you are ready to install a DB2 client. Go to Chapter 6, “Installing DB2 Clients” on page 57 for more information.

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 43 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 147.

> 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 6. 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 *\winnt\system32\drivers\etc* directory.

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 79.

2 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 6, “Installing DB2 Clients” on page 57 for more information.

Configuring NetBIOS on the Server

To access a remote server through NetBIOS, 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 43 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 147.

The following steps are required to set up NetBIOS communications:

1 Identify and record parameter values.

2 Configure the server:

  a Configure the NetBIOS Interface.

  b 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 7. NetBIOS Values Required at the Server

Parameter	**Description**	**Sample Value**	**Your Value**
Adapter number (*adapter_number*)	The local logical adapters that will be used for the NetBIOS connection. The server uses adapter `0` if this parameter is not configured.	`0`	
Workstation name (*nname*)	The NetBIOS name of the server workstation. *nname* is a name, chosen by the user, where the last *4* characters must be unique among all NetBIOS server nodes in the network. For more information on workstation names, see "Workstation Names (nname)" on page 242	`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. Configure the NetBIOS Interface

In order for a client to connect to a Windows NT server, the network route of the adapter number used must be **Nbf** for native NetBIOS operation. If the *db2nbadapters* parameter is not set, the default local adapter will be *0*. If you want to use additional server adapters, or you need to specify a server adapter other than *0*, set the *db2nbadapters* parameter to the number that you want to use.

> DB2 uses registry values to control its use of the NetBIOS resources on the server. Use the *db2nbadapters* registry parameter when you want to specify a value other than the default Logical adapter number `0`.
>
> For the DB2 server, set the *db2nbadapters* parameter by issuing the **db2set db2nbadapters=***adapter_number* command.
>
> For the Administration Server, set the *db2nbadapters* parameter by issuing the **db2set db2nbadapters=***adapter_number -i DB2DAS00* command.
>
> For more information, see Chapter 17, "Controlling Your DB2 Environment" on page 115.

To view or modify the NetBIOS interface configuration, which maps network routes to adapter numbers, do the following:

1. Click on **Start** and select **Settings**->**Control Panel**.
2. Double-click on the **Network** icon and select the **Services** tab.
3. Select the **NetBIOS Interface** option from the Network Services window and click on the **Properties** push button.
4. Scroll through the network routes until you find the Logical adapter number associated with **Nbf** and record it on your worksheet.
5. To change the logical adapter number associated with **Nbf**, select the associated **Lana Number**, and click on the **Edit** push button. Enter the new adapter number `0` (or the value that you set for *db2nbadapters*).
6. Record the new adapter number associated with **Nbf** on your worksheet.
7. Click on **OK**.
8. Click on the **Close** push button.
9. Shutdown and reboot your system for these changes to take effect.

Each adapter number must be uniquely associated with a network route. Windows NT has a built in checking facility that will not allow you to specify the same adapter number for different network routes. If a Network route already exists that is using the adapter number *0*, assign a different number to that route. (The valid range for adapter numbers is *0* to *255*.) This will allow you to select *0* as the adapter number that corresponds to **Nbf**. Approve the changes by clicking on **OK**.

B. Update the Database Manager Configuration File

You must update the database manager configuration file with the server's Workstation name (*nname*) 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 79.

2. Update the database manager configuration file with the server's Workstation name (*nname*) parameter using the following commands in the command line processor:

```
update database manager configuration using nname nname
db2stop
db2start
```

For example, if the server's workstation name (*nname*) is `server1`, use:

```
update database manager configuration using nname server1
db2stop
db2start
```

If you are configuring the Administration Server for NetBIOS, you must update the admin server configuration file. Use the following command:

```
update admin configuration using nname server1
db2admin stop
db2admin start
```

Autostarting DB2 with NetBIOS

If your NetBIOS protocol was configured when you installed on the instance-owning machine, the setup program automatically created a dependency for the Administration Server. If you created the Administration Server on anything other than the instance-owning machine, you will need to manually create a dependency on NetBIOS for each new instance.

Note: You can only autostart the Administration Server; you cannot autostart the instance.

You can create this dependency with the **db2depnb** command as follows:

1 Go to the *x:\sqllib\misc* directory, where x: is the drive on which DB2 was installed

2 Enter the **db2depnb** command as follows:

```
db2depnb instance_name
```

This records a dependency on the startup order which causes NetBIOS to start before any DB2 instances start.

If you remove the NetBIOS protocol from your network, you **must** remove the dependencies that were created during installation, and any dependencies that you created for additional instances. Failure to remove these dependencies may cause problems when running DB2 after the NetBIOS protocol has been removed from the network.

To remove a dependency, enter the **db2depnb** command as follows:

```
db2depnb instance_name /r
```

Now that you have configured the server, you are ready to install a DB2 client. Go to Chapter 6, "Installing DB2 Clients" on page 57 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 43 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 147.

> 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 8. 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)	
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 79.

2 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*.

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 43 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 147.

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 9. 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 79.

2 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 21. 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 165.
- 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 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 10. 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.

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.

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 20, “Setting up Communications on the Server Using the Command Line Processor” on page 147. 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.

The *Administration Guide* summarizes the CCSIDs that a DRDA-AR must use in order to connect to the DB2 DRDA-AS.

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 instance that handles the TPN X'07'6DB is specified by the DB2 registry value DB2SERVICETPINSTANCE.

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. If you choose APPC as your communications protocol, you may be restricted in the types of database manager configuration authentication setting you can use at the DB2 UDB Server. If you have the SNA Syncpoint Manager configured, you can use any available authentication (SERVER, CLIENT, DCS). If you do not have the SNA Syncpoint Manager configured then you are limited to authentication CLIENT or DCS.

Connections via TCP/IP can use any of the available database manager authentication options (SERVER,CLIENT,DCS).

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 11 identifies which functions are implemented in the AS on the DB2 Universal Database server. The subsequent table lists the supported bind options.

Table 11. 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 12 (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 12 (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 12 (Page 3 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
	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 12 (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 12 (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 *DB2PATH\misc*, for example:

```
c:\SQLLIB\misc\
```

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 segments, that is allocated for use by the Application Server.

Refer to the *Administration Guide* for more information about database manager configuration.

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 183 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 183 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 183 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 185.

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.TXT* file. This file is located in the directory where the product is installed.

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 OS/2, Windows 95, and Windows NT systems:

1. Copy the compressed PostScript files to a hard drive on your system. The files have a file extension of *.exe* and are located in the *x:\doc\language\ps* directory, where x: is the letter representing the CD-ROM drive and where *language* is the two-character country code that represents your language (for example, *EN* for English). Table 14 on page 238 lists the code for each available language.

2. Decompress the file by running the book name. The result from this step is a printable PostScript file with a file extension of *.ps*.

3. Ensure that your default printer is a PostScript printer capable of printing Level 1 (or equivalent) files.

4. Enter the following command from a command line:

   ```
   print filename.ps
   ```

> 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, insert the Portfolio CD-ROM into the CD-ROM drive. The Netscape installation automatically begins. Follow the instructions provided on the panels to install the product. If the installation does not automatically begin, see the *README.TXT* file located at the root directory on 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. Installation Troubleshooting

Use this section to correct errors that occur during installation.

Error: Invalid TCP/IP port range
Solution: Ensure that the port range designated for the instance-owning machine is available on this machine. To correct this error, do one of the following:

- Free up the port range that you specified for the instance-owning machine on this machine by editing the *x:\winnt\system32\drivers\etc\services* file, where *x:* represents the drive where you installed Windows NT.
- Issue the **db2set** command to use a different port range for the partitioned database system as follows:

  ```
  db2set DB2PORTRANGE=begin_port,ending_port
  ```

  Note: You must issue the **db2set** command on all machines that make up your partitioned database system.

Error: Invalid TCP/IP port for the Performance Monitor
Solution: Ensure that the port number designated for the Performance Monitor is available on this machine. To correct this error, do one of the following:

- Free up the port that you specified during the installation of the instance-owning machine on this machine by editing the *x:\winnt\system32\drivers\etc\services* file, where *x:* represents the drive where you installed Windows NT.
- Issue the **db2set** command to globally set a different port for the Performance Monitor as follows:

  ```
  db2set -g DB2CCMSRV=port_number
  ```

  Note: You must issue the **db2set** command on all machines that make up your partitioned database system.

Error: Name conflict
Solution: An instance with the same name already exists on this machine. To correct this error, do one of the following:

- Add this machine to a different instance
- Drop the local instance that is causing the name conflict.

Error: Cannot validate username during configuration
Solution: Ensure that the specified username exists in the domain.

Error: Cannot validate password during configuration
Solution: Ensure that the username that you are using to run the setup program belongs to the local Administrators group and has the *"Act as part of the operating system"* advanced user right. If it doesn't, assign the required permissions, then log off and log back on as this user; alternatively, specify a different username that has the required permissions.

Appendix C. 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.

To run the **rah** command or the **db2_all** command, you must be logged on with a user account that is a member of the Administrators group.

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, the output is 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 an ampersand (&). 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 node configuration 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 the P*xxxx.nnn* file in the instance directory, where P*xxxx* is the process ID and *nnn* is the node number.

These commands execute **rah** to run in parallel at all machines.

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 $
```

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 (\\).

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 window provides some way of entering a single quotation mark inside a singly quoted string.

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 "||db2 get db cfg for sample"
```

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, background 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.

|| 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.

; Same as ||& above. This is an alternative shorter form.

' 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. Do not edit the node configuration file. To obtain the machine index, use the **db2nlist** command. See “Listing Database Partition Servers in an Instance” on page 89.
2. When " is specified, duplicates are not eliminated from the list of machines. See “Eliminating Duplicate Entries from the List of Machines” on page 195 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 node configuration file. You can override this by:

- Specifying a pathname to the file that contain the list of machines by setting the environment variable RAHOSTFILE.
- Specifying the list explicitly, as a string of names separated by spaces, by setting the environment variable RAHOSTLIST. If both of these environment variables are set, then RAHOSTLIST takes precedence.

Note: To avoid introducing inconsistencies into the node configuration file, do *not* edit it manually. To obtain the list of machines in the instance, use the **db2nlist** command. See "Listing Database Partition Servers in an Instance" on page 89.

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.

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 193 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 13.

Name	Meaning	Default
RAHOSTFILE	file containing list of hosts	*db2nodes.cfg*
RAHOSTLIST	list of hosts as a string	extracted from $RAHOSTFILE
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. 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
RAHUSER	the logon account associated with the DB2 Remote Command Service	$USER

Setting the Default Environment Profile

To set the default environment profile for the **rah** command, use a file called *db2rah.env*, which should be created in the instance directory. The file should have the following format:

```
; This is a comment line
DB2INSTANCE=instancename
DB2DBDFT=database
; End of file
```

You can specify all the environment variables that you need to initialize the environment for **rah**.

Appendix D. Implementing Failover Support in the Windows NT Environment

You can set up your partitioned database system so that if a machine fails, the database partition server on the failed machine can run on another machine. On Windows NT, you implement failover support with Microsoft Cluster Server (MSCS). To use MSCS, you require Windows NT Version 4.0 Enterprise Edition with the MSCS feature installed.

MSCS can perform both failure detection and the restarting of resources in a clustered environment, such as failover support for physical disks and IP addresses. (When the failed machine is online again, resources will not automatically fall back to it, unless you previously configure them to do so. For more information, see "Failback Considerations" on page 216.)

Before you enable DB2 instances for failover support, ensure that you are familiar with the Cluster Administrator tool and how to work with MSCS resources. Then perform the following planning steps:

1 Decide on the number of database partition servers that you want in the instance.

2 Decide which disks you want to use for data storage. Each database partition server should be assigned at least one disk for its own use. The disk that you use to store data must be attached to a shared disk subsystem, and must be configured as an MSCS disk resource.

3 Decide which database partition servers you want to enable to fail over. You should enable the database partition server that runs on the instance-owning machine for failover. If you use a database partition server as a dedicated catalog node, this would also be a good candidate for failover support.

4 Ensure that you have one IP address for each database partition server that you want to use as a coordinator node.

To enable failover support, perform the following steps:

1 Set up DB2 groups and resources using the Cluster Administrator tool. See "Registering a Resource Type of DB2" on page 201 for information about registering the resource type, and "Setting up a Hot Standby Configuration" on page 202 and "Setting up a Mutual Takeover Configuration" on page 206 for details on how to set up the DB2 groups and resources.

2 Create a new DB2 instance and enable it for MSCS support. See "Creating the DB2 Instance" on page 212. You can also migrate an existing instance to enable it for MSCS support. See "Migrating an Instance to Use MSCS Support" on page 212.

3 Enable the instance for failover support on the second MSCS node in the cluster. See “Enabling the DB2 Resource to Fail Over to the Second MSCS Node” on page 214.

4 Add additional database partition servers to the instance. See “Adding an Additional Database Partition Server to the Instance” on page 214.

5 Register database drive mapping so that databases can be created on the clustered disk. See “Registering Database Drive Mapping for Mutual Takeover Configurations” on page 217.

After you finish enabling the instance for failover support, your configuration will resemble Figure 12.

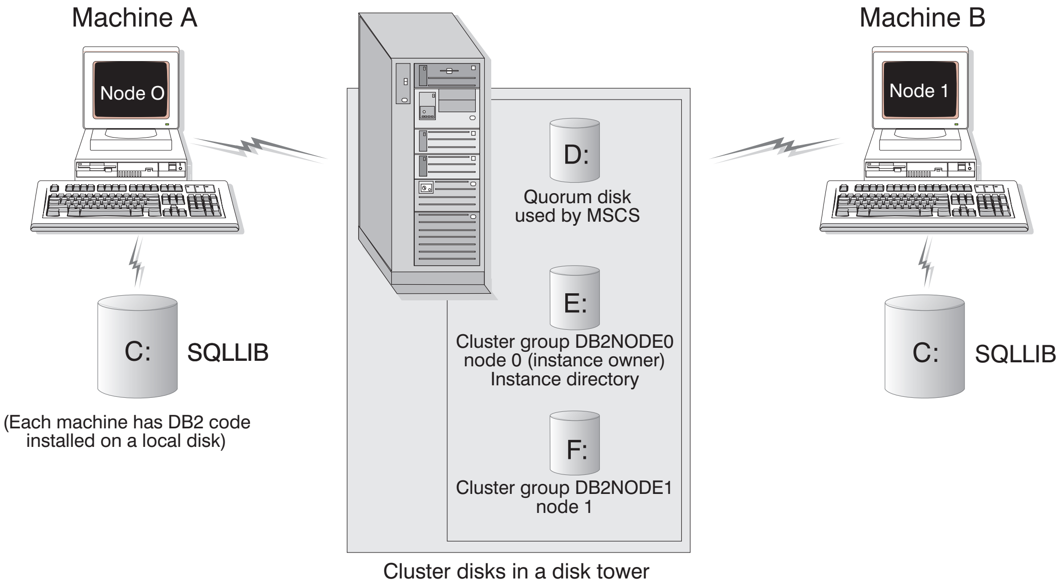

Machine A
Node O
C:
SQLLIB
(Each machine has DB2 code installed on a local disk)
D:
Quorum disk used by MSCS
E:
Cluster group DB2NODE0 node 0 (instance owner) Instance directory
F:
Cluster group DB2NODE1 node 1
Cluster disks in a disk tower
Machine B
Node 1
C:
SQLLIB

Figure 12. Example MSCS Configuration

The following sections describe the different types of failover support, and how to implement them. Before performing any of the steps described below, you must already have the MSCS software installed on every machine that you want to use in an MSCS cluster. In addition, you must also have DB2 installed on every machine.

Failover Configurations

Two types of configuration are available:

- Hot standby
- Mutual takeover

Currently, MSCS supports clusters of two machines. When you set up the MSCS system, the clusters do not all have to have the same type of configuration. You can have some clusters that are set up to use hot standby, and others that are set up for mutual takeover. For example, if your DB2 instance consists of five workstations, you can have two machines set up to use mutual takeover configuration, two to use hot standby configuration, with the last machine not configured for failover support.

Hot Standby Configuration

In a hot standby configuration, one machine in the MSCS cluster provides dedicated failover support, and the other 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 failover machine. Figure 13 shows an example of a hot standby configuration.

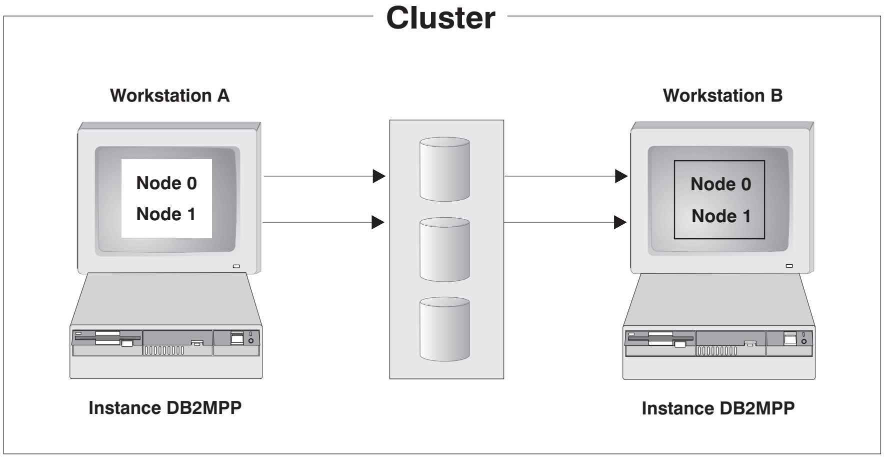


Figure 13. Hot Standby Configuration

To configure for a hot standby configuration, you:

1 Register the DB2 resource type. See “Registering a Resource Type of DB2” on page 201.

2 Create a group for DB2.

3 Set up a disk resource to store the user data.

4 Set up an MSCS File Share for the instance directory.

5 Create the IP resource for TCP/IP connections.

6 Create a network name.

7 Create a DB2 resource for each database partition server in the cluster.

For more information, see “Setting up a Hot Standby Configuration” on page 202.

Mutual Takeover Configuration

In a mutual takeover configuration, both workstations participate in the database system (that is, each machine has at least one database partition server running on it). If one of the workstations in the MSCS cluster fails, the database partition server on the failing machine will be started to run as a logical node 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.

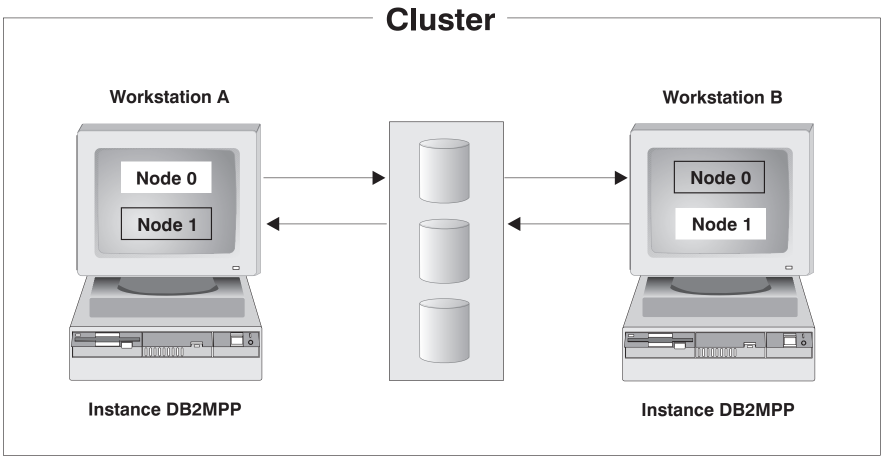


Figure 14. Mutual Takeover Configuration

To configure for a mutual takeover configuration, you:

1 Register the DB2 resource type. See “Registering a Resource Type of DB2” on page 201.

2 Create a group for each database partition server on the machine.

3. Set up a disk resource to store the user data.
4. Set up an MSCS File Share for the instance directory.
5. Create the IP resource for TCP/IP connections.
6. Create a network name.
7. Create a DB2 resource for each database partition server on the machine.

For more information, see “Setting up a Mutual Takeover Configuration” on page 206.

Setting up DB2 Groups and Resources from the Cluster Administrator

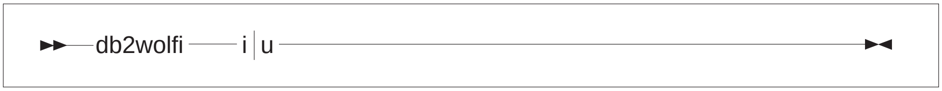
The following sections describe how to set up DB2 groups and resources from the Cluster Administrator tool.

Registering a Resource Type of DB2

Before you can set up your DB2 groups and resources, you must register a resource type of DB2 by issuing the **db2wolfi** command. Each MSCS resource that you register with a type of DB2 maps to a database partition server. Issue the following command from any command window *after* DB2 is installed. You only run this command once for each MSCS cluster:

```
db2wolfi i
```

The syntax of **db2wolfi** is as follows:



Where:

i Installs the DB2 resource type with MSCS.

u Uninstalls the DB2 resource type with MSCS.

Setting up a Hot Standby Configuration

If you want to set up a hot standby configuration for each MSCS cluster, you only require one DB2 group. To set up the DB2 group and resources for a hot standby configuration, perform the following steps for each cluster:

1. Start the Cluster Administrator tool, if it is not already active.
2. Create a named group called, for example, `DB2GROUP`. You can either:
   - Set up a new group.
   - Rename an existing group.
3. Set up a disk resource for the physical disks that will store the user data. You have two options:
   - If you already have a disk resource that was created for the physical disk, then move it to `DB2GROUP`.
   - If you do not have a disk resource for the physical disk, create one in `DB2GROUP`.
4. Set up an MSCS File Share for the instance directory. You perform this step on the machine that will contain node 0 (that is, the instance-owning machine). This step enables the instance directory to fail over, which ensures that the instance remains available:

   a. Select the `DB2GROUP` group and click the right mouse button.

   b. Select **New->Resource** from the pop-up menu.

   c. On the New Resource window:

      1. Type `Instance Directory` in the **Name** field.
      2. Type a description in the **Description** field. This step is optional.
      3. Select `File Share` from the list for the **Resource type** field.
      4. Leave the **Group** field set to `DB2GROUP`.
      5. Click on **Next**.

   d. On the Possible Owners window, click on the **Next** push button. This action accepts the default settings, so that both machines in the cluster are possible owners of the instance directory.

   e. On the Dependencies window:

      1. Select the disk resource (or resources) associated with the instance directory, and click on the **Add** push button. This action adds the resource dependency to the File Share resource.

Note: The disk must be online before the File Share resource can be brought online.

2 Click on the **Next** push button.

f On the Parameters window:

1 Type a name in the **Share name** field in the form `DB2-`*`instance_name`*. For example, `DB2-DB2MPP`.

Note: The 'Share Name' parameter of the 'File Share' resource must be used when creating the partitioned database system. See "Creating the DB2 Instance" on page 212 for more information.

2 Type a path that exists on the cluster disk in the **Path** field, for example, `G:\profile`.

3 Type a comment in the **Comment** field. This step is optional.

4 Click on the **Finish** push button.

5 Create the IP resource for TCP/IP connections. Only perform this step if you want the database partition server on the machine to function as a coordinator node for remote applications. This step also helps ensure that the instance directory can fail over:

a Select the `DB2GROUP` group and click the right mouse button.

b Select **New->Resource** from the pop-up menu.

c On the New Resource window, create the IP resource for the TCP/IP connection in `DB2GROUP`:

1 Type a name in the **Name** field, for example, `IP Address for DB2`.

2 Type a description in the **Description** field. This step is optional.

3 Select `IP Address` from the list for the **Resource type** field.

Note: The 'IP Address' parameter for the 'IP resource' should be used when cataloging the TCP/IP node on the client machine:

```
db2 catalog tcpip node node_name
   remote cluster_IP_address
   server service_name
```

The syntax of this command has been split for formatting reasons.

4 Leave the **Group** field set to `DB2GROUP`.

5 Click on the **Next** push button.

d On the Possible Owners window, click on the **Next** push button. This action accepts the default settings for possible owners.

e On the Dependencies window, click on the **Next** push button. This action accepts the default resource dependencies.

f On the TCP/IP Address Parameters window:

1 Type a new IP address that DB2 will use in the **Address** field.

2 Type a new subnet mask that DB2 will use in the **Subnet mask** field.

3 Select the network that you want to use to support remote connections for the **Network to use** field.

4 Click on the **Finish** push button.

g To bring the IP resource online and verify the parameter settings:

1 Select **DB2GROUP** in the object tree, and click the right mouse button.

2 Select the **IP Address for DB2** resource, and click the right mouse button.

3 Select **Bring Online** from the pop-up menu.

Note: If an error occurs, select the `IP Address for DB2` resource and click the right mouse button. Select **Properties** from the pop-up menu. On the Parameters tab, verify the values in the **Address** and **Subnet mask** fields.

6 Create a network name. You perform this step on the machine that will contain node 0. This step, combined with the step to create the File Share allows DB2 to access the instance directory when the instance directory fails over:

a Select `DB2GROUP` from the object tree, and click the right mouse button.

b Select **New->Resource** from the pop-up menu.

c On the New Resource window, create a network name resource:

1) Type a name for the network name in the **Name** field. For example, `Network Name for DB2`.

2) Type `Network Name` in the **Description** field. This step is optional.

3) Select `Network Name` from the list for the **Resource type** field.

4) Leave the **Group** field set to `DB2GROUP`.

5) Click on the **Next** push button.

d On the Possible Owners window, click on the **Next** push button. This action accepts the default settings for possible owners.

e On the Dependencies window:

1 Select the `IP Address for DB2` to the **Resource dependencies** list by using the **Add** push button.

2 Click on the **Next** push button.

f On the Network Name Parameters window, type a name in the **Name** field to use for the network. For example, `DB2WOLF`.

g Bring the network name resource online:

1 Select **DB2GROUP** in the object tree, and click the right mouse button.

2 Select the **Network Name for DB2** resource, and click the right mouse button.

3 Select **Bring Online** from the pop-up menu.

7 Create a DB2 resource for each database partition server:

a Select `DB2GROUP` from the object tree, and click the right mouse button.

b Select **New->Resource** from the pop-up menu.

c On the New Resource window:

1) Type a two-part name in the form *instance_name-node_number* in the **Name** field. For example, `DB2MPP-0`.

   The two-part name is made of the instance name (`DB2MPP`), and the node number (`0`). Ensure that you include the hyphen (-). If the instance does not already exist, you must use the same name when you create it with the **db2icrt** command. In addition, the node number that you specify for the resource must be the same number that you specify for the **db2ncrt** command.

   Note: The node number for each database partition server must be unique.

2) Type `Node 0` in the **Description** field. This step is optional.

3) Select `DB2` from the list for the **Resource type** field.

4) Leave the **Group** field set to `DB2GROUP`.

5) Click on the **Next** push button.

d On the Possible Owners window, click on the **Next** push button. This action accepts the default settings for possible owners.

e On the Dependencies window:

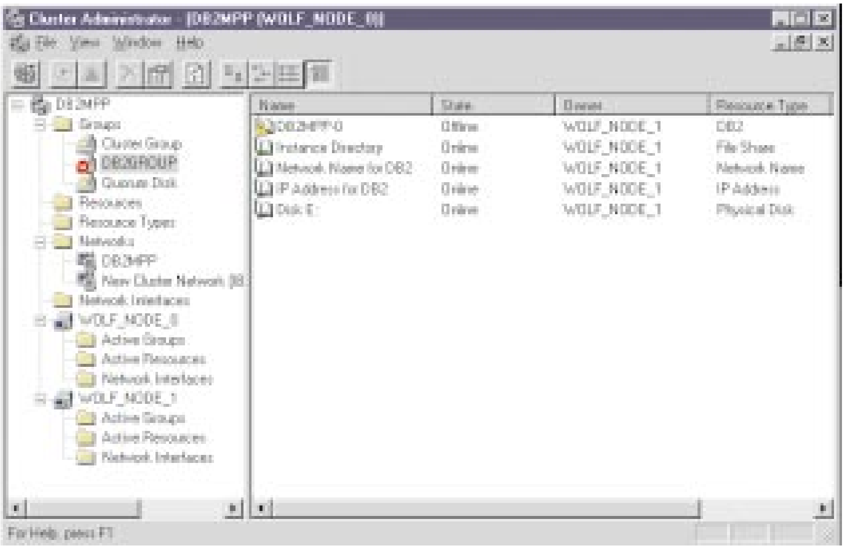
1 Select the available disk resources and IP resources to the **Resource dependencies** list by using the **Add** push button.

2 Click on the **Finish** push button.

Notes:

- **a** Do not bring the resource for the database partition server (DB2MPP–0) online yet, as DB2 is not fully set up yet.
- **b** When creating the DB2 resource, set its dependency on the Disk and IP resource so that it will be started after the other resources are started.

When you finish, the Cluster Administrator tool will resemble Figure 15.



Name	State	Owner	Resource Type
DB2MPP-0	Offline	WOLF_NODE_1	DB2
Instance Directory	Online	WOLF_NODE_1	File Share
Network Name for DB2	Online	WOLF_NODE_1	Network Name
IP Address for DB2	Online	WOLF_NODE_1	IP Address
Disk E:	Online	WOLF_NODE_1	Physical Disk

Figure 15. Cluster Administrator Tool for Hot Standby Configuration

Setting up a Mutual Takeover Configuration

To set up a mutual takeover configuration, you must create one DB2 group for every database partition server (node) that is on the machine. For example, for node 0, you would perform the following:

1 Start the Cluster Administrator tool, if it is not already active.

2 Create a named group called, for example, `DB2 Node 0`. You can either:

- Set up a new group.
- Rename an existing group.

Note: Be sure to assign a unique node number to the group that will be used by each database partition server.

3 Set up a disk resource for the physical disks that will store the user data. You have two options:

- If you already have a disk resource that was created for the physical disk, then move it to `DB2 Node 0`.
- If you do not have a disk resource for the physical disk, create one in `DB2GROUP`.

4 Set up an MSCS File Share for the instance directory. You perform this step on the machine that will contain node 0 (that is, the instance-owning machine). This step enables the instance directory to fail over, which ensures that the instance remains available:

a Select the `DB2 Node 0` group and click the right mouse button.

b Select **New->Resource** from the pop-up menu.

c On the New Resource window:

1) Type `Instance Directory` in the **Name** field.

2) Type a description in the **Description** field. This step is optional.

3) Select `File Share` from the list for the **Resource type** field.

4) Leave the **Group** field set to `DB2 Node 0`.

5) Click on the **Next** push button.

d On the Possible Owners window, click on the **Next** push button. This action accepts the default settings, so that both machines in the cluster are possible owners of the instance directory.

e On the Dependencies window:

1 Select the disk resource (or resources) associated with the instance directory, and click on the **Add** push button. This action adds the resource dependency to the File Share resource.

Note: The disk must be online before the File Share resource can be brought online.

2 Click on the **Next** push button.

f On the Parameters window:

1 Type a name in the **Share name** field in the form `DB2-`*`instance_name`*. For example, `DB2-DB2MPP`.

Note: The 'Share Name' parameter of the 'File Share' resource must be used when creating the partitioned database system. See “Creating the DB2 Instance” on page 212 for more information.

2 Type a path that exists on the cluster disk in the **Path** field, for example, `G:\profile`.

3 Type a comment in the **Comment** field. This step is optional.

4 Click on the **Finish** push button.

5 Create the IP resource for TCP/IP connections. Only perform this step if you want the database partition server on the machine to function as a coordinator node. This step also helps ensure that the instance directory can fail over:

a Select the `DB2 Node 0` group and click the right mouse button.

b Select **New->Resource** from the pop-up menu.

c On the New Resource window, create the IP resource for the TCP/IP connection in `DB2 Node 0`:

1) Type a name in the **Name** field, for example, `IP Address for DB2`.

2) Type a description in the **Description** field. This step is optional.

3) Select `IP Address` from the list for the **Resource type** field.

Note: The 'IP Address' parameter for the 'IP resource' should be used when cataloging the TCP/IP node on the client

```
db2 catalog tcpip node node_name
   remote cluster_IP_address
   server service_name
```

The syntax of this command has been split for formatting reasons.

4) Leave the **Group** field set to `DB2 Node 0`.

5) Click on the **Next** push button.

d On the Possible Owners window, click on the **Next** push button. This action accepts the default settings for possible owners.

e On the Dependencies window, click on the **Next** push button. This action accepts the default resource dependencies.

f On the TCP/IP Address Parameters window:

1 Type a new IP address that DB2 will use in the **Address** field.

2 Type a new subnet mask that DB2 will use in the **Subnet mask** field.

3 Select the network that you want to use to support remote connections for the **Network to use** field.

4 Click on the **Finish** push button.

g To bring the IP resource online and verify the parameter settings:

1 Select **DB2 Node 0** in the object tree, and click the right mouse button.

2 Select the **IP Address for DB2** resource, and click the right mouse button.

3 Select **Bring Online** from the pop-up menu.

Note: If an error occurs, select the `IP Address for DB2` resource and click the right mouse button. Select **Properties** from the pop-up menu. On the Parameters tab, verify the values in the **Address** and **Subnet mask** fields.

6 Create a network name. You perform this step on the machine that will contain node 0. This step, combined with the step to create the File Share allows DB2 to access the instance directory when the instance directory fails over:

a Select `DB2GROUP` from the object tree, and click the right mouse button.

b Select **New->Resource** from the pop-up menu.

c On the New Resource window, create a network name resource:

1) Type a name for the network name in the **Name** field. For example, `Network Name for DB2`.

2) Type `Network Name` in the **Description** field. This step is optional.

3) Select `Network Name` from the list for the **Resource type** field.

4) Leave the **Group** field set to `DB2GROUP`.

5) Click on the **Next** push button.

d On the Possible Owners window, click on the **Next** push button. This action accepts the default settings for possible owners.

e On the Dependencies window:

1 Select the **IP Address for DB2** to the **Resource dependencies** list by using the **Add** push button.

2 Click on the **Next** push button.

f On the Network Name Parameters window, type a name in the **Name** field to use for the network. For example, `DB2WOLF`.

g Bring the network name resource online:

1 Select **DB2GROUP** in the object tree, and click the right mouse button.

2 Select the **Network Name for DB2** resource, and click the right mouse button.

3 Select **Bring Online** from the pop-up menu.

7 Create a DB2 resource for the database partition server:

a Select `DB2 Node 0` from the object tree, and click the right mouse button.

b Select **New->Resource** from the pop-up menu.

c On the New Resource window:

1) Type a two-part name in the form *instance_name-node_number* in the **Name** field. For example, `DB2MPP-0`.

   The two-part name is made of the instance name (`DB2MPP`), and the node number (`0`). Ensure that you include the hyphen (-). If the instance does not already exist, you must use the same name when you create it with the **db2icrt** command. In addition, the node number that you specify for the resource must be the same number that you specify for the **db2ncrt** command.

   Note: The node number for each database partition server must be unique.

2) Type `Node 0` in the **Description** field. This step is optional.

3) Select `DB2` from the list for the **Resource type** field.

4) Leave the **Group** field set to `DB2 Node 0`.

   Note: The value displayed in the **Group** field will change according to the node number.

5) Click on the **Next** push button.

d On the Possible Owners window, click on **Next**. This action accepts the default settings for possible owners.

e On the Dependencies window:

1 Select the available disk resources and IP resources to the **Resource dependencies** list by using the **Add** push button.

2 Click on the **Finish** push button.

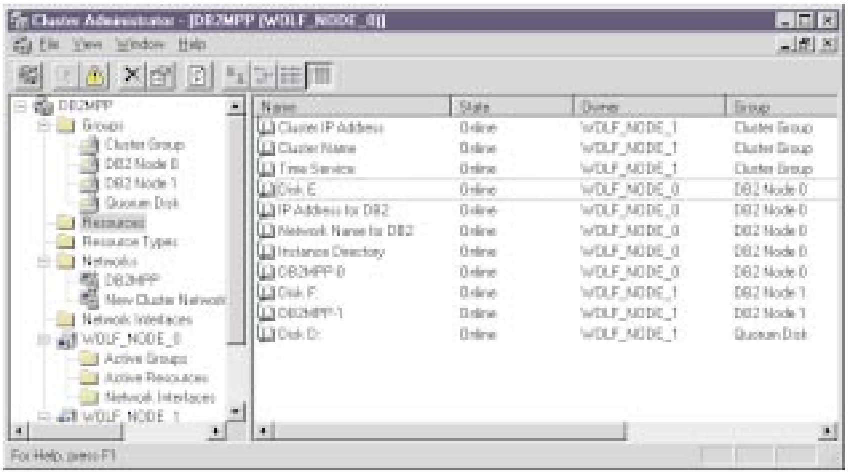
Notes:

a Do not bring the resource for the database partition server (DB2MPP–0) online yet, as DB2 is not fully set up yet.

b When creating the DB2 resource, set its dependency on the Disk and IP resource so that it will be started after the other resources are started.

For each database partition server on each machine, repeat the procedures to create a named group, set up a disk resource for user data, create the IP resource, and create a DB2 resource. (You do not have to repeat the steps to create an MSCS File Share for the instance directory, nor do you have to create another network name.) Remember to specify a different node number for each disk resource that you create, and to use the same node number when you issue the **db2ncrt** command to add a database partition server to the instance.

When you finish, the Cluster Administrator tool will resemble Figure 16 on page 211.



Name	State	Owner	Group
Cluster IP Address	Online	WOLF_NODE_1	Cluster Group
Cluster Name	Online	WOLF_NODE_1	Cluster Group
Time Service	Online	WOLF_NODE_1	Cluster Group
Disk E:	Online	WOLF_NODE_0	DB2 Node 0
IP Address for DB2	Online	WOLF_NODE_0	DB2 Node 0
Network Name for DB2	Online	WOLF_NODE_0	DB2 Node 0
Instance Directory	Online	WOLF_NODE_0	DB2 Node 0
DB2MPP-0	Online	WOLF_NODE_0	DB2 Node 0
Disk F:	Online	WOLF_NODE_1	DB2 Node 1
DB2MPP-1	Online	WOLF_NODE_1	DB2 Node 1
Disk D:	Online	WOLF_NODE_1	Quorum Disk

Figure 16. Cluster Administrator Tool for Mutual Takeover Configuration

Setting up the DB2 Instance for Failover Support

To set up the DB2 instance to use failover support, perform the following steps:

1. Use the **db2icrt** command to create the instance on the current MSCS node and, at the same time, enable it for MSCS support. See “Creating the DB2 Instance” on page 212.

   If you have an existing instance, you can use the **db2iclus migrate** command to migrate the instance to use MSCS. See “Migrating an Instance to Use MSCS Support” on page 212.

2. Use the **db2iclus add** command to enable the instance to fail over to the second MSCS node in the cluster. See “Enabling the DB2 Resource to Fail Over to the Second MSCS Node” on page 214.

3. Use the **db2ncrt** command to add an additional database partition server to the instance. (This step is optional.) See “Adding an Additional Database Partition Server to the Instance” on page 214.

Creating the DB2 Instance

To create the instance, use the **db2icrt** command. Be sure that you run the command from the machine which you previously assigned as node 0. The command syntax is as follows:

```
►►──db2icrt──── instance_name ── /mpp ── /p:instprof_path ──────────►

►── /u:account_name, password ─┬───────────────────────────┬─ /c ──────►◄
                               └─ /r:base_port, end_port ──┘
```

Where:

instance_name
Is the name of the instance.

/mpp
Specifies that this is a partitioned database system.

/p: *instprof_path*
Is the instance profile path. The path must use the universal naming convention, and include both the network name and the share name that were created as MSCS resources for node 0. The form is as follows: */p:\\network_name\share_name*. For example: */p:\\DB2WOLF\DB2-DB2MPP*.

/u: *account_name***,** *password*
Is the domain account used as the logon account name of the DB2 Service. For example:

```
/u:domainA\db2nt,password
```

/r: *base_port***,** *end_port*
Specifies the TCP/IP port range for the FCM; for example: /r:9010,9015

/c
Specifies that DB2 will use MSCS support.

The instance profile path must reside on an MSCS cluster disk, so that when the machine fails and its resources fail over to the other machine, the MSCS cluster disk also fails over and the instance can still access the instance directory. To achieve this, specify the MSCS network name (instead of the machine name) and the MSCS File Share (instead of a normal network share) for the **/p:** parameter. See “Setting up DB2 Groups and Resources from the Cluster Administrator” on page 201 for more information.

Migrating an Instance to Use MSCS Support

Use the **db2iclus migrate** command to migrate a partitioned database instance for MSCS support (The instance must have been created with the **/mpp** parameter of the **db2icrt** command.) You must run this command on the machine that runs node 0 (that is, the instance-owning machine). The command syntax is as follows:

```
►►──db2iclus  migrate──┬─────────────────┬── /p:instprof_path ──────►
                       └─/i:instance_name─┘

►──┬──────────────────┬──────────────────────────────────────────►◄
   └─/c:cluster_name──┘
```

Where:

/i: *instance_name* Is the name of the instance.

/p: *instprof_path* Is the instance profile path. The path must use the universal naming convention, and include both the network name and the share name that were created as MSCS resources for node 0. The form is as follows: */p:\\network_name\share_name*. For example: */p:\\DB2WOLF\DB2-DB2MPP*.

/c: *cluster_name* Is the name of the MSCS cluster as it is known on the LAN. This name is specified when the MSCS cluster is first created.

Note: If a database already exists in the instance and you want it to be available if the instance fails over, you must migrate it to an MSCS share disk. If you do not migrate the database to a share disk, it will not be available if the instance fails over. To migrate a database:

1 Take an offline backup copy of all database partitions for the database.

2 Uncatalog the database.

3 Migrate the instance to use MSCS support.

4 Set up database drive mapping. See “Registering Database Drive Mapping for Mutual Takeover Configurations” on page 217 for details.

5 Restore the database partitions to the share disk.

6 Catalog the database that you restored.

7 Verify that the instance is working correctly.

8 Using a different name, recatalog the original database.

9 Drop the original database.

For information about database recovery, refer to the *Administration Guide*.

Enabling the DB2 Resource to Fail Over to the Second MSCS Node

After the DB2 instance is created, configure the second MSCS node in the cluster to support the DB2 resource (database partition server) that fails over.

To enable the DB2 resource to fail over to the second node, use the **db2iclus add** command. The command syntax is as follows:

```
>>──db2iclus  add ──┬───────────────────┬──/u:account_name, password ──►
                    └─/i:instance_name──┘

►──┬──────────────────┬──┬─────────────────┬──────────────────────────►◄
   └─/m:machine_name──┘  └─/c:cluster_name─┘
```

Where:

/i: *instance_name* Is the name of the instance. (This parameter overrides the setting of the DB2INSTANCE environment variable.)

/u: *account_name*, *password* Is the domain account used as the logon account name of the DB2 Service. For example:

```
/u:domainA\db2nt,password
```

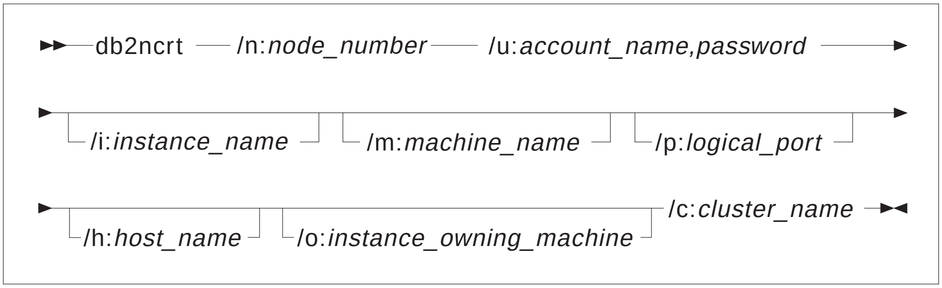
/m:*machine_name* Is the computer name of the machine where the database partition server is to fail over to. You must specify this option if you run the command from a machine other than the one that is to support the database partition server that fails over.

/c: *cluster_name* Is the name of the MSCS cluster as it is known on the LAN. This name is specified when the MSCS cluster is first created.

After you run the **db2iclus add** command, both machines in the MSCS cluster will be configured to run DB2 resource.

Adding an Additional Database Partition Server to the Instance

To add a database partition server to an existing instance, use the **db2ncrt** command. The command syntax is as follows:



Where:

/n:*node_number* Is the unique node number to identify the database partition server. The number can be from 0 to 999 in ascending sequence.

/u:*account_name***,** *password* Is the domain account used as the logon account name of the DB2 service.

/i :*instance_name* Is the instance name; the default is the current instance. (This parameter overrides the setting of the DB2INSTANCE environment variable.)

/m:*machine_name* Is the computer name of the machine on which the node resides; the default name is the computer name of the local machine.

/p:*logical_port* Specifies the logical port number used for the database partition server, if the logical port number is not 0.

/h:*host_name* Is the TCP/IP host name of the IP address that is used by FCM for internal communications, if the host name is not the local host name.

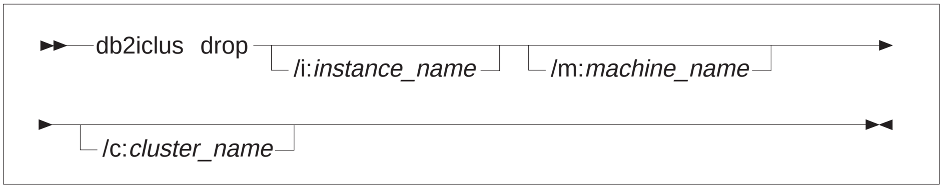
/o:*instance_owning_machine* Is the computer name of the machine that is the instance-owning machine; the default is the local machine. This parameter is required when the **db2ncrt** command is invoked on any machine that is not the instance-owning machine.

/c: *cluster_name* Is the name of the MSCS cluster as it is known on the LAN. This name is specified when the MSCS cluster is first created.

You can run the **db2ncrt** command from either the machine on which you installed DB2, or from the instance-owning machine.

Disabling DB2 Resource Failover to the Second MSCS Node

To disable a DB2 resource (database partition server) from failing over to the second MSCS node, use the **db2iclus drop** command. The command syntax is as follows:



```
►►──db2iclus  drop──┬────────────────────┬──┬───────────────────┬──────►
                    └─/i:instance_name───┘  └─/m:machine_name───┘

►──┬──────────────────┬─────────────────────────────────────────────►◄
   └─/c:cluster_name──┘
```

Where:

/i :*instance_name*
: Is the instance name; the default is the current instance. (This parameter overrides the setting of the DB2INSTANCE environment variable.)

/m:*machine_name*
: Is the computer name of the machine where the DB2 resource can currently fail over to. You must specify this option if you run the command from a machine other than the one that the DB2 resource can failover to.

/c: *cluster_name*
: Is the name of the MSCS cluster as it is known on the LAN. This name is specified when the MSCS cluster is first created.

Failback Considerations

By default, groups are set not to fail back to the original (failed) machine. Unless you manually configure a DB2 group to fail back after failing over, it continues to run on the alternative MSCS node after the cause of the failover has been resolved.

If you configure a DB2 group to automatically fail back to the original machine, all the resources in the DB2 group including the DB2 resource will fail back as soon as the original machine is available. If, during the fail back, a database connection exists, the DB2 resource cannot be brought offline, and the failback processing will fail.

If you want to force all database connections during the failback processing, set the DB2_FALLBACK registry variable to ON. This variable must be set as follows:

```
db2set DB2_FALLBACK=ON
```

You do not have to reboot or restart the cluster service after setting this registry variable.

Registering Database Drive Mapping for Mutual Takeover Configurations

When you create a database in the partitioned database environment, you can specify a drive letter for the **create database** command to indicate where the database is to be created. When the command runs, it expects that the drive that you specify will be simultaneously available to all the machines that participate in the instance. Because this is not possible, DB2 uses database drive mapping to assign the same drive a different name for each machine.

For example, assume that a DB2 instance called DB2 contains two database partition servers:

```
NODE0 is active on machine WOLF_NODE_0
NOSE1 is active on machine WOLF_NODE_1
```

Also assume that the share disk E: belongs to the same group as NODE0, and that the share disk F: belongs to the same group as NODE1.

To create a database on the share disk E:, the **create database** command would be as follows:

```
db2 create database mppdb on E:
```

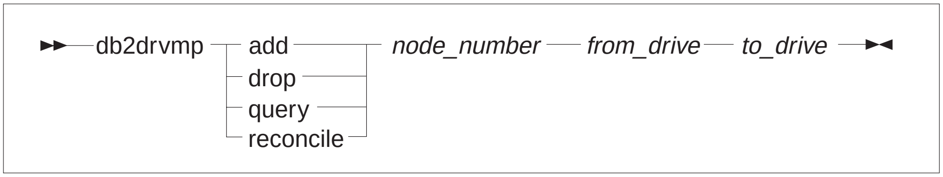
For the **create database** command to be successful, drive E: must be available to both machines. In a mutual takeover configuration, each database partition server may be active on a different machine, and the cluster disk E: is only available to one machine. In this situation, the **create database** command will always fail.

To resolve this problem, the database drive should be mapped as follows:

```
For NODE0, the mapping is from drive F: to drive E:
For NODE1, the mapping is from drive E: to drive F:
```

Any database access for NODE0 to drive F: is then mapped to drive E:, and any database access for NODE1 to drive E: is mapped to drive F:. Using drive mapping, the **create database** command will create database files on drive E: for NODE0 and drive F: for NODE1.

Use the **db2drvmp** command to set up the drive mapping. The command is as follows:



The parameters are as follows:

add Assigns a new database drive map.

drop Removes an existing database drive map.

query Queries a database map

reconcile Repairs a database map drive when the registry contents are damaged. See “Reconciling Database Drive Mapping” for more information.

node_number The node number. This parameter is required for add and drop operations.

from_drive The drive letter to map from. This parameter is required for add and drop operations.

to_drive The drive letter to map to. This parameter is required for add operations. It is not applicable to other operations.

If you wanted to set up database drive mapping from F: to E: for NODE0, you would use the following command:

```
db2drvmp add 0 F E
```

Note: Database drive mapping does not apply table spaces, containers, or any other database storage objects.

Similarly, to set up database drive mapping from drive E to drive F for NODE1, you would issue the following command:

```
db2drvmp add 1 E F
```

Any setup of, or changes to, database drive mapping do not take effect immediately. To activate the database drive mapping, use the Cluster Administrator tool to bring the DB2 resource offline, then online.

Reconciling Database Drive Mapping

When a database is created on a machine that has database drive mapping in effect, the map is saved on the drive in a hidden file. This is to prevent the database drive from being removed after the database is created. To reconcile the map, run the **db2drvmp reconcile** command for each database partition server that contains the database. (A situation in which you would want to reconcile the database drive mapping would be if you accidentally dropped the database drive map.) The command syntax is as follows:

```
►►──db2drvmp reconcile──┬─────────────┬──┬───────┬──────────────►◄
                        └─node_number─┘  └─drive─┘
```

The parameters are as follows:

node_number The node number of the node to be repaired. If *node_number* is not specified, the command reconciles the mapping for all nodes.

drive The drive to reconcile. If the drive is not specified, the command reconciles the mapping for all drives.

The **db2drvmp** command scans all drives on the machine for database partitions that are managed by the database partition server, and reapplies the database drive mapping to the registry as required.

Example - Setting up DB2 Failover Support in a Mutual Takeover Configuration

The objective for this example is to set up a four-node partitioned database system with failover support in a mutual takeover configuration. In this example, four servers are configured into two MSCS clusters. By using the mutual takeover configuration, when any of the machine fails, the database partition servers configured for that machine will fail over to the alternative machine, as configured using the MSCS software, and run as a logical node on the alternative machine.

There are two MSCS clusters in the resulting configuration. Each cluster has:

- Two servers, each with 64 MB of memory and one local SCSI disk of 2 GB
- One SCSI disk tower that has three shared SCSI disks of 2 GB each.

In addition, each machine has one 100X Ethernet Adapter card installed.

Each machine has the following software installed:

- Windows NT Version 4.0 Enterprise Edition with the MSCS feature installed
- DB2 Universal Database Extended Enterprise Edition Version 5.0.

The resulting network configuration is as follows:

Server 1: • Machine name:db2test1 • TCP/IP hostname:db2test1 • IP Address: 9.9.9.1 (subnet mask: 255.255.255.0 • MSCS cluster name: ClusterA	Server 2: • Machine name:db2test2 • TCP/IP hostname:db2test2 • IP Address: 9.9.9.2 (subnet mask: 255.255.255.0 • MSCS cluster name: ClusterA
Server 3: • Machine name:db2test3 • TCP/IP hostname:db2test3 • IP Address: 9.9.9.3 (subnet mask: 255.255.255.0 • MSCS cluster name: ClusterB	Server 4: • Machine name:db2test4 • TCP/IP hostname:db2test4 • IP Address: 9.9.9.4 (subnet mask: 255.255.255.0 • MSCS cluster name: ClusterB

All machines in the network are configured with TCP/IP and connected to a private LAN using an Ethernet 100 T-base Hub. In the absence of a Domain Name Server (DNS), all machines have a local TCP/IP *hosts* file. Each *hosts* file contains the following entries:

```
9.9.9.1 db2test1   # for Server 1
9.9.9.2 db2test2   # for Server 2
9.9.9.3 db2test3   # for Server 3
9.9.9.4 db2test4   # for Server 4
9.9.9.5 ClusterA   # for MSCS Cluster 1
9.9.9.6 ClusterB   # for MSCS Cluster 2
9.9.9.7 db2tcp     # for DB2 remote client connection
```

Preliminary Tasks

Before you perform the following tasks, it is assumed that all four machines belong to the same domain, called DB2NTD, the machine db2test1 is the domain controller, and the machine db2test2 is the backup domain controller:

1 Create a domain account for DB2 that is a member of the Domain Administrators group. Use the account for performing all tasks:

- Set the user name to db2nt.
- Set the password to db2nt.

2 Create a second domain account with the "*password never expires*" characteristic. This account will be associated with DB2 services:

- Set the user name to db2mpp.
- Set the password to db2mpp.

3 Install the MSCS feature on the machines db2test1 and db2test2:

- Name the MSCS cluster ClusterA.
- The cluster IP Address is 9.9.9.5.
- Share disk D: will be used by the MSCS software.
- Share disks E: and F: will be used by DB2.

4 Install the MSCS feature on the machines db2test3 and db2test4:

- Name the MSCS cluster ClusterB.
- The cluster IP Address is 9.9.9.6.s
- Select share disk D: will be used by the MSCS software
- Share disks E: and F: will be used by DB2.

5 Install DB2 Extended Enterprise Edition on the machine db2test1:

- Select the "*This machine will be the instance-owing database partition server*" option.
- The account for the DB2 service is db2mpp. The password is db2mpp.

6 Install DB2 Extended Enterprise Edition on the machines db2test2, db2test3, and db2test4:

- Select the "*This machine will be a new node on an existing partitioned database system*" option.
- Select db2test1 as the instance-owning machine.
- The account for the DB2 service is db2mpp. The password is db2mpp.
- Install the software on *C:\SQLLIB*.

Setting up the db2test1 Machine

To set up the db2test1 machine, perform the following tasks:

1 Log on as user db2nt, which belongs to the local Administrators group. The password is db2nt.

2 Register the DB2 resource type with MSCS. From the command line, enter:

```
db2wolf i
```

3 From the Cluster Administrator tool, create DB2 groups and the dependent resources. For detailed instructions, see “Setting up a Mutual Takeover Configuration” on page 206:

a. Start the Cluster Administrator tool.

b. Create a new group called DB2NODE0.

c. Create a new physical disk resource with the following parameters:

- Name is Disk E:

- Resource Type is Physical Disk
- Group is DB2NODE0
- No dependency
- Disk is E: (physical disk 4).

4 Bring the disk resource online.

5 Create a new directory, E:\profiles, on the E: drive.

6 Create an IP Address resource with the following parameters:

- Name is IP Address for DB2
- Resource Type is IP Address
- Group is DB2NODE0
- No dependency
- Address is 9.9.9.5
- Subnet mask is 255.255.255.0
- Network to use is ClusterA.

7 Bring the IP Address resource online.

8 Create a Network Name resource with the following parameters:

- Name is Network Name for DB2
- Resource Type is Network Name
- Group is DB2NODE0
- Dependency is IP Address for DB2
- Name is DB2WOLF.

9 Bring the Network Name resource online.

10 Create a File Share resource with the following parameters:

- Name is File Share for DB2
- Resource Type is File Share
- Group is DB2NODE0
- Dependency is Disk E:
- Share name is DB2MPP
- Path is E:\profiles.

11 Bring the File Share resource online.

12 Create a new DB2 resource with the following parameters:

- Name is DB2MPP-0

- Resource Type is DB2
- Group is DB2NODE0
- Dependency is all available resources in the group

Note: Do not bring the DB2 resource online at this time.

13 Create a new group called DB2NODE1.

14 Create a new physical disk resource with the following parameters:

- Name is Disk F:
- Resource Type is Physical Disk
- Group is DB2NODE1
- No dependency
- Disk is F: (physical disk 5).

15 Bring the disk resource online.

16 Create a new DB2 resource with the following parameters:

- Name is DB2MPP-1
- Resource Type is DB2
- Group is DB2NODE1
- Dependency is all available resources in the group

Note: Do not bring the DB2 resource online at this time.

17 Migrate the instance to use failover support. Enter the following command:

```
db2iclus migrate /i:db2mpp /c:ClusterA /m:db2test1 /p:\\DB2WOLF\DB2MPP
```

18 Log out from the db2nt account.

Enable the DB2 Resource to Fail Over to the db2test2 Machine

To enable the DB2 resource (database partition server) to fail over to the db2test2 machine, perform the following tasks:

1 On the machine db2test1, log on as user db2mpp, which is the account associated with DB2 services. The password is db2mpp.

2 To enable the DB2 resource to fail over to the db2test2 machine, enter the following command:

```
db2iclus add /i:DB2MPP /c:ClusterA /m:db2test2 /u:db2mpp,db2mpp
```

Register the Database Drive Mapping for ClusterA

To register the database drive mapping for MSCS cluster ClusterA, perform the following tasks:

1 On the machine db2test1, log on as user db2mpp, which is the account associated with DB2 services. The password is db2mpp.

2 To register the database drive mapping, enter the following commands:

```
db2drvmp add 0 F E

db2drvmp add 1 E F
```

Bring the DB2 Resources for ClusterA Online

To bring the DB2 resources for ClusterA online, use the Cluster Administrator tool to start the following resources:

- DB2MPP-0
- DB2MPP-1.

Setting up the db2test3 Machine

To set up the db2test3 machine, perform the following tasks:

1 Log on as user db2nt, which belongs to the local Administrators group. The password is db2nt.

2 Register the DB2 resource type with MSCS. From the command line, enter:

```
db2wolf i
```

3 From the Cluster Administrator tool, create DB2 groups and the dependent resources. For detailed instructions, see “Setting up a Mutual Takeover Configuration” on page 206:

a. Start the Cluster Administrator tool.

b. Create a new group called DB2NODE2.

c. Create a new physical disk resource with the following parameters:

   - Name is Disk E:
   - Resource Type is Physical Disk
   - Group is DB2NODE2
   - No dependency

- Disk is E: (physical disk 4).

4 Bring the disk resource online.

5 Create a new DB2 resource with the following parameters:

- Name is DB2MPP-2
- Resource Type is DB2
- Group is DB2NODE2
- Dependency is all available resources in the group

Note: Do not bring the DB2 resource online at this time.

6 Create a new group called DB2NODE3.

7 Create a new physical disk resource with the following parameters:

- Name is Disk F:
- Resource Type is Physical Disk
- Group is DB2NODE3
- No dependency
- Disk is F: (physical disk 5).

8 Bring the disk resource online.

9 Create a new DB2 resource with the following parameters:

- Name is DB2MPP-3
- Resource Type is DB2
- Group is DB2NODE3
- Dependency is all available resources in the group

Note: Do not bring the DB2 resource online at this time.

10 Migrate the instance to use failover support. Enter the following command:

```
db2iclus migrate /i:db2mpp /c:ClusterB /m:db2test3 /p:\\DB2WOLF\DB2MPP
```

11 Log out from the db2nt account.

Enable the DB2 Resource to Fail Over to the db2test4 Machine

To enable the DB2 resource (database partition server) to fail over to the db2test4 machine, perform the following tasks:

1 On the machine db2test3, log on as user db2mpp, which is the account associated with DB2 services. The password is db2mpp.

2 To enable the DB2 resource to fail over to the db2test4machine, enter the following command:

```
db2iclus add /i:DB2MPP /c:ClusterB /m:db2test4 /u:db2mpp,db2mpp
```

Register the Database Drive Mapping for ClusterB

To register the database drive mapping for MSCS cluster ClusterB, perform the following tasks:

1 On the machine db2test3, log on as user db2mpp, which is the account associated with DB2 services. The password is db2mpp.

2 To register the database drive mapping, enter the following commands:

```
db2drvmp add 2 F E

db2drvmp add 3 E F
```

Bring the DB2 Resources for ClusterB Online

To bring the DB2 resources for ClusterB online, use the Cluster Administrator tool to start the following resources:

- DB2MPP-2
- DB2MPP-3.

Administering DB2 in an MSCS Environment

If you are using MSCS clusters, your DB2 instance requires additional planning with regards to daily operation, database deployment, and database configuration. For DB2 to execute transparently on any MSCS node, additional administrative tasks must be performed. All DB2 dependent operating system resources must be available on all MSCS nodes. Some of these operating system resources fall outside the scope of MSCS. That is, they cannot be defined as an MSCS resource. You must ensure that each system is configured such that the same operating system resources are available on all MSCS nodes. The sections that follow describe the additional work that must be done.

Starting and Stopping DB2 Resources

You must start and stop DB2 resources from the Cluster Administrator tool. Several mechanisms are available to start a DB2 instance such as the **db2start** command, and the **Services** option from the Control Panel. However, if DB2 is not started from the Cluster Administrator, the MSCS software will not be aware of the state of DB2 instance. If a DB2 instance is started using the Cluster Administrator and stopped using the **db2stop** command, the MSCS software will interpret the **db2stop** command as a software failure and attempt to restart DB2. (The current MSCS interfaces do not support notification of a *resource state*.)

Similarly, if you use **db2start** to start a DB2 instance, MSCS cannot detect that the resource is online. If a database partition server failed, MSCS would not bring the DB2 resource online on the failover machine in the cluster.

Three operations can be applied to a DB2 instance:

Online This operation is equivalent to using the **db2start** command. If DB2 is already active, this operation can be used simply to notify MSCS that DB2 is active. Any errors during this operation will be written to the Windows NT Event Log.

Offline This operation is equivalent to using the **db2stop** command. If there are any active connections to an instance, this operation will fail. This is consistent with the behavior of **db2stop**.

Fail resource This operation is equivalent to using the **db2stop** command with the **force** option specified. DB2 will disconnect all applications off the DB2 system and stop all database partition servers.

Running Scripts

You can execute scripts both before and after a DB2 resource is brought online. These scripts *must* reside in the instance profile directory that is specified for the DB2INSTPROF environment variable. This directory is the directory path that is specified by the **-p** parameter of the **db2icrt** command. You can obtain this value by issuing the following command:

```
db2set -i: instance_name DB2INSTPROF
```

This file path must be on a clustered disk so that the instance directory is available on all cluster nodes.

These script files are not required, and are only executed if they are found in the instance directory. They are launched by the MSCS Cluster Service in the background. The script files must redirect standard output to record any output as a result of commands within the script file. The output is not displayed to the screen.

By default, the same script will be used by every database partition server in the instance. If you need to distinguish among the different database partition servers in the instance, use different assignments of the DB2NODE environment variable to target specific node numbers.

Running Scripts Before Bringing DB2 Resources Online

If you want to run a script before you bring a DB2 resource online, the script *must* be named *db2cpre.bat*. DB2 calls functions that will launch this batch file from the Windows NT command line processor and wait for the command line processor to complete execution before the DB2 resource is brought online. You can use this batch file for tasks such as modifying the DB2 database manager configuration. You may want to change some database manager parameter values if the failover system is constrained, and you must reduce the system resources consumed by DB2.

The commands placed in the *db2cpre.bat* script should execute synchronously. Otherwise the DB2 resource may be brought online before all tasks in the script are completed, which may result in unexpected behavior. Specifically, **db2cmd** should not be executed in the *db2cpre.bat* script, because it, in turn, launches another command processor, which will execute DB2 commands asynchronously to the **db2cmd** program.

If you want to use DB2 CLP commands in the *db2cpre.bat* script, the commands should be placed in a file and executed as a CLP batch file from within a program that initializes the DB2 environment for the DB2 command line processor, then waits for the completion of the DB2 command line processor. For example:

```
#include <windows.h>

int WINAPI DB2SetCLPEnv_api(DWORD pid);

void main (int argc, char *argv [ ] )
{
      STARTUPINFO         startInfo   = {0};
      PROCESS_INFORMATION pidInfo     = {0};
      char     title  [32]   = "Run Synchronously";
      char     runCmd [64]  =
                             "DB2 -z c:\\run.out -tvf c:\\run.clp";
/* Invoke API to setup a CLP Environment */
      if ( DB2SetCLPEnv_api (GetCurrentProcessId ()) == 0 )(1 - see notes below)
      {
         startInfo.cb          = sizeof(STARTUPINFO);
         startInfo.lpReserved  = NULL;
         startInfo.lpTitle     = title;
         startInfo.lpDesktop   = NULL;
         startInfo.dwX         = 0;
         startInfo.dwY         = 0;
         startInfo.dwXSize     = 0;
         startInfo.dwYSize     = 0;
         startInfo.dwFlags     = 0L;
         startInfo.wShowWindow = SW_HIDE;
         startInfo.lpReserved2 = NULL;
         startInfo.cbReserved2 = 0;


               if ( CreateProcessA( NULL,
                              runCmd, (2)
                              NULL,
                              NULL,
                              FALSE,
                              NORMAL_PRIORITY_CLASS | CREATE_NEW_CONSOLE,
                              NULL,
                              NULL,
                              &startInfo,
                              &pidInfo))
         {
            WaitForSingleObject (pidInfo.hProcess, INFINITE);
            CloseHandle (pidInfo.hProcess);
            CloseHandle (pidInfo.hThread);
         }
      }
      return;
}
```

Notes:

1. The API DB2SetCLPEnv_api is resolved by the import library DB2API.LIB. This API sets an environment that allows CLP commands to be invoked. If this program is invoked from the *db2cpre.bat* script, the command processor will wait for the CLP commands to complete.
2. runCmd is the name of the script file that contains the DB2 CLP commands.

A sample program called *db2clpex.exe* can be found in the *MISC* subdirectory of the DB2 install path. This executable is similar to the example provided, but accepts the DB2 CLP command as a command line argument. If you want to use this sample program, copy it to the *BIN* subdirectory. You can use this executable in the *db2cpre.bat* script as follows. Note that the example has been split at `-tvf` for formatting reasons.

```
  db2clpex "DB2 -Z \\cluster_name\share_name\wolfinst\pre.log
   -tvf \cluster_name\share_name\wolfinst\pre.clp"
```

All DB2 **attach** commands or **connect** statements should explicitly specify a user, otherwise they will be executed under the user account associated with the cluster service. CLP scripts should also complete with the **terminate** command to end the CLP background process.

The following is an example of a *db2cpre.bat* file:

```
db2cpre.bat : (1 - see notes below)
------------------------
set DB2INSTANCE=WOLFINST  (2)
dir \\network_name\share_name\wolfinst > \\network_name\share_name\wolfinst.log (3)
db2clpex "db2 -z \\network_name\share_name\wolfinst\pre-%DB2NODE%.log (4)
   -tvf \\network_name\share_name\wolfinst\pre.clp" (5, 6)
------------------------

PRE.CLP (7)
------------------------
update dbm cfg using MAXAGENTS 200;
get dbm cfg;
terminate;
------------------------
```

Notes:

1. The *db2cpre.bat* script executes under the user account associated with the Cluster Service. If DB2 actions are required, the user account associated with the Cluster Service must be a valid SQL identifier, as defined by DB2.
2. Ensure that the DB2INSTANCE environment variable is set so that the script executes under the correct DB2 instance
3. The directory must use the universal naming convention, and include both the cluster name and the share name (that is, the name of the MSCS shared disk resource that is used for the instance directory). The form is as follows: *\\network_name\share_name*.
4. The name of the log file must be different for each node to avoid file contention when both logical nodes are brought online at the same time.
5. *db2clpex.exe* is the sample program previously provided using the command line argument to specify the CLP command to execute. (This line of the example has been split at `-tvf` for formatting reasons.)
6. The *db2clpex.exe* sample program must be made available on all MSCS cluster nodes.
7. The CLP commands in this example set a limit on the number of agents.

Running Scripts After Bringing DB2 Resources Online

If you want to run a script after you bring a DB2 resource online, it *must* be named *db2cpost.bat*. The script will be executed asynchronously from MSCS after the DB2 resource has been successfully brought online. The **db2cmd** command can be used in this script to execute DB2 CLP script files. Use the **-c** parameter of the **db2cmd** command to specify that the utility should close all windows on completion of the task. For example:

```
db2cmd -c db2 -tvf mycmds.clp
```

The **-c** parameter must be the first argument to the **db2cmd** command, as it prevents orphaned command processors in the background.

The *db2cpost.bat* script is useful if you want to perform database activities immediately after the DB2 resource fails over and becomes active. For example, you can restart or activate databases in the instance so that they are primed for user access.

The following is an example of a *db2cpost.bat* script:

```
db2cpost.bat (1 - see notes below)
------------------------
SET DB2INSTANCE=WOLFINST (2)
db2cmd -c db2 -z \\network_name\share_name\wolfinst\post-%DB2NODE%.log (3)
   -tvf \\network_name\share_name\wolfinst\post.clp (4, 5)
------------------------

POST.CLP (6)
------------------------
restart database SAMPLE;
connect reset;
activate database SAMPLE;
terminate;
------------------------
```

Notes:

1. The *db2cpost.bat* script executes under the user account associated with the Cluster Service. If DB2 actions are required, the user account associated with the Cluster Service must be a valid SQL identifier, as defined by DB2.
2. Ensure that the DB2INSTANCE environment variable is set so that the script executes under the correct DB2 instance
3. The name of the log file must be different for each node to avoid file contention when both logical nodes are brought online at the same time.
4. The directory must use the universal naming convention, and include both the cluster name and the share name (that is, the name of the MSCS shared disk resource that is used for the instance directory). The form is as follows:

\\*network_name**share_name*. (This line of the example has been split at `-tvf` for formatting reasons.)

5. The **db2cmd** command can be used because the *db2cpost.bat* script can execute asynchronously. The **-c** parameter must be used to terminate the command processor.
6. The CLP script in this example contains commands to restart and activate the database. This script returns the database to an active state immediately after the database manager is started. If multiple DB2 resources are brought online on the same machine at the same time, the **restart database** command may fail because another logical node is activating the database. If this occurs, rerun the script to ensure that the database is restarted correctly.

Database Considerations

An instance with MSCS support can support both databases that will fail over, and databases that are local to a specific system. To ensure that a database will fail over, define all its dependent resources as cluster resources, and place them in the same resource group so that the Cluster Administrator tool can treat the resources as a single unit. In addition, you should configure a dependency using the Cluster Administrator to ensure that all dependent resources are brought online and available before the DB2 resource is started.

When you create a database, ensure that the database path refers to a share disk. This allows the database to be seen on all MSCS nodes. All logs and other database files must also refer to clustered disks for DB2 to failover successfully. If you do not perform these steps, a DB2 system failure will occur as it will seem to DB2 that files have been deleted or are unavailable.

Also ensure that the database manager and database configuration parameters are set so that amount of system resources consumed by DB2 is supported on either MSCS node. The *autorestart* database configuration parameter should be set to `ON` so that the first database connection on failover will bring the database to a consistent state. The default setting for *autorestart* is `ON`. The database can also be brought to a ready state by using the *db2cpost.bat* script to restart and activate the database. This method is preferred, because there will be no dependency on *autorestart*, and the database is brought to a ready state independent of a user connection request.

User and Group Support

DB2 relies on Windows NT for user authentication and group support. For a DB2 instance to fail over from one MSCS node to another in a seamless fashion, each MSCS node must have access to the same Windows NT security databases. You can achieve this by using Windows NT Domain Security.

Define all DB2 users and groups in a Domain Security database. The MSCS nodes must be members of this Domain or the Domain must be a Trusted Domain. DB2 will then use the Domain Security database for authentication and group support, independent of which MSCS node DB2 is executing on.

If you are using local accounts, the accounts must be replicated on each MSCS node. This approach is not recommended because it is error prone and requires dual maintenance.

DCE Security is also a supported authentication mode, providing that all MSCS nodes are clients in the same DCE cell.

You should associate the MSCS service with a user account that follows DB2 naming conventions. This allows the MSCS service to perform actions against DB2 that may be required in the *db2cpre.bat* and *db2cpost.bat* scripts.

Communications Considerations

DB2 supports two LAN protocols in an MSCS Environment:

- TCP/IP
- NetBIOS

TCP/IP is supported because it is a supported cluster resource type. To enable DB2 to use TCP/IP as a communications protocol, create an IP Address resource and place it in the same group as the DB2 resource that represents the database partition server that you want to use as a coordinator node for remote applications. Then create a dependency using the Cluster Administrator tool to ensure that the IP resource is online before the DB2 resource is started. DB2 clients can then catalog TCP/IP node directory entries to use this TCP/IP address.

The TCP/IP port associated with the *svcename* database manager configuration parameter must be reserved for use by the DB2 instance on all machines that participate in the instance. The service name associated with the port number must also be the same in the *services* file on all machines.

Although NetBIOS is not a supported cluster resource, you can use NetBIOS as a LAN protocol because the protocol ensures that NetBIOS names are unique on the LAN. When DB2 registers a NetBIOS name, NetBIOS ensures the name is not in use on the LAN. In a failover scenario, when DB2 is moved from one system to another, the *nname* used by DB2 will be deregistered from one partner machine in the MSCS cluster and registered on the other machine.

DB2 NetBIOS support uses NetBIOS Frames (NBF). This protocol stack can be associated with different logical adapter numbers (LANA). To ensure consistent NetBIOS access to the server, the LANA associated with the NBF protocol stack should be the same on all clustered nodes. You can configure this by using the **Networks** option from the Control Panel. You should associate NBF with LANA 0, as this is the default setting expected by DB2.

System Time Considerations

DB2 uses the system time to timestamp certain operations. All MSCS nodes that participate in DB2 failover must have the system time zone and system time synchronized to ensure DB2 behaves consistently on all machines.

Set the system time zone using the **Date/Time** option from the Control Panel dialog. MSCS has a time service that synchronizes the date and time when the MSCS nodes join to form a cluster. The time service, however, only synchronizes the time every 12 hours, which may result in problems if the time is changed on one system and DB2 fails over before the time is synchronized.

If the date/time is changed on one of the MSCS cluster nodes, the time should be manually synchronized on the other cluster nodes using the command:

```
net time /set /y \\remote node
```

Where *remote node* is the machine name of the cluster node.

Administration Server and Control Center Considerations

The DB2 Administration Server is (optionally) created during the installation of DB2 Universal Database. It is not a partitioned database system. The Control Center uses the services provided by the Administration Server to administer DB2 instances and databases.

In an MSCS environment, a DB2 instance can reside on multiple MSCS nodes. This implies that a DB2 instance must be cataloged on multiple systems under the Control Center so that the instance remains accessible, regardless of which MSCS node the DB2 instance is active on.

The Administration Server instance directory is not shared. You must mirror all user-defined files in the Administration Server directory to all MSCS nodes to provide the same level of administration to all MSCS nodes. Specifically, you must make user scripts and scheduled executables available on all nodes. You must also ensure that scheduled activities are scheduled on both machines in an MSCS cluster.

You may want to have the Administration Server fail over. For the purposes of the following example, assume that you have two MSCS nodes in the cluster, and they are called NODE0 and NODE1. NODE0 has access to a cluster disk that will be used by the Administration Server. Also assume that both NODE0 and NODE1 have an Administration Server. You would perform the following steps:

1. Stop the Administration Server on both machines by executing the **db2admin stop** command on each machine.

2. On all administration client machines, uncatalog all references to the Administration Servers on NODE0 and NODE1 using the **db2 uncatalog admin** command.

3. Drop the Administration Server from NODE1 by executing the **db2admin drop** command from NODE1. (You would only perform this step if you had an Administration Server on both machines.)

4. Determine the name of the Administration Server by issuing the **db2admin** command from NODE0. (The default name is `DB2DAS00`.)

5. In the same resource group that you are using for the DB2 resources, create a DB2 resource named `DB2DAS00` that has dependencies on the IP and disk resources. (If you have a mutual takeover configuration, you would put the resource in the group that holds the DB2 resource for NODE0.) This resource will be used as the MSCS resource that supports the Administration Server.

6. Assuming that NODE0 owns the cluster disk that will be used for the instance directory, execute the following command from NODE0:

   ```
   db2iclus migrate -i:DB2DAS00 -c:cluster_name -p:instprof_path
   ```

   This command adds the Administration Server to the cluster. For details about the **db2iclus migrate** command, see “Migrating an Instance to Use MSCS Support” on page 212.

7. From NODE0, execute the following command to enable the Administration Server to fail over to NODE1:

   ```
   db2iclus add -i:DB2DAS00 -u: account_name,password /m: NODE1
   ```

   For details about the **db2iclus add** command, see “Enabling the DB2 Resource to Fail Over to the Second MSCS Node” on page 214.

8. On NODE1 execute the following command to set DB2DAS00 as the Administration Server:

   ```
   db2set -g db2adminserver=DB2DAS00
   ```

9. On NODE0, modify the start-up properties of DB2DAS00 through the Services program so that it is brought up manually and not automatically, because DB2DAS00 is now controlled by MSCS.

When the Administration Server is enabled for failover, all remote access should use an MSCS IP resource for communicating with the Administration Server. The Administration Server will now have the following properties:

- The Administration Server instance directory will fail over with the Administration Server.
- Clients will only catalog a single node to communicate with the Administration Server, regardless of which MSCS node it is active on.

- Jobs only need to be scheduled once against the Administration Server.
- Local instances can only be controlled by the Administration Server when the Administration Server is active on the same MSCS node as the local instance.
- The Administration Server is not accessible if the Cluster Service is not active.

Limitations and Restrictions

When you run DB2 in an MSCS environment:

- You cannot use physical I/O on shared disks, unless the shared disks have the same physical disk number across both MSCS nodes. You can use logical I/O because the disk is accessed using a partition identifier.
- You must configure all DB2 resource for MSCS support. If you do not, system errors will occur during DB2 runtime (DB2 cannot properly operate in the absence of system resources). For example, if the database logs are not on a MSCS shared disk, DB2 cannot restart the database.
- You must manage a DB2 instance from the Cluster Administrator tool. MSCS will view other mechanisms that are used to start and stop the database manager as software inconsistencies. For example, if you use MSCS to start DB2 and the **db2stop** command to stop DB2, MSCS will detect this as a software failure and will restart the instance. This also means that you should not use the Control Center to start and stop DB2.
- To uninstall DB2, you must first stop MSCS.

Appendix E. National Language Support (NLS)

This appendix contains information about the National Language Support (NLS) provided by DB2, including information about supported languages and code pages. For more detailed information on developing applications, refer to the *Embedded SQL Programming Guide*.

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 14 on page 238 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 14. 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 240
- "Object Names" on page 241
- "Usernames, Group Names, and Instance Names" on page 241
- "DB2SYSTEM Naming Rules" on page 243
- "Workstation Names (nname)" on page 242
- "Password Rules" on page 244

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 239.

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.
2. On Windows NT systems, ensure that no instance name is the same as a service name.

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 239.

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 239.

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 239.

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 239.

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

A

- Access Profile
  - creating 136
  - using 136
- accessing data
  - through DB2 Connect 14
  - through Net.Data or JDBC 16
- accessing multiple servers 7
- Administration Server
  - CCA 94
  - Control Center 94
  - create manually 94
  - logon account 96
  - overview 10, 93
  - remove 97
  - start 96
  - stop 96
  - updating instance lists 95
- Administrator name
  - valid names 46
- ADSM
  - software requirements 41
- agents
  - coordinating 31
  - pools 31
  - purpose 31
  - subagent 31
- Alert Center 10
- APPC
  - server 160
  - software requirements 41, 42, 43
  - support on OS/2
    - Communications Manager for OS/2 42
    - Communications Server for OS/2 42
  - support on Windows NT
    - IBM Communications Server for NT 41
    - Microsoft SNA Server 41
  - supported platforms 41
- application development
  - using Net.Data or JDBC 16
  - using ODBC 101
- application programs
  - connecting to logical node 113
- application programs *(continued)*
  - coordinator node 22
  - migrating to DB2 Extended Enterprise Edition 22
  - serviced by agents 31
- authentication
  - client 57
  - DCS
  - server 147
- authority
  - levels 80, 85

B

- binding
  - utilities and applications 100
- book
  - printing Postscript 184, 185
  - related books 177
  - viewing online 184, 185

C

- capacity
  - disk storage 25
- catalog node 27
- cataloging
  - IPX/SPX node 159
  - NetBIOS node 156
  - TCP/IP node 153
- CD-ROM
  - installing DB2 Client Applications Enablers 57
  - installing DB2 Software Developer's Kit 57, 63
- Client Application Enabler
  - installing 57
  - installing on OS/2 63
  - installing on Windows 3.x 65
  - software requirements 42
- Client Configuration Assistant
  - administering clients communications 71
  - and DB2 setup 93
  - changing privileges 82
  - client profiles 71
    - exporting 137
    - importing 139
    - using 137

Client Configuration Assistant *(continued)*
- discover 141
- discovering databases 72
- overview 12
- searching the network 71, 139
- setting configuration parameters 127, 143
- testing the database connection 73
- using access profiles 136

client profile
- exporting 137
- importing 139

clients
- installing 57
- operating systems supported 5, 57
- overview 5, 57
- verifying the installation 73

CLP
- *See* using the command line processor (CLP)

column
- long-field, partitioning key considerations 28

Command Center
- entering DB2 commands 109
- entering SQL statements 109
- overview 8

commands
- connecting to logical node 113
- db2admin 93, 96
- db2admin create 94
- db2admin drop 97
- db2admin setid 96
- db2admin start 96
- db2admin stop 96
- db2icrt 87
- db2idrop
- db2ilist 90
- db2iset 95
- db2nlist 95
- db2set 116, 148
- db2start 91, 95
- db2stop 91
- echo 117, 118
- set 117

communication protocols
- APPC 41, 42, 43, 160
- IPX/SPX 41, 42, 43, 157
- Named Pipes 41, 42, 43, 149
- NetBIOS 41, 42, 43, 153
- TCP/IP 41, 42, 43, 150

communications
- configuring the client 71

communications *(continued)*
- configuring the server 75, 147
- managing 147
  - Client Configuration Assistant 71
  - Control Center 12, 75
  - fast communication manager 30
- supported protocols
  - IPX/SPX 157
  - Named Pipes 149
  - NetBIOS 153
  - TCP/IP 150

concurrency
- improving with logical nodes 29

configuration parameters
- intra-parallel 31
- max_time_diff 46
- maxagents 31
- num_poolagents 31
- setting DB2 57, 82, 115, 127, 147
- SYSADM_GROUP 82

configuring
- APPC 160
- Named Pipes 149
- NetBIOS 153
- ODBC driver 104, 106
- TCP/IP 150

configuring client communications
- setting configuration parameters 127
- using the Client Configuration Assistant 71
- using the command line processor 147

configuring communications
- using Client Configuration Assistant 71
- using Control Center 75

configuring server communications
- generating an access profile 53
- setting DB2COMM 136, 147
- using the command line processor 147
- using the Control Center 75

configuring servers 75, 147

connecting to a database 80

connecting to the Internet using Net.Data 16

Control Center
- components 8
- overview 8

controlling the rah command 196

coordinator agent
- description 31

coordinator node
- description 22

cost-based query optimizer
  description 22
creating the sample database
  using First Steps 80
  using the command line processor 80

D

DAS considerations 94
database
  definition 21
  naming rules 240
  partition 21
database access
  two-phase commit 7
database administration tools
  Control Center 8
  overview 8
database alias
  naming rules 240
database manager
  definition 21
  starting 90, 91
  stopping 90
database objects
  naming rules 241
database partition server
  definition 21
DB2 Client Application Enabler
  accessing databases 6
  installing
    Windows 3.x 65
    Windows 95 59
    Windows NT 59
  overview 5, 6, 57
  platforms supported 5
DB2 Client Application Enablers CD-ROM
  Control Center 94
  installing clients from 57
DB2 Connect
  overview 3, 4, 14
DB2 Enterprise Edition
  memory requirements 35
DB2 Extended Enterprise Edition
  configuration planning 35
  hardware configuration 25
DB2 Installer
  See installation
DB2 server, definition x
DB2 Snapshot Monitor
  overview 11
db2 terminate command 91
DB2 Universal Database
  components included
    Client Configuration Assistant 12
    Control Center 8
    DB2 Client Application Enabler 5
    DB2 Performance Monitor 12, 129
    DB2 Snapshot Monitor 11, 16
    Visual Explain 11, 12
  DAS considerations 93
  getting started 79
  overview 8
  platforms supported 8
  software requirements 41
db2_all 191, 192
db2_call_stack 192
db2_kill 192
db2.ini file
  environment keywords 119
db2admin command
  db2admin create 94
  db2admin drop 97
  db2admin setid 96
  db2admin start 96
  db2admin stop 96, 97
DB2COMM 147
DB2DMNBCKCTLR
  registry value 121
db2fcmdm daemon
  description 30
db2icrt command 87
db2idrop command
db2ilist command 90
DB2INSTANCE 90
DB2NBADAPTERS 154
DB2NODE
  connecting to logical node 113
  overview 29
db2set command
  using 116, 148
db2start command 91, 94
db2stop command 91
developing applications
  using Net.Data or JDBC 16
Discovery
  administration configuration file 95
  choosing a discovery method 141
  client configuration 93

Discovery *(continued)*
  searching the network 139
  setting Discovery parameters 143
disk requirements
  client 39
  server 39
Distributed Computing Environment
  software requirements 41, 42
documentation
  related publications 177

E

eliminating duplicate entries from machine list 195
environment variables 30
  DB2INSTANCE 90, 125
  db2instprof 125
  DB2NODE 122
    connecting to logical node 113
  db2path 125
execs
  db2perfc.exe 132
  DB2PERFI.EXE 130
  DB2PERFR.EXE 130
Extended Enterprise Edition
  capacity 25
  storage 25
Extended Enterprise Edition configuration
  logical nodes 29
  security for an instance 29
  shared nothing definition 21

F

failover
  adding database partition server to instance 214
  Administration Server considerations 234
  communications considerations 233
  considerations for administering DB2 226
  Control Center considerations 234
  creating DB2 instance 212
  database considerations 232
  disabling DB2 resource failover on second MSCS node 216
  enabling DB2 resource to fail over to second MSCS node 214
  failback considerations 216
  hot standby 199
  limitations 236
  migrating default instance for failover support 212
failover *(continued)*
  mutual takeover 200
  mutual takeover configuration example
    bringing DB2 resource online for ClusterA 224
    bringing DB2 resource online for ClusterB 226
    enabling DB2 resource to fail over to db2test2 machine 223
    enabling DB2 resource to fail over to db2test4 machine 225
    objectives 219
    preliminary tasks 220
    registering database drive mapping for ClusterA 224
    registering database drive mapping for ClusterB 226
    setting up db2test1 machine 221
    setting up db2test3 machine 224
  overview 197
  planning 197
  reconciling database drive mapping 218
  registering DB2 resource type 201
  restrictions 236
  running scripts after DB2 resource brought online 231
  running scripts before DB2 resource brought online 228
  running scripts, overview 227
  setting database drive mapping for mutual takeover 217
  setting up DB2 instance for failover support 211
  setting up hot standby configuration 202
  setting up mutual takeover configuration 206
  starting and stopping DB2 resources 227
  system time considerations 234
  types 199
  user and group support 232
fast communication manager (FCM)
  db2fcmdm daemon 30
  description 30
finding topics 185
fixed disks
  hardware requirements 39

G

GB (gigabyte) 25
gigabyte (GB) 25

H

hardware requirements
- fixed disk 39

help
- See information 177

I

IBMCATGROUP nodegroup 27
IBMDEFAULTGROUP nodegroup 27
IBMTEMPGROUP 27
IBMTEMPGROUP nodegroup 26
information
- online help 177
- ordering 184
- overview 177
- related publications 177
- viewing online 184, 185

installation
- CID using SystemView LAN 50, 61
- clients
  - OS/2 63
  - Windows 3.X 65
  - Windows 95 59
  - Windows NT 59
- DB2 clients
  - OS/2 63
  - Windows 3.1x 65
  - Windows 95 59
  - Windows NT 59
- disk requirements
  - client 39
  - server 39
- errors 50, 61
- log 50, 61
- memory requirements
  - client 35
  - server 35
- mounting the CD-ROM
  - AIX
  - Solaris
- Netscape 186
- verifying the installation 51

installing Extended Enterprise Edition
- enabling FCM communications 46
- preparations 45

instance owner 30
instance-owning machine
- definition 30

instances
- creating multiple 85
- definition 29, 85
- listing 90
- multiple on one processor 30
- naming restrictions 87, 241
- removing 92
- running multiple 92
- selecting 90
- starting 90, 91
- stopping 90

inter-operator parallelism 23
IPX/SPX
- server 157
- software requirements 42
- supported platforms 157
- verifying the connection 73

issuing commands to multiple database partition servers 191

J

Java Database Connectivity 16
Java, running programs 107
JDBC
- *See* Java Database Connectivity

JDBC, running programs 107
Journal 10

L

logging off the system 80
logging on to the system 80
Logical LAN Adapter Number (Lana Number)
- default value 155

logical node
- connecting to 113
- default 113
- improving concurrency 29
- purpose 28
- supported configurations 29

long-field columns
- partitioning key considerations 28

M

managing connections
- client
  - overview 12
  - using the Client Configuration Assistant 12, 71

managing connections *(continued)*
  server
    overview 12
    using the Client Configuration Assistant 71
    using the Command Line Processor 147
managing databases using the Control Center 8
managing server communications
  overview 12
manuals
  See book
memory requirements
  client 35
  estimating 35
  recommended 35
  server 35
messages
  getting online help 177
Microsoft ODBC Driver Manager 103
Microsoft SNA Server
  version required 43
migration
  of applications 22
multipartition nodegroup
  See nodegroup

N

Named Pipes
  server 149
  software requirements 42
naming rules
  database 240
  database alias 240
  database objects 241
  general 239
  groups 241
  instance names 241
  password 244
  userids 241
  username 59, 241
  workstation name (nname) 242
National Language Support (NLS)
  code set 237
  codepage support 237
  determining codepage 238
Nbf
  network route required by DB2 155
Net.Data 42
  overview 16
NetBIOS
  autostarting NetBIOS 156
  on the client 42
  on the server 147, 153
  supports SEARCH discovery 142
network route
  setting to Nbf 154, 155
NLS
  See National Language Support (NLS)
node
  definition 21
  in the same instance 29
nodegroup
  IBMCATGROUP 27
  IBMDEFAULTGROUP 27
  multipartition nodegroup 25

O

ODBC
  registering the driver manager 103
  running ODBC-enabled applications 101
  running programs 101
odbcad32.exe 103
operating systems
  client 5
  server 57
ordering related publications 177, 184

P

parameters
  SYSADM_GROUP 82
partition
  definition 21
partitioning key
  definition 28
partitioning map
  overview 28
password
  naming rules 244
Performance Monitor
  using 11
  using with DB2 129
  Windows NT Performance monitor
    configuring 129
    using 129
planning
  DB2 configuration 35
  DB2 Connect configuration 35

pools
  agent 29
prefix sequences 194
printing postscript 185
privileges
  required 82
product
  descriptions 3
  overview 3
profile registry
  using 115
protocols
  APPC 160
  IPX/SPX 157
  Named Pipes 149
  NetBIOS 153
  TCP/IP 150

Q

query
  cost-based optimizer 22

R

rah 191, 192
RAHOSTFILE 195
RAHOSTLIST 195
registering
  ODBC driver manager 103
registry value
  DB2DMNBCKCTLR 121
registry values
  db2account 120
  db2codepage 121
  db2comm 125, 147
  db2country 121
  db2dbdft 121
  db2discoverytime 121
  db2include 122
  db2instdef 90, 122
  db2libpath 123
  db2nbadapters 126, 154
  db2netreq 122, 126
  db2options 123
  db2portrange 126
  db2slogon 123
  db2timeout 123
  db2tracename 124
  db2traceon 124
registry values *(continued)*
  db2trcflush 124
  db2trcsyserr 124
  db2yield 125
  general 120
  setting 115
removing
  administration server 97
restrictions
  instance name 87, 241
running applications
  database client 99
  ODBC considerations 101

S

sample database
  See creating the sample database
scaling the database system
  definition 28
Script Center 10
SDK Kit
  See Software Developer's Kit
security
  for an instance 29
selecting an instance 90
setting
  environment variables 115
setting configuration parameters 147
setting the default environment profile for rah 196
setting up client communications
  using the Client Configuration Assistant 71
setting up server communications
  using the command line processor 147
  using the Control Center 75
shared-nothing configuration
  definition 21
  storage 25
SMP
  See symmetric multiprocessor architecture
snapshot monitoring considerations 11
Software Developer's Kit
  overview 13
software registration 82
software requirements
  communication protocols 41
  DB2 Client Application Enabler 41, 42, 43
  DB2 Connect 41
  DB2 SDK 41, 42, 43
  DB2 Universal Database 41

software requirements *(continued)*
  Net.Data 42, 43
specifying list of machines for rah 195
SQL
  access plans
    viewing using Visual Explain 12
starting
  database manager 91
  DB2 91
stopping
  DB2 91
subagent
  function 31
symmetric multiprocessor architecture 28
SYSADM
  privileges
    controlling 82
SYSADM_GROUP parameter 82
SYSCTRL 85
SYSMAINT 85
system configuration
  with DB2 Connect 14
  with DB2 Universal Database 7
system environment variables
  DB2INSTANCE 90
system management 90
SystemView LAN 42

T

table
  in multinode nodegroup 28
  in single-node nodegroup 28
  maximum size 25
TB (terabyte) 25
TCP/IP
  preventing socket collisions 150
  resolving host address 153
  server 150
  set with db2set 125
  software requirements 42
  supported to host systems 20
  supports only one-phase commit 165
  supports SEARCH discovery 142
  troubleshooting 150
temporary table spaces
  See IBMTEMPGROUP
terabyte (TB) 25
Tools Setting 10
transparent parallelism 22

U

user privileges
  See privileges
username
  naming rules 241
using the command line processor (CLP)
  connecting to a database 73, 80
  updating the database manager configuration 152, 155, 159, 161
  verifying the connection 73
utilities
  binding 99

V

verifying the connection
  using the Client Configuration Assistant 73
  using the command line processor
    IPX/SPX 157
    Named Pipes 149
    NetBIOS 153
    TCP/IP 150, 160
verifying the installation
  client 73
  server 51
Visual Explain
  overview 11, 12

W

Windows NT
  applications 132
  database manager performance object 131
  databases 131
  Performance Monitor 129
Windows NT Performance Monitor
  using with DB2 129

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

Part Number: 09L6713

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

09L6713

S09L-6713-00