

**IBM DB2 Universal Database
Installing and Configuring DB2 Clients
Version 5.2**

IBM DB2 Universal Database

IBM

Installing and Configuring DB2 Clients

Version 5.2

IBM DB2 Universal Database

IBM

Installing and Configuring DB2 Clients

Version 5.2

Before using this information and the product it supports, be sure to read the general information under Appendix D, "Notices" on page 283.

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.

Copyright International Business Machines Corporation 1993, 1998. All rights reserved.
Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Contents

Welcome to DB2 Universal Database	ix
Conventions	ix
Road Map	x

Part 1. Installing DB2 Clients	1
Chapter 1. Planning for Installation	3
Disk Requirements	3
Estimating Fixed Disk Requirements	3
Software Requirements	5
Client Product Requirements	5
Possible Client-to-Server Connectivity Scenarios	9
Chapter 2. Installing DB2 Clients	11
Chapter 3. Installing DB2 Clients on OS/2 Operating Systems	13
Chapter 4. Installing DB2 Clients on Windows 32-Bit Operating Systems	17
Chapter 5. Installing DB2 Clients on Windows 3.x Operating Systems	21
Chapter 6. Installing DB2 Clients on UNIX Operating Systems	25
Before You Begin	25
Installing DB2 Version 5 Clients	26
Step 1. Update Kernel Configuration Parameters	26
Step 2. Mount the DB2 Client Application Enablers CD-ROM	28
Step 3. Perform the Installation	32
Installing DB2 Version 2 Clients	35
Installing DB2 Version 2 Clients on SINIX	35
Installing DB2 Version 2 Clients on SCO OpenServer	36
Create or Assign Groups and Users	37
Create an Instance of the Product	38
Set Up the DB2 Operating Environment	39
Create Links for DB2 Files	39
Chapter 7. Migrating UNIX Clients	41
Migrating Instances	41
Prepare the DB2 Instance for Migration	41
Migrate the DB2 Instance	42
Chapter 8. Installing DB2 Clients on Macintosh Operating Systems	45

Part 2. Configuring DB2 Communications 47

Chapter 9. Configuring Client-to-Server Communications Using the Client Configuration Assistant	49
Verifying the Connection	52

Chapter 10. Configuring Client-to-Server Communications Using the Command Line Processor	55
Configuring Named Pipes on the Client	55
Step 1. Identify and Record Parameter Values	56
Step 2. Configure the Client	56
Step 3. Test the Client-to-Server Connection	59
Configuring TCP/IP on the Client	60
Step 1. Identify and Record Parameter Values	61
Step 2. Configure the Client	62
Step 3. Test the Client-to-Server Connection	67
Configuring NetBIOS on the Client	69
Step 1. Identify and Record Parameter Values	69
Step 2. Configure the Client	70
Step 3. Test the Client-to-Server Connection	75
Configuring IPX/SPX on the Client	77
Step 1. Identify and Record Parameter Values	78
Step 2. Configure the Client	79
Step 3. Test the Client-to-Server Connection	83
Configuring APPC on the Client	85
Step 1. Identify and Record Parameter Values	86
Using the Client Setup Tool to Configure a Windows 3.x Client	115

Part 3. Using DB2 Universal Database 117

Chapter 11. Getting Started with DB2 Universal Database	119
Logging on to the System	119
Connecting to a Database	120

Chapter 12. Working with Instances	123
The Default Instance	123
Instance Directory	124
Creating Instances	124
Listing Instances	126
Setting the Current Instance	126
Updating Instances on UNIX Systems	128
Removing Instances	128

Chapter 13. Running Your Own Applications	131
Binding Database Utilities	131
Running CLI/ODBC Programs	133
Platform Specific Details for CLI/ODBC Access	134

Detailed Configuration Information	143
Running Java Programs	147
Configuring the Environment	148
Java Applications	149
Java Applets	150
Chapter 14. Entering DB2 Commands and SQL Statements	151
Using the Command Center	151
Using the Command Line Processor	153
Interactive Input Mode	153
Command Line Mode	154
Chapter 15. Controlling Your DB2 Environment	157
Controlling the DB2 Profile Registry	158
Setting Your Environment on OS/2	159
Setting Your Environment on Windows 32-Bit Operating Systems	160
Setting Your Environment on UNIX Systems	161
Setting Your Environment on Windows 3.x	161
DB2 Registry Values and Environment Variables	163
Configuration Parameters	169
Setting Database Manager Configuration Parameters on a Client Instance	169
DB2 CLI/ODBC Configuration Keyword Listing	170
Configuration Keywords by Category	170
Configuration Keyword Descriptions	173

Part 4. Advanced Installation and Configuration 203

Chapter 16. Unattended DB2 Installation on OS/2 or Windows 3.x Operating Systems	205
Installing DB2 Products from a Hard Disk or CD-ROM	205
Step 1. Make DB2 Files Available for Installation	206
Step 2. Create a Response File for Unattended Installation	207
Step 3. Run the CMD/BAT File from the Remote Workstation	209
Installing DB2 Products Remotely Using TME Netfinity Server	212
Step 1. Copy the DB2 Install Files to the Code Server	212
Step 2. Add the DB2 Product to the Software Library	213
Step 3. Configure the DB2 Product	214
Step 4. Install DB2 Products on the Remote Workstation	214
Chapter 17. Unattended DB2 Installation on Windows 32-bit Operating Systems	215
Installing DB2 Products from a Hard Disk or CD-ROM	215
Before You Begin	216
Step 1. Make DB2 Files Available for Installation	216
Step 2. Create a Response File for Unattended Installation	218
Step 3. Run Setup from the Workstation	222
Installing DB2 Remotely Using SMS	223
Before You Begin	224

Step 1. Make DB2 Files Available for Installation	225
Step 2. Create a Response File for Unattended Installation	226
Step 3. Import the DB2 Install Files into SMS	230
Step 4. Distribute the DB2 Installation Package Using SMS	230
Step 5. Run the Installation on the SMS Client	231
Chapter 18. Installing Thin Clients on Windows 32-bit Operating Systems	233
Installing Thin Clients using the Code Server	234
Before You Begin	234
Thin Client Security Considerations	235
Step 1. Install DB2 Client Application Enabler on the Code Server	236
Step 2. Create Windows 95 or Windows 98 Code Base on Windows NT Code Server	237
Step 3. Create a Response File for Thin Client Installation	238
Step 4. Make Thin Client Setup Program Accessible to the Target Workstation	241
Step 5. Install Thin Client on the Target Workstation	241
Installing Thin Clients using SMS	243
Before You Begin	243
Step 1. Install DB2 Client Application Enabler on the Code Server	244
Step 2. Create a Response File for Thin Client Installation	245
Step 3. Import the Thin Client PDF File into SMS	247
Step 4. Distribute the Thin Client Installation Package Using SMS	248
Step 5. Run the Installation on the SMS Clients	249
Chapter 19. Using the Client Configuration Assistant	251
Configuring Database Connections	251
Using an Access Profile	252
Searching the Network for Databases	255
Manually Configure a Connection to a DB2 Database	260

Part 5. Appendixes 263

Appendix A. How the DB2 Library Is Structured	265
SmartGuides	265
Online Help	266
DB2 Books	268
Viewing Online Books	272
Searching Online Books	273
Printing the PostScript Books	273
Ordering the Printed DB2 Books	274
Information Center	275
Appendix B. National Language Support (NLS)	277
Language and Codeset Support for UNIX Operating Systems	277
Code Page and Language Support for OS/2 and Windows Operating Environments	278
Appendix C. Naming Rules	279

General Naming Rules	279
Database, Database Alias, and Catalog Node Names	280
Object Names	280
Usernames, Group Names, and Instance Names	281
Workstation Names (nname)	282
DB2SYSTEM Naming Rules	282
Password Rules	282
Appendix D. Notices	283
Trademarks	284
Trademarks of Other Companies	284
Appendix E. Contacting IBM	287
Index	291

Welcome to DB2 Universal Database

This book describes how to install and use the basic functions of DB2 Universal Database and configure DB2 Universal Database clients.

Part 1 provides disk and software requirements, and describes how to install a DB2 Client Application Enabler or a DB2 Software Developer's Kit on any supported operating system.

Part 2 provides information on configuring communications between DB2 clients and DB2 Universal Database servers and gateways.

Part 3 provides general information on using the DB2 Client Application Enabler.

Part 4 describes how to perform advanced installation and configuration tasks: such as performing a thin client installation of DB2 on Windows 32-bit operating systems; performing an unattended installation; performing administration tasks using the Client Configuration Assistant; and using the command line processor for setting up communications.

Part 5 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 book titles and to emphasize words.

`Monospace` indicates file names, directory paths, and examples of text you enter exactly as shown.



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. It provides 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...
Confirm that your system meets the DB2 disk and software prerequisites,	"Disk Requirements" on page 3 and "Software Requirements" on page 5.
Install the client products,	Chapter 2, "Installing DB2 Clients" on page 11.
Install DB2 Thin Clients on Windows 32-bit operating systems,	Chapter 18, "Installing Thin Clients on Windows 32-bit Operating Systems" on page 233.
Remotely install DB2 clients on OS/2, Windows 3.x and Windows 32-bit operating systems,	Chapter 16, "Unattended DB2 Installation on OS/2 or Windows 3.x Operating Systems" on page 205 or Chapter 17, "Unattended DB2 Installation on Windows 32-bit Operating Systems" on page 215
Set up communications on the client to access databases,	Chapter 9, "Configuring Client-to-Server Communications Using the Client Configuration Assistant" on page 49 or Chapter 10, "Configuring Client-to-Server Communications Using the Command Line Processor" on page 55.
Use the DB2 components,	Chapter 11, "Getting Started with DB2 Universal Database" on page 119.
Connect to a database,	"Connecting to a Database" on page 120.
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 133.
Use the DB2 Command Center or the DB2 command line processor to enter SQL statements or DB2 commands interactively,	Chapter 14, "Entering DB2 Commands and SQL Statements" on page 151.
Learn about last-minute changes to the product,	The Installation Notes on the CD-ROM or the Release Notes that are installed with a DB2 product.

To locate information on other topics, see Appendix A, “How the DB2 Library Is Structured” on page 265 for a complete description of the DB2 library.

Part 1. Installing DB2 Clients

Chapter 1. Planning for Installation

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.

Disk Requirements

This section shows the *minimum* amount of disk space that is required to install DB2 products and components. 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. 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 that you want to install. Include an allowance for your application data.

Client Components

Use Table 1 to estimate the amount of disk space you need on each of your client workstations.

Table 1 (Page 1 of 3). Estimating Disk Requirements for Client Components

	Recommended Minimum Disk (MB)
DB2 Client Application Enabler for AIX	
DB2 Client Application Enabler for AIX	22 MB
Online documentation in HTML format	25 MB
DB2 Client Application Enabler for HP-UX	
DB2 Client Application Enabler for HP-UX	18 MB
Online documentation in HTML format	25 MB

Table 1 (Page 2 of 3). Estimating Disk Requirements for Client Components

	Recommended Minimum Disk (MB)
DB2 Client Application Enabler for Macintosh	
DB2 Client Application Enabler for Macintosh	8 MB
Online documentation in HTML format	7 MB
DB2 Client Application Enabler for OS/2	
DB2 Client Application Enabler for OS/2 — this amount includes the following components:	70 MB
Client Configuration Assistant	2 MB
Graphical Tools	10 MB
Control Center	24 MB
Performance Monitor	8 MB
Visual Explain	4 MB
Online documentation in HTML format	25 MB
DB2 Client Application Enabler for SCO OpenServer Version 2	
DB2 Client Application Enabler for SCO OpenServer Version 2	16 MB
Online documentation in HTML format	25 MB
DB2 Client Application Enabler for SCO UnixWare 7	
DB2 Client Application Enabler for SCO UnixWare 7	16 MB
Online documentation in HTML format	25 MB
DB2 Client Application Enabler for Silicon Graphics IRIX	
DB2 Client Application Enabler for Silicon Graphics IRIX	14 MB
Online documentation in HTML format	19 MB
DB2 Client Application Enabler for SINIX Version 2	
DB2 Client Application Enabler for SINIX	16 MB
Online documentation in HTML format	21 MB
DB2 Client Application Enabler for Solaris	
DB2 Client Application Enabler for Solaris	16 MB
Online documentation in HTML format	25 MB
DB2 Client Application Enabler for Windows 3.x	
DB2 Client Application Enabler for Windows 3.x	10 MB
Online documentation in HTML format	10 MB
DB2 Client Application Enablers for Windows 32-bit Operating Systems	
DB2 Client Application Enablers for Windows 32-bit operating systems — this amount includes the following components:	94 MB
Client Configuration Assistant	2 MB
Control Center	18 MB
Performance Monitor	6 MB
Visual Explain	3 MB

Table 1 (Page 3 of 3). Estimating Disk Requirements for Client Components

	Recommended Minimum Disk (MB)
Online documentation in HTML format	30 MB

Software Requirements

This section outlines the software required to run DB2 Universal Database and its associated products or components. These associated products include the DB2 Client Application Enabler and DB2 Software Developer's Kit.

Client Product Requirements

Table 2 lists the software requirements needed for the DB2 client products.

Table 2 (Page 1 of 5). Software Requirements for Clients

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

Table 2 (Page 2 of 5). Software Requirements for Clients

Component	Operating System	Communications
DB2 Client Application Enabler for HP-UX Version 10.10 or later DB2 Software Developer's Kit for HP-UX	HP 9000 Series 700 or 800 system and the following: HP-UX Version 10.10 or later The following patches are required: For HP-UX Version 10.20 – PHSS_10556 – PHSS_10436 – PHSS_10053 – PHSS_10113 For HP-UX Version 10.10 – PHSS_10437 – PHSS_10053 – PHSS_10113 For systems with the ANSI C or C++ compilers: – PHSS_10261 – PHSS_7505 – PHSS_9096 for C++	APPC or TCP/IP For APPC connectivity, HP-UX Version 10.10 and 10.20 require one of the following: – SNAplus and the following components: – SNAplusLink Version A.10.10 – SNAplusAPI Version A.10.10 OR – SNAplus2 and the following components: – SNAplus2Link Version A.10.10 – SNAplus2API Version A.10.10 The HP-UX base operating system provides TCP/IP connectivity. Notes: 1. DB2 for HP-UX has a server IPX communications manager 2. If you want to use DCE (Distributed Computing Environment), you require a DCE product that is provided by the HP-UX Version 10 or later base operating system.
DB2 Client Application Enabler for HP-UX Version 11 DB2 Software Developer's Kit for HP-UX	HP 9000 Series 700 or 800 system and the following: HP-UX Version 11.00 or later	APPC or TCP/IP For APPC connectivity, you require either of the following: – SNAplus2 Link R6.11.00.00 – SNAplus2 API R6.11.00.00 The HP-UX base operating system provides TCP/IP connectivity. Notes: 1. DB2 for HP-UX has a server IPX communications manager 2. If you want to use DCE (Distributed Computing Environment), you require a DCE product that is provided by the HP-UX Version 11 or later base operating system.
DB2 Client Application Enabler for Macintosh DB2 Software Developer's Kit for Macintosh	System 7.5 or later	TCP/IP For TCP/IP connectivity, you require MacTCP Version 2.0.6. Note: If you are running in a 68K environment, you need to install Apple Shared Library Manager (ASLM) Version 2.0 or later.

Table 2 (Page 3 of 5). 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:		
<ol style="list-style-type: none"> OS/2 Warp Server provides the Netfinity LAN client that Netfinity LAN needs for installing clients and servers. Net.Data requires OS/2 Warp Version 3 or later and a Web server such as IBM's Internet Connection Server. 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 SCO OpenServer DB2 Software Developer's Kit for SCO OpenServer Net.Data for SCO OpenServer	SCO OpenServer Version 5.0.2 or later	APPC, IPX/SPX, or TCP/IP For APPC connectivity, you require Interface Systems CLEO SNNA Version 4.1.1. The SCO OpenServer base operating system provides IPX/SPX and TCP/IP connectivity.
DB2 Client Application Enabler for SCO UnixWare 7 DB2 Software Developer's Kit for SCO UnixWare 7 Net.Data for SCO UnixWare 7	SCO UnixWare 7	APPC, IPX/SPX, or TCP/IP For APPC connectivity, you require: – IBM eNetwork Communications Server Version 5 The SCO UnixWare 7 base operating system provides IPX/SPX and TCP/IP connectivity.
DB2 Client Application Enabler for Silicon Graphics IRIX DB2 Software Developer's Kit for Silicon Graphics IRIX	Silicon Graphics IRIX, Version 6.x, and the following filesets: – eoe.sw.oampkg – eoe.sw.svr4net	TCP/IP Note: The Silicon Graphics IRIX base operating system provides TCP/IP connectivity.
DB2 Client Application Enabler for SINIX DB2 Software Developer's Kit for SINIX	Siemens Nixdorf RM200 or RM400 or RM600 system and the following: SINIX-N or SINIX-Y Version 5.42 or later, Reliant UNIX	APPC or TCP/IP For APPC connectivity, you require the following: – SINIX TRANSIT-SERVER V3.3, SNA Communication Server – SINIX TRANSIT-CLIENT V3.3 SNA Communication Server / Local Function – SINIX TRANSIT-CPIC V3.3 SNA LU6.2 Communication and CPI-C. The SINIX base operating system provides TCP/IP connectivity.

Table 2 (Page 4 of 5). Software Requirements for Clients

Component	Operating System	Communications
DB2 Client Application Enabler for Solaris DB2 Software Developer's Kit for Solaris Net.Data for Solaris	Solaris SPARC-based computer and the following: Solaris Version 2.5.1 or later The following patches are required with Version 2.5.1: 101242 Rev.11 or higher 103566 Rev.08 or higher 103600 Rev.13 or higher 103640 Rev.20 or higher	APPC or TCP/IP For APPC connectivity, you require SunLink SNA 9.0 and the following: – SunLink P2P LU6.2 9.0 – SunLink PU2.1 9.0 – SunLink P2P CPI–C 9.0 The Solaris base operating system provides TCP/IP connectivity. Notes: 1. If you want to use DCE (Distributed Computing Environment), you require a DCE product with Transarc DCE Version 1.1 for Solaris 2.5 and 2.5.1, patch level 18 or higher. 2. DB2 for Solaris also has a server IPX communications manager.
DB2 Client Application Enabler for Windows 3.x DB2 Software Developer's Kit for Windows 3.x	Microsoft Windows 3.x	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 3.x using WIN-OS/2 DB2 Software Developer's Kit for Windows 3.x using WIN-OS/2	OS/2 Warp Version 3 or Version 4 OS/2 Warp Connect Version 3 OS/2 Warp Server Version 4 OS/2 Warp Server Advanced V4 OS/2 Warp Server Advanced V4 with SMP Feature	NetBIOS, IPX/SPX, TCP/IP, or Named Pipes (Local) For NetBIOS connectivity, you require IBM NTS/2 Version 1.0, IBM Communications Manager Version 1.1, or IBM OS/2 LAN Requester. For IPX/SPX connectivity, you require Novell NetWare client for OS/2 Version 2.10 or later with virtual session support. For TCP/IP connectivity, you require IBM TCP/IP for OS/2 with the Windows Access kit.
DB2 Client Application Enabler for Windows 95 DB2 Software Developer's Kit for Windows 95	Windows 95 4.00.950 or later	IPX/SPX, Named Pipes, NetBIOS, or TCP/IP The Windows 95 base operating system provides NetBIOS, IPX/SPX, TCP/IP, and Named Pipes connectivity. Note: IPX/SPX connectivity is supported between Windows 95 clients and Windows NT servers. There is no support for IPX/SPX connectivity between Windows 95 clients and OS/2 and UNIX servers.

Table 2 (Page 5 of 5). Software Requirements for Clients

Component	Operating System	Communications
DB2 Client Application Enabler for Windows 98 DB2 Software Developer's Kit for Windows 98	Windows 98	IPX/SPX, Named Pipes, NetBIOS, or TCP/IP The Windows 98 base operating system provides NetBIOS, IPX/SPX, TCP/IP, and Named Pipes connectivity. Note: IPX/SPX connectivity is supported between Windows 98 clients and Windows NT servers. There is no support for IPX/SPX connectivity between Windows 98 clients and OS/2 and UNIX servers.
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: <ul style="list-style-type: none"> - IBM Communications Server for Windows NT Version 5.01. - 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-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 3 (Page 1 of 2). Possible Client-to-Server Connectivity Scenarios

	DB2 Servers							
	AIX	HP-UX	OS/2	SCO UnixWare 7	Solaris	Windows NT	SCO OpenServer	SINIX
AIX	APPC TCP/IP	TCP/IP	APPC TCP/IP	TCP/IP	APPC TCP/IP	APPC TCP/IP	APPC TCP/IP	TCP/IP
HP-UX	APPC TCP/IP	TCP/IP	APPC TCP/IP	TCP/IP	APPC TCP/IP	APPC TCP/IP	APPC TCP/IP	TCP/IP
Macintosh	TCP/IP	TCP/IP	TCP/IP	TCP/IP	TCP/IP	TCP/IP	TCP/IP	TCP/IP
OS/2	APPC IPX/SPX(1),(2) TCP/IP	IPX/SPX(1),(2) TCP/IP	APPC IPX/SPX(1),(2) NetBIOS TCP/IP	IPX/SPX(1), TCP/IP	APPC IPX/SPX(1) TCP/IP	APPC IPX/SPX(1) NetBIOS TCP/IP	APPC IPX/SPX(1) TCP/IP	TCP/IP
SCO OpenServer	APPC IPX/SPX(1) TCP/IP	IPX/SPX(1) TCP/IP	APPC IPX/SPX(1) TCP/IP	IPX/SPX(1) TCP/IP	APPC IPX/SPX(1) TCP/IP	APPC IPX/SPX(1) TCP/IP	APPC IPX/SPX(1) TCP/IP	TCP/IP
SCO UnixWare 7	APPC IPX/SPX(1) TCP/IP	IPX/SPX(1) TCP/IP	APPC IPX/SPX(1) TCP/IP	IPX/SPX(1) TCP/IP	APPC IPX/SPX(1) TCP/IP	APPC IPX/SPX(1) TCP/IP	APPC IPX/SPX(1) TCP/IP	TCP/IP

Table 3 (Page 2 of 2). Possible Client-to-Server Connectivity Scenarios

	DB2 Servers							
	AIX	HP-UX	OS/2	SCO UnixWare 7	Solaris	Windows NT	SCO OpenServer	SINIX
Silicon Graphics IRIX	TCP/IP	TCP/IP	TCP/IP	TCP/IP	TCP/IP	TCP/IP	TCP/IP	TCP/IP
SINIX	APPC TCP/IP	TCP/IP	APPC TCP/IP	TCP/IP	APPC TCP/IP	APPC TCP/IP	APPC TCP/IP	TCP/IP
Solaris	APPC TCP/IP	TCP/IP	APPC TCP/IP	TCP/IP	APPC TCP/IP	APPC TCP/IP	APPC TCP/IP	TCP/IP
Windows 3.x	IPX/SPX(1),(2) TCP/IP	IPX/SPX(1),(2) TCP/IP	IPX/SPX(1),(2) NetBIOS TCP/IP Local Named Pipes	IPX/SPX(1) TCP/IP	IPX/SPX(1) TCP/IP	IPX/SPX(1) NetBIOS TCP/IP	IPX/SPX(1) TCP/IP	TCP/IP
Windows 95 or Windows 98	IPX/SPX(1) TCP/IP	IPX/SPX(1) TCP/IP	IPX/SPX(1) NetBIOS TCP/IP	IPX/SPX(1) TCP/IP	IPX/SPX(1) TCP/IP	IPX/SPX(1) Named Pipes NetBIOS TCP/IP	IPX/SPX(1) TCP/IP	TCP/IP
Windows NT	APPC IPX/SPX(1) TCP/IP	IPX/SPX(1) TCP/IP	APPC IPX/SPX(1) NetBIOS TCP/IP	IPX/SPX(1) TCP/IP	APPC IPX/SPX(1) TCP/IP	APPC IPX/SPX(1) Named Pipes NetBIOS TCP/IP	APPC IPX/SPX(1) TCP/IP	TCP/IP
1 Direct Addressing 2 File Server Addressing Note: SCO OpenServer and SINIX are DB2 Version 2 only.								

Chapter 2. Installing DB2 Clients

Use the instructions in this section to install a DB2 client. You can also run unattended installations of DB2 to remote target machines running OS/2, Windows 3.x, and Windows 32-bit operating systems. For unattended installations of DB2, see 289.

A DB2 Client Application Enabler is installed with any DB2 Version 5 product. If a DB2 Version 5 product is already installed, there is no need to install a DB2 Client Application Enabler. To configure your client to access remote servers, see Chapter 9, "Configuring Client-to-Server Communications Using the Client Configuration Assistant" on page 49, or for UNIX users, see Chapter 10, "Configuring Client-to-Server Communications Using the Command Line Processor" on page 55.

Clients on a LAN that will connect to a DB2 Universal Database server must have the appropriate DB2 Client Application Enabler or a DB2 Software Developer's Kit installed.

DB2 Client Application Enablers and DB2 Software Developer's Kits are available for the following: AIX, DOS, HP-UX, Macintosh, OS/2, SCO UnixWare 7, SCO OpenServer, Silicon Graphics IRIX, SINIX, Solaris, Windows 3.x, and Windows 32-bit operating systems.

You may install DB2 Client Application Enablers on any number of workstations; licensing restrictions are controlled at the server.



Go to the section that gives instructions for the DB2 client that you want to install.

Chapter 3, "Installing DB2 Clients on OS/2 Operating Systems" on page 13.

Chapter 4, "Installing DB2 Clients on Windows 32-Bit Operating Systems" on page 17.

Chapter 5, "Installing DB2 Clients on Windows 3.x Operating Systems" on page 21.

Chapter 6, "Installing DB2 Clients on UNIX Operating Systems" on page 25.

Chapter 8, "Installing DB2 Clients on Macintosh Operating Systems" on page 45.

Chapter 16, "Unattended DB2 Installation on OS/2 or Windows 3.x Operating Systems" on page 205.

Chapter 17, "Unattended DB2 Installation on Windows 32-bit Operating Systems" on page 215.

Chapter 18, "Installing Thin Clients on Windows 32-bit Operating Systems" on page 233

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 Kits are on the DB2 Software Developer's Kit CD-ROM that comes with the Application Developer's Kit. The DB2 Client Application Enablers (except the DOS client) are on the DB2 Client Application Enablers CD-ROM.

Chapter 3. Installing DB2 Clients on OS/2 Operating Systems

This section contains the information that you need to install the DB2 Client Application Enabler or the DB2 Software Developer's Kit on OS/2 operating systems.

To install the DB2 Client Application Enabler or DB2 Software Developer's Kit for OS/2:

- 1 Insert the appropriate CD-ROM into the drive:

To install the DB2 Client Application Enabler, use the DB2 Client Application Enablers CD-ROM containing the OS/2 client.

To install the DB2 Software Developer's Kit, use the DB2 Software Developer's Kit CD-ROM.

- 2 Begin the installation program as follows:

- a Open an OS/2 window and set the drive to *x*:, where *x* 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 23 on page 278 lists the codes for each available language.
- c Enter the **install** command.

The IBM DB2 for OS/2 Version 5 Installation window opens.



Select the **IBM DB2 Client Application Enabler** check box, and click on the **Continue** push button.

Invoke the online help by clicking on the **HELP** push button at any time. The Install 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. xx`. This is the recommended method.

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.

The Install - directories window opens.

- 4 The Install - directories window provides a list of 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 the components you want to install.
- b In the **File directory** field, type the directory where you want the product installed. The default directory is `c:\sqllib`.
Note: If a DB2 Version 5 product is already installed on this 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 begin 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.

- 5** At the end of the installation, click on the **Exit** push button to exit the installation program. You must shut down and reboot before you can use the client.

Note: If you want to run Windows 3.x applications on your OS/2 client, you must also install the DB2 Client Application Enabler for Windows 3.x on your system. You should install this Client Application Enabler 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 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 9, "Configuring Client-to-Server Communications Using the Client Configuration Assistant" on page 49.

Chapter 4. Installing DB2 Clients on Windows 32-Bit Operating Systems

This section contains the information you need to install the DB2 Client Application Enabler or the DB2 Software Developer's Kit on Windows 32-bit operating systems.

To install a DB2 Client Application Enabler or DB2 Software Developer's Kit:

- 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 appropriate CD-ROM into the drive:

To install the DB2 Client Application Enabler, use the DB2 Client Application Enablers CD-ROM .

To install the DB2 Software Developer's Kit, use the Application Developer's Kit CD-ROM.



To manually invoke the setup program, do the following:

- 1 Click on **Start** and select the **Run** option.

- 2 Type the following in the **Open** field:

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

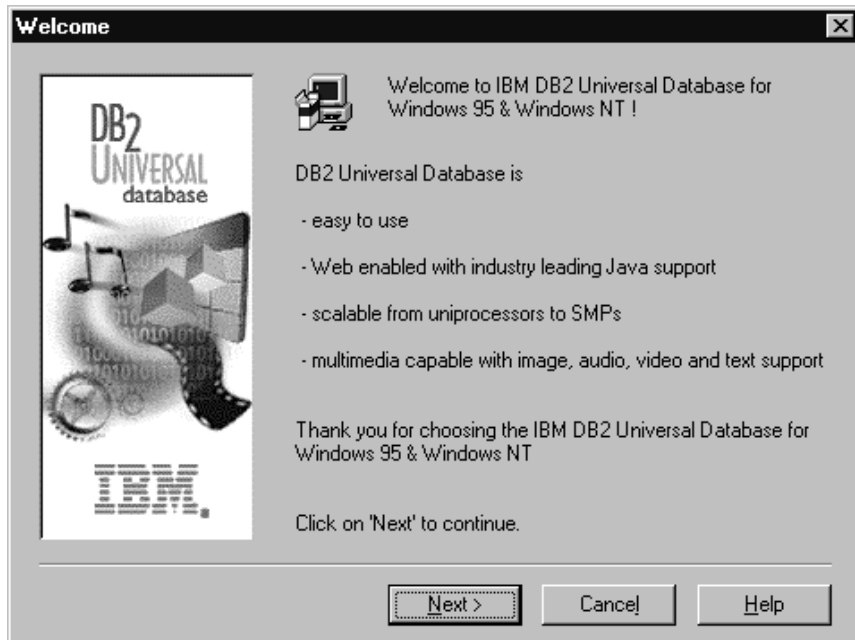
where:

x: represents your CD-ROM drive

language represents the two-character country code for your language (for example, EN for English). Table 23 on page 278 lists the code for each available language.

- 3 Click on **OK**.

- 4 The Welcome window opens.



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. 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. A DB2 instance is created.



Compact Install: Installs only the required DB2 components and ODBC support. A DB2 instance is created.



Custom Install: Installs only those components that you select. A DB2 instance is created.

- 7 Respond to the setup program's prompts. Online help is available to guide 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 before you can begin to use it. Select a reboot option and click on the **Finish** push button. This completes the installation.

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 9, "Configuring Client-to-Server Communications Using the Client Configuration Assistant" on page 49.

Chapter 5. Installing DB2 Clients on Windows 3.x Operating Systems

This section contains the information you need to install the DB2 Client Application Enabler or the DB2 Software Developer's Kit on Windows 3.x operating systems.



If you are planning to support only local communications from a WIN-OS/2 session to an OS/2 server, it is recommended that you use the local Windows Support supplied with the DB2 for OS/2 server, and do not install the Windows 3.x Client Application Enabler. Refer to the *Quick Beginnings for OS/2* manual to install Windows support on your OS/2 server.

Note: 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 or the DB2 Software Developer's Kit on Windows 3.x:

- 1 Insert the appropriate CD-ROM into the drive:

To install the DB2 Client Application Enabler, use the DB2 Client Application Enablers CD-ROM containing the Windows 3.x client.

To install the DB2 Software Developer's Kit, use the Software Developer's Kit CD-ROM.

- 2 To begin the installation program enter:

```
x:\db2\windows\language\install\install.exe
```

where *x*: represents your CD-ROM drive and where *language* is the two-character country code that represents your language (for example, EN for English). Table 23 on page 278 lists the codes for each available language.

The Install window opens.



- 3 Indicate whether or not you want to update the `autoexec.bat` file as part of the installation.

Note: The `config.sys` file will not be modified by the installation.

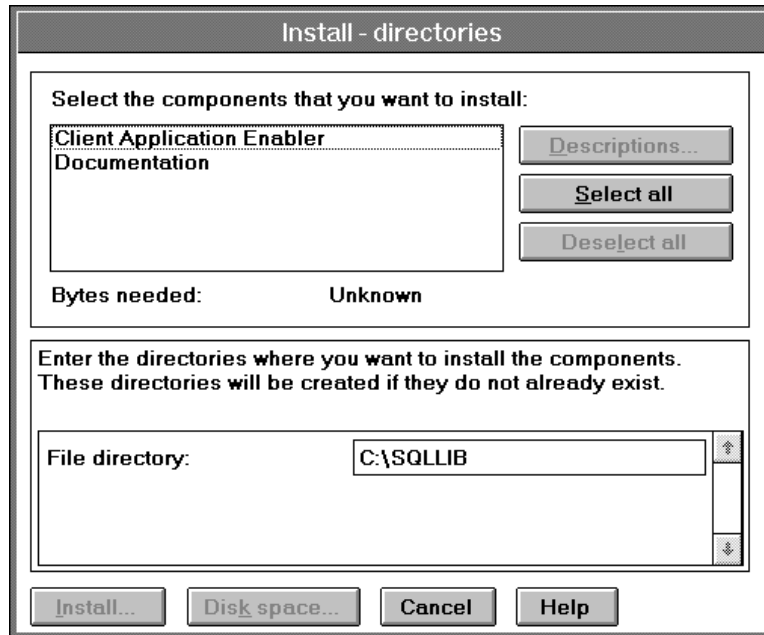
If you want the `autoexec.bat` file updated, check that the **Update CONFIG.SYS/AUTOEXEC.BAT** check box is selected, then click on **OK**. A backup copy, called `autoexec.bak`, is created in the same directory as your `autoexec.bat` file. This is the recommended method.

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.



- 5 Select the components that you want to install.
- 6 In the **File directory** field, type the directory where you want the product installed. The default directory 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 Version 5 DB2 Client Application Enabler for OS/2 installed, enter the same directory in the **File directory** field as the DB2 Client Application Enabler for OS/2. 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.

- 7 Click on the **Disk space** push button to see how much space you have on the selected drive.
- 8 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 exit the installation program.
- 9 Click on the **Exit** push button to end the installation.
- 10 Once you have completed the installation, restart your WIN-OS/2 session before using the DB2 for Windows 3.x Client Application Enabler.

Chapter 6. Installing DB2 Clients on UNIX Operating Systems

This section contains the information that you need to install the DB2 Client Application Enabler or the DB2 Software Developer's Kit on UNIX operating systems. See the following sections as appropriate:

“Installing DB2 Version 5 Clients” on page 26.

“Installing DB2 Version 2 Clients” on page 35.



If you want to remove the DB2 Client Application Enabler, refer to the *Quick Beginnings* manual.

Before You Begin

Before you begin installing DB2 products using the DB2 Installer program, you need to gather the following information:

Where is the CD-ROM mount point?

You need to mount the CD-ROM before you can install the DB2 product. To mount the CD-ROM, you must know where the mount point is. For example, the CD-ROM mount point could be `/cdrom`.

Note: On the Solaris operating system, the CD-ROM is automatically mounted if the Volume Manager (vold) is running.

How do I assign a group name and user name for a DB2 instance?

A separate user name must be used for each DB2 Instance. We recommend that a new group be created which should be used as a primary group for the DB2 instance user.

Installing DB2 Version 5 Clients

To install the DB2 Client Application Enabler or the DB2 Software Developer's Kit on AIX, HP-UX, SCO UnixWare 7, Silicon Graphics IRIX, and Solaris systems, perform the following steps:

Step 1. Update Kernel Configuration Parameters

To run DB2 Client Application Enabler or the DB2 Software Developer's Kit for UNIX systems, you may have to update some kernel configuration parameters.



This step is not required on AIX. If you are installing an AIX client, skip this step and go to "Step 2. Mount the DB2 Client Application Enablers CD-ROM" on page 28.

Recommended Values for HP-UX Version 10 and Version 11

Table 4 lists the recommended values for the specified HP-UX kernel configuration parameters. These values are valid for HP-UX 10 and HP-UX 11.

Kernel Parameter	Recommended Value
msgseg	8192
msgmnb	65535 (1)
msgmax	65535 (1)
msgssz	16

Notes:

1. Parameters `msgmnb` and `msgmax` must be set to 65535.
2. To maintain the interdependency among kernel parameters, change parameters in the same sequence in which they appear in Table 4.

To change a value:

- 1 Enter the **sam** command to start the System Administration Manager (SAM) program.
- 2 Double-click on the **Kernel Configuration** icon.
- 3 Double-click on the **Configurable Parameters** icon.

- 4 Double-click on the parameter that you want to change and enter the new value in the **Formula/Value** field.
- 5 Click on **OK**.

The HP-UX operating system automatically reboots after you change the values for the kernel configuration parameters.



Go to "Step 2. Mount the DB2 Client Application Enablers CD-ROM" on page 28 to continue with the installation.

Recommended Values for SCO UnixWare 7

Table 5 lists the recommended values for the specified SCO UnixWare 7 kernel configuration parameters.

<i>Table 5. SCO UnixWare 7 Kernel Configuration Parameters (Recommended Values)</i>	
Kernel Parameter	Recommended Value
msgmax	65535 (1)
msgmnb	65535 (1)
msgssz	524288
msgmni	256

Notes:

1. Parameters *msgmax* and *msgmnb* should be set at least to 65535.
2. To maintain the interdependency among kernel parameters, change parameters in the same sequence in which they appear in the preceding table.

To change a value, do the following:

- 1 Enter the **scoadmin** command to start the System Administration tool.
- 2 Double-click on the **System** folder.
- 3 Double-click on the **System Tuner** icon.
- 4 Click on the drop down box and select the **Inter-Process Communication (IPC) Parameters**.
- 5 Select the parameter to be changed and enter the new value.
- 6 Click on **OK** when you have finished changing all the parameters.
- 7 Click on the **Yes** push button to rebuild the kernel.
- 8 Reboot the system so that the changes can take effect.



Go to “Step 2. Mount the DB2 Client Application Enablers CD-ROM” on page 28 to continue with the installation.

Recommended Values for Solaris

Table 6 lists the recommended values for the specified Solaris kernel configuration parameters.

Kernel Parameter	Recommended Value
<code>msgsys:msginfo_msgmax</code>	65535 (1)
<code>msgsys:msginfo_msgmnb</code>	65535 (1)
<code>msgsys:msginfo_msgseg</code>	8192
<code>msgsys:msginfo_msgssz</code>	16

Notes:

1. Parameters `msgsys:msginfo_msgmnb` and `msgsys:msginfo_msgmax` must be set to 65535.

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

```
set parameter_name = value
```

where *parameter_name* represents the parameter you want to change.

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

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

After changing the kernel parameters, reboot the system so that the changes can take effect.



Go to “Step 2. Mount the DB2 Client Application Enablers CD-ROM” to continue with the installation.

Step 2. Mount the DB2 Client Application Enablers CD-ROM

To install DB2 Client Application Enabler using the DB2 Installer, you must first mount the CD-ROM. Once you have mounted the CD-ROM, you can start installing the DB2 products.

Mounting on AIX Systems

Perform the following steps to mount the CD-ROM on AIX operating systems:

- 1 Log in as user with root authority.
- 2 Insert the DB2 Client Application Enablers CD-ROM in the drive.
- 3 Create a directory to mount the CD-ROM by entering the following command:

```
mkdir -p /cdrom
```

where `cdrom` is the CD-ROM mount directory.
- 4 Allocate a CD-ROM file system by entering the following command:

```
smitty storage
```
- 5 Select **File Systems**.
- 6 Select **Add / Change / Show / Delete File Systems**.
- 7 Select **CDROM File Systems**.
- 8 Select **Add CDROM File System**.
- 9 Select **Device Name**.



Device names for CD-ROM file systems must be unique. If there is a duplicate device name, you may need to delete a previously-defined CD-ROM file system or use another name for your directory.

- 10 In the pop-up window, enter the following as **mount point**:

```
/cdrom
```

- 11 Mount the CD-ROM file system by entering the following command:

```
smitt mountfs
```
- 12 Select the **FileSystem** name. For example, the name could be `/dev/cd .`
- 13 Select the **Directory** name, `/cdrom`.
- 14 Select the **Type of filesystem**, `cdrfs`.
- 15 Set the **Mount as READ-ONLY system** to **Yes**.
- 16 Log off.



Go to "Step 3. Perform the Installation" on page 32 to continue with the installation.

Mounting on HP-UX Systems

Perform the following steps to mount the CD-ROM on HP-UX operating systems:

- 1 Log in as user with root authority.
- 2 Insert the CD-ROM in the drive and mount it as in the following example:

```
mkdir /cdrom  
/usr/sbin/mount /dev/dsk/c t2d /cdrom
```

where `/cdrom` is the CD-ROM mount directory.

- 3 Log out.



The CD-ROM may also be mounted using the System Administration (**SAM**) tool. Consult your HP-UX documentation for more information about **SAM**.



Go to "Step 3. Perform the Installation" on page 32 to continue with the installation.

Mounting on SCO UnixWare 7

Perform the following steps to mount the CD-ROM on SCO UnixWare 7 operating systems:

- 1 Log in as user with root authority.
- 2 Insert the DB2 Client Application Enablers CD-ROM into the drive and mount it. For example, to mount the CD-ROM as `/cdrom`, type:

```
mount -F cdfs -o ro /dev/cdrom/c b t51 /cdrom
```

where `c b t51` is the device name found under `/dev/cdrom` directory and varies with the particular hardware on the system.

- 3 Log out.



Go to "Step 3. Perform the Installation" on page 32 to continue with the installation.

Mounting on Silicon Graphics IRIX

Perform the following steps to mount the CD-ROM on Silicon Graphics IRIX operating systems:

- 1 Log in as user with root authority.
- 2 Insert the DB2 Client Application Enablers CD-ROM into the drive and mount it using the **mount** command as follows:

```
mount -t iso9666 device mount_point
```

For example, to mount the CD-ROM as `/cdrom`, enter the following commands:

```
mkdir /cdrom
mount -t iso9666 /dev/scsi/sc d71 /cdrom
```

- 3 Log out.



Go to "Step 3. Perform the Installation" on page 32 to continue with the installation.

Mounting on Solaris Systems

Perform the following steps to mount the CD-ROM on Solaris operating systems:

- 1 Log in as user with root authority.
- 2 If the Volume Manager is not running on your system, enter the following commands to mount the CD-ROM:

```
mkdir -p /cdrom/unnamed_cdrom
mount -F hsfs -o ro /dev/dsk/c t6d s2 /cdrom/unnamed_cdrom
```

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



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

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

```
/cdrom/unnamed_cdrom
```

- 3 Log out.



Go to “Step 3. Perform the Installation” on page 32 to continue with the installation.

Step 3. Perform the Installation

After you mount the CD-ROM file system, use the DB2 Installer program to install the DB2 products.



If you are installing the DB2 Client Application Enabler from a remote server, it is better to use the **telnet** command to open a telnet session instead of using the **rlogin** command to connect to your remote server.

- 1 Log in as user with root authority.
- 2 Insert the DB2 Client Application Enablers CD-ROM in the drive and if required, mount the CD-ROM. Refer to “Step 2. Mount the DB2 Client Application Enablers CD-ROM” on page 28 if you need to mount the CD-ROM.
- 3 Change to the directory where the DB2 install CD-ROM is mounted and run the following command:

```
./db2setup
```

- 4 DB2 install images are available in the following directories (assuming that the mount point for the CD-ROM is `/cdrom`):

On AIX:

```
cd /cdrom/db2/aix
```

On HP-UX:

```
cd /cdrom/db2/hpux1 (for HP-UX Version 1 )
```

```
cd /cdrom/db2/hpux11 (for HP-UX Version 11)
```

On SCO UnixWare 7:

```
cd /cdrom/db2/scouw
```

On SGI:

```
cd /cdrom/db2/sgi
```

On Solaris:

```
cd /cdrom/unnamed_cdrom/db2/solaris
```

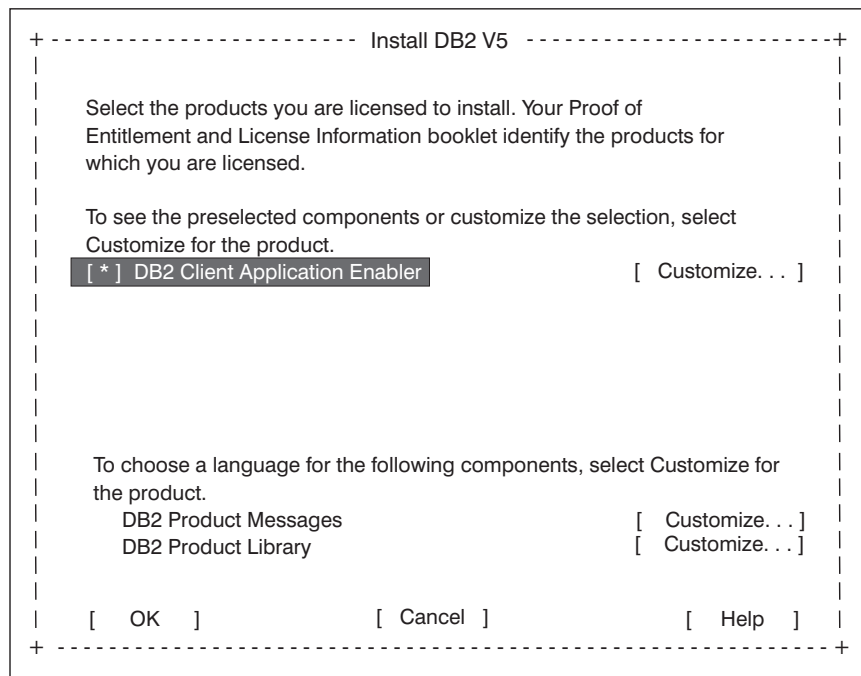
- 5 To install the client, enter the following commands:

```
cd /cdrom
./db2setup
```


The installation program will proceed to install the appropriate client on your workstation.



It will take some time for the DB2 Installer program to start up, as it is scanning your system for information.



- 6 From the product list on the Install DB2 V5 window, select the DB2 products that you want to install.

To display the required and optional components for a product that you want to install, select **Customize**. The optional components that are most typically used are pre-selected for you.



To refresh the current screen, press the F5 key or Ctrl+L.

When you have finished your selection, select **OK**. To undo any selections you made, select **Default**. The Create DB2 Services window opens.



Go to step 9 on page 34 if you do not want to create a DB2 Instance at this time.

- 7 At the Create DB2 Services window, select the **Create a DB2 Instance** option. The DB2 Instance window opens:

```

+-----+ Create DB2 Services -----+
|
| +-----+ DB2 Instance -----+
| |
| | Authentication:
| | Enter User ID, Group ID, Home Directory and Password that
| | will be used for the DB2 Instance.
| | User Name      [ db2inst1 ]
| | User ID       :           : [*] Use default UID
| | Group Name    [db2iadm1]
| | Group ID     :           : [*] Use default GID
| | Home Directory [ /home/db2inst1 ]
| | Password     [ ***** ]
| | Verify Password [ ***** ]
| |
| | Select Properties to view or change      : Properties :
| | more options.
| | Select Default to restore all            [ Default ]
| | default settings.
| |
| | [ OK ]           [ Cancel ]           [ Help ]
| |
| +-----+

```

You can use the default values displayed at the DB2 Instance window or change them. To change the default **User ID** for the instance, deselect **Use default UID** and enter a new value for the **User ID**.

- 8 Complete the fields and select **OK**.
- 9 Select **OK** and the Summary Report window opens. The report lists the installable items as well as the location of the installation log file.
- 10 Select the Summary Report field and use the Up or Down arrow keys to review the Summary Report. If the selections are not correct, press the **ESC** or **F3** key, to return to the previous screen. Otherwise select **Continue** to start the installation.
- 11 When the installation is complete, use the Up or Down arrow keys to review the Status Report. Select **View Log** to view the installation log file. Select **OK** to close the Status Report screen.
- 12 Select **Close** from the DB2 Installer screen to terminate the DB2 Installer program.

After the installation is complete, the software is installed in the `DB2DIR` directory,

where DB2DIR = /usr/lpp/db2_05_00 on AIX
= /opt/IBMd2/V5.0 on HP-UX, SCO UnixWare
7, SGI IRIX, or Solaris



Proceed to Chapter 10, "Configuring Client-to-Server Communications Using the Command Line Processor" on page 55.

Installing DB2 Version 2 Clients

This section details installing DB2 products on SINIX and SCO OpenServer Clients.

Installing DB2 Version 2 Clients on SINIX

To install the DB2 Client Application Enabler on SINIX, perform the following steps:

- 1 Log in as user with root authority.
- 2 Insert the DB2 Client Application Enablers CD-ROM into the drive and mount it using either the **sysadm** or the **mount** command as follows:

```
mkdir /cdrom  
mount -F hs /dev/ios /sdisk 6s /cdrom
```

where /cdrom is the CD-ROM mount directory.

The install images for DB2 for SINIX products are available in the /cdrom/db2/sinix/IBMd2 directory.

- 3 Install the packages as follows:

For the DB2 Client Application Enabler, type the following command:

```
pkgadd -d /cdrom/db2/sinix/IBMd2 db2cliv21 db2msgDe db2conv
```

For the DB2 Software Developer's Kit for SINIX, type the following command:

```
pkgadd -d /cdrom/db2/sinix/IBMd2 db2cliv21 db2msgDe db2conv db2c  
pkgadd -d /cdrom/db2/sinix/IBMd2 db2sdkmisc db2skdc sdbsdkli db2
```

Some kernel configuration parameters must be changed to use DB2 Client Application Enabler for SINIX. Use the SINIX **idtune** command to update the values as given in Table 7 on page 36. After doing so, rebuild the SINIX kernel and reboot the system for the changes to take effect.

Table 7 on page 36 lists the recommended values for the specified SINIX kernel configuration parameters.

<i>Table 7. SINIX Kernel Configuration Parameters (Recommended Values)</i>	
SINIX RM600 Kernel Parameter	Recommended Value
msgmax	65535
msgmnb	65535
msgseg	8192
msgssz	16

Note: The kernel configuration parameters `msgmnb` and `msgmax` must be changed to 65535.

Installing DB2 Version 2 Clients on SCO OpenServer

To install the DB2 Client Application Enabler on SCO OpenServer, perform the following steps:

- 1** Log in as user with root authority.
- 2** Insert the DB2 Client Application Enablers CD-ROM into the drive and mount the CD-ROM by entering:

```
mkdir /cdrom
mount /dev/cd /cdrom
```

where `/cdrom` is the CD-ROM mount directory.

The install images for the DB2 products are available in the `/cdrom/db2/sco/IBMdb2` directory.

- 3** Enter the **custom** command. The Software Manager window opens.
- 4** From the Software Manager window, do the following:
 - a** From the **Software** pulldown menu, select **Install New**.
 - b** If you are doing a remote install, select the host from which you want to install. Otherwise, select **Continue**.
 - c** From the **Media Images** menu, select **Media Images** and select **Continue**.
 - d** In the **Image Directory** field, enter:

```
/cdrom/db2/sco/IBMdb2
```
 - e** Choose the software you want to install:

To install all components, select **Full**.

To install selected components, choose **Partial**. On the Software Selection window, highlight the products you want to install:

- DB2 Client Application Enabler for SCO OpenServer

- DB2 Software Developer's Kit for SCO OpenServer
- DB2 Product Messages for SCO OpenServer.

Press Enter.

f Update the kernel configuration parameters.

Change the values as given in Table 8 by updating the `/etc/conf/cf.d/stune` file. Then, rebuild the kernel and reboot the system for the changes to take effect.

Kernel Parameter	Recommended Value
<code>msgssz</code>	64
<code>semmap</code>	1
<code>semmi</code>	1
<code>semmns</code>	1
<code>semnu</code>	1

Note: The kernel configuration parameters `msgmnb` and `msgmax` must be changed to 65535.

Create or Assign Groups and Users

If you do not already have a user ID to own the instance and a system administration group ID that is the primary group of the instance owner, create these as follows:

- 1** Log in as user with root authority.
- 2** Create a system administration group ID using either the **sysadm** or **groupadd** command. For example, to create a group named `dbadmin1`, type:

```
groupadd dbadmin1
```

- 3** Create a user ID that will be the instance owner using either the **sysadm** or **useradd** command. For example, to create a user named `inst1` type:

```
useradd -g dbadmin1 -G dbadmin1 -d /home/inst1 inst1
passwd inst1
```



When choosing a name for a group or user, follow the rules described in Appendix C, "Naming Rules" on page 279.

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

Create an Instance of the Product

Use the **db2icrt** command to create an instance of the product. The **db2icrt** command is located in the `/opt/IBMDB2/V5.1/instance` directory.

On SCO OpenServer, go to the `/usr/IBMDB2/V2.1/instance` directory. On SINIX, go to the `/opt/IBMDB2/V2.1/instance` directory.

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

```
▶▶ db2icrt — InstName — [ -a AuthType ] — [ -u FencedID ] ▶▶
```

where:

instance_name Is the login name of the instance owner.

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

Note: The authentication type of the instance applies to all databases under the instance.

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

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

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

```
db2icrt db2inst1
```

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

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



To avoid a potential loss of data if an instance is deleted, you should not create user files or directories under the `INSTHOME/sqllib` directory, other than those created by DB2. The only exceptions is if your system supports fenced user defined functions and fenced stored procedures, put the fenced applications in the `INSTHOME/sqllib/function` directory.

Set Up the DB2 Operating Environment

Before starting DB2, you must execute a script to set up the DB2 operating environment and select an instance. The sample script files, `db2profile` (for Bourne or Korn shell) and `db2cshrc` (for C shell) are provided to help you set up the operating environment for a DB2 instance.

The instance owner may customize these scripts for all users of an instance. These scripts are available in the `INSTHOME/sqllib` directory, where `INSTHOME` is the home directory of the instance.

Determine the most appropriate way to execute the commands in the sample `db2profile` or `db2cshrc` script file in your environment and instruct your users to do it. For example you could add the following to the start-up script:

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

Create Links for DB2 Files

If you are developing or running applications, you may want to create links for DB2 libraries to avoid specifying the full path to the product libraries and the include files. To create these links, use the **db2ln** command.

On SCO OpenServer systems, type:

```
/usr/IBMDB2/V2.1/cfg/db2ln
```

On SINIX systems, type:

```
/opt/IBMDB2/V2.1/cfg/db2ln
```

This creates links from the DB2 libraries to `/usr/lib`, and from the DB2 include files to `/usr/include`.



To configure your client to access remote servers, go to Chapter 10, "Configuring Client-to-Server Communications Using the Command Line Processor" on page 55.

Chapter 7. Migrating UNIX Clients

Once you've installed DB2, you may want to use an existing instance from a pre-version 5 of DB2. To do so, you must migrate the instance.

For more information about migration paths and other specific migration issues please see the *Quick Beginnings* manual.

Migrating Instances

If you upgraded your installed DB2 product by applying a PTF or a FixPak, you should update instances, using the **db2iupdt** command, instead of migrating them. Instance updates do not apply to *DB2 Product Documentation* and *DB2 Product Messages*.

Before you can migrate an instance to use the latest version of DB2, you must install DB2 Version 5 on your system.



If there are several DB2 instances using previous versions of DB2, you do not need to migrate all of these instances at this time. Instances that are not migrated will continue to use previous versions of DB2.

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

- 1 Prepare the DB2 instance for migration.
- 2 Migrate the DB2 instance.

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

Prepare the DB2 Instance for Migration

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

- 1 Log in as the DB2 instance owner.
- 2 Stop all command line processor sessions by entering the **db2 terminate** command in each session that was running the command line processor.

3 Enter the **db2_kill** command to clean up any remaining DB2 resources.

4 Log off.

The DB2 instance is now ready for migration.

Migrate the DB2 Instance

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

1 Log in as user with root authority.



If the *library_path* environment variable is set to `/usr/lib` on AIX or `/opt/lib` on HP-UX, SCO UnixWare 7, or Solaris, and there is a link in `/usr/lib` or `/opt/lib` to the Version 5 `libdb2.a` DB2 library, this can cause an error when using the **db2imigr** command. To fix the error, you should reset the *library_path* environment variable so that it does not reference the libraries in those paths by entering the following command:

```
unset library_path
```

where *library_path* is:

LIBPATH on AIX

SHLIB_PATH on HP-UX

LD_LIBRARY_PATH on Solaris or SCO UnixWare 7

After migrating the DB2 instance, you should reset *LIBPATH* to its original setting.

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

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

where <i>DB2DIR</i>	= <code>/usr/lpp/db2_05_00</code>	on AIX
	= <code>/opt/IBMDB2/V5.0</code>	on HP-UX, SCO UnixWare 7, or Solaris

and where:

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

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

Notes:

- a. The authentication type of the instance applies to all databases owned by the instance.
- b. While authentication type (DCE) is an optional parameter, it is not valid to choose (DCE) for this command.

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

InstName Is the login name of the instance owner.



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

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



To install clients, see Chapter 2, "Installing DB2 Clients" on page 11.

Chapter 8. Installing DB2 Clients on Macintosh Operating Systems

To install the DB2 Client Application Enabler or the DB2 Software Developer's Kit on a Macintosh operating system, perform the following steps:

- 1 Insert the appropriate CD-ROM into the drive:
 - To install the DB2 Client Application Enabler, use the DB2 Client Application Enablers CD-ROM containing the Macintosh client.
 - To install the DB2 Software Developer's Kit, use the Application Developer's Kit CD-ROM.
- 2 Double-click on the **DB2_CAE_V502** icon.
- 3 Double-click on the **MAC** folder.
- 4 Double-click on the *language* folder, where *language* is the two-character country code that represents your language (for example, **EN** for English). Table 23 on page 278 lists the codes for each available language.
- 5 Double-click on the appropriate folder, either the **CLIENT** folder to install the DB2 Client Application Enabler, or the **SDK** folder to install the DB2 Software Developer's Kit.
- 6 Double-click on the installer icon, either the **CAEMAC** icon or the **SDKMAC** icon to start the installation.
- 7 Follow the instructions provided to complete the installation.
- 8 When the install program is finished, click on the **Restart** push button to restart your computer and complete the installation.



To configure your client to access a remote DB2 server, see Chapter 10, "Configuring Client-to-Server Communications Using the Command Line Processor" on page 55.

Part 2. Configuring DB2 Communications

Chapter 9. Configuring Client-to-Server Communications Using the Client Configuration Assistant



The information in this section describes how to use the Client Configuration Assistant to configure OS/2 and Windows 32-bit operating systems to access remote DB2 Universal Database servers.

To configure all other clients, see Chapter 10, "Configuring Client-to-Server Communications Using the Command Line Processor" on page 55.

Use the Client Configuration Assistant to configure your OS/2 and Windows 32-bit operating systems to access remote DB2 servers. The Client Configuration Assistant provides three configuration methods:

- One method makes use of a server's access profile

- One method searches the network for databases

- One method requires that you enter the database name and the communication protocol parameters for the DB2 server.

Perform the following steps to configure your workstation to access remote servers:

- 1 Start the Client Configuration Assistant.

For OS/2:

Double-click on the **Client Configuration Assistant** icon, located in the **DB2 for OS/2** folder.

For Windows 32-bit operating systems:

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 a server profile, select the **Use an access profile** radio button and click on the **Next** push button.

- a Click on the **Browse** push button and select the appropriate access profile.

- b** You are presented with a list of systems, instances, and databases. Select the database that you want to add 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.

- 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 a 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.



The **Other Systems (Search the network)** icon will only appear if the client's *DISCOVER* parameter is set to SEARCH (this is the default setting).

- c** If the system that contains the database that you want to add 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 a system to get a list of the instances and databases in it. Select the database that you want to add and proceed to Step 3.
- d** If the system you want is still not listed, it can be added to the list of systems by clicking on the **Add System** push button. Enter the required communication protocol parameters for the remote Administration Server and click on **OK**. For more information, click on the **Help** push button.
Select the database that you want to add and proceed to Step 3.



The Client Configuration Assistant may be unable to detect the remote system if:

The Administration Server *DISCOVER* configuration parameter on the remote system is set to *DISCOVER=DISABLE*.

The *DB2COMM* registry value and the *DISCOVER_COMM* configuration parameter are not configured on the Administration Server of the remote system with a discovery protocol being used on the client (the client discovery protocol is specified by the *DISCOVER_COMM* configuration parameter in the client's configuration).

The Administration Server is not running on the remote system.

By default, the Discovery function will search the network for 40 seconds; this may not be long enough to detect the remote system. You can configure the *DB2DISCOVERYTIME* registry value to specify a longer period of time.

The network that the Discovery request is running on may have been configured so that the Discovery request does not reach the remote system desired.

If you are using NetBIOS as the Discovery protocol, you may need to configure the *DB2NBDISCOVERRCVBUFS* registry value to a larger value to enable the client to receive more concurrent Discovery replies.

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 and click on the **Next** push button. If you selected the **TCP/IP** or **APPC** radio button, select the radio button that corresponds to the type of system where the database that you are trying to connect to resides.
- b** Enter the required communication protocol parameters and click on the **Next** push button. For more information, click on the **Help** push button.
- c** Enter the database alias name of the database that you want to connect to in the **Target database** field and click on the **Next** push button.



When a database is created on the remote server, if a database alias is not specified during database creation, the database is created with a database alias=*database_name*; otherwise, the database alias is the name specified.

- 3** To specify a local database alias name or to add a description, click on the **Next** push button. If you do not want to specify a database alias name, or add a description, click on the **Done** push button. If you do not specify a database alias name, the default will be the same as the remote database alias 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:
 - a Select the **Register this database for ODBC** check box, if it is not pre-selected.
 - b Select the radio button that describes the type of data source that you want to register this database as. For more information, click on the **Help** push button.
 - c Click on the **Application** drop down box and select the application that you want to use.
 - d Click on the **Done** push button to finish using the Add Database SmartGuide and add the database that you selected.
- 6 The Confirmation window opens. Click on the **Test Connection** push button to test the connection to this 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 this database. If you want to access another database, click on 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 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. Refer to the *Quick Beginnings* manual for more information.

- 1 Start the database manager by entering the **db2start** command on the server.
- 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 did not provide an alias when configuring the client, the default is the same as the database name (*database_name*). The database alias (*database_alias*) is the name that you use when connecting to a database from a client.

For more information, see “Connecting to a Database” on page 120.

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.

Chapter 10. Configuring Client-to-Server Communications Using the Command Line Processor

This section describes how to use commands to configure a client to communicate with a DB2 server using the command line processor.



If you are planning to use an OS/2 or Windows 32-bit client for communications, the Client Configuration Assistant (CCA) makes it easy to automate the tasks of configuring and administering DB2 clients to communicate with DB2 servers. If you have installed the CCA, it is recommended that you use this utility to configure your DB2 clients for communications. See Chapter 9, "Configuring Client-to-Server Communications Using the Client Configuration Assistant" on page 49 for more information.

For instructions on using the command line processor, see Chapter 14, "Entering DB2 Commands and SQL Statements" on page 151. For a description of database naming rules, see Appendix C, "Naming Rules" on page 279.



Go to the communication protocol that you want your client to use to access a remote server.

Named Pipes - see "Configuring Named Pipes on the Client"

TCP/IP - see "Configuring TCP/IP on the Client" on page 60

NetBIOS - see "Configuring NetBIOS on the Client" on page 69

IPX/SPX - see "Configuring IPX/SPX on the Client" on page 77

APPC - see "Configuring APPC on the Client" on page 85

Configuring Named Pipes on the Client

This section assumes that Named Pipes is functional on the client and DB2 server workstations. See "Software Requirements" on page 5 for the communication protocol requirements for your platform. See "Possible Client-to-Server Connectivity Scenarios" on page 9 for the supported communication protocols for your particular client and server.

The following steps are required to set up a client to use Named Pipe communications:

- 1** Identify and record parameter values.

- 2 Configure the client:
 - a Catalog the Named Pipes node.
 - b Catalog the database.
- 3 Test the connection between the client and server.

Step 1. Identify and Record Parameter Values

As you proceed through the configuration steps, complete the *Your Value* column in the following table. You can fill in some of the values before you start configuring this protocol.

Table 9. Named Pipe Values Required at the Client

Parameter	Description	Sample Value	Your Value
Computer name (<i>computer_name</i>)	The computer name of the server machine where the DB2 database resides. On the server machine, to locate the value for this parameter, click on Start and select Settings-> Control Panel . Double-click on the Network folder and select the Identification tab. Record the computer name.	server1	
Instance name (<i>instance_name</i>)	The name of the DB2 instance where the database resides.	db2	
Node name (<i>node_name</i>)	A local alias, or nickname, that describes the node where the database resides. You can choose any name you want, however, all node name values within your local node directory must be unique.	db2node	

Step 2. Configure the Client

The following steps describe how to configure this protocol on the client. Replace the sample values with your worksheet values.

A. Catalog the Named Pipes Node

You must add an entry to the client's node directory to describe the remote node.

This entry specifies the chosen alias (*node_name*), the server's Computer name (*computer_name*), and the Instance name (*instance_name*) that the client will use to access the remote server.

To catalog the Named Pipes node, perform the following steps:

- 1 Log on to the system as a user with System Administrative (SYSADM) or System Controller (SYSCTRL) authority.



If you have trouble logging on to the system, see “Logging on to the System” on page 119.

- 2 Catalog the node by entering the following commands in the command line processor:

```
catalog npipe node node_name remote computer_name instance instance_name  
terminate
```

For example, to catalog a remote node called *db2node*, which is located on the server called *server1*, in the *db2* instance, use:

```
catalog npipe node db2node remote server1 instance db2  
terminate
```



If you need to change values that were set with the **catalog node** command, first run the **uncatalog node** command in the command line processor as follows:

```
uncatalog node node_name
```

Recatalog the node with the value that you want to use.

B. Catalog the Database

Before a client application can access a remote database, the database must be cataloged on the server node and on any client nodes that will connect to it. When you create a database, it is automatically cataloged on the server with the database alias (*database_alias*) the same as the database name (*database_name*). The information in the database directory, along with the information in the node directory, is used on the client to establish a connection to the remote database.

To catalog a database on the client, perform the following steps.

- 1 Log on to the system as a user with System Administrative (SYSADM) or System Controller (SYSCTRL) authority.



If you have trouble logging on to the system, see “Logging on to the System” on page 119.

- 2 Fill in the *Your Value* column in the following worksheet.

Table 10. Worksheet: Parameter Values for Cataloging Databases

Parameter	Description	Sample Value	Your Value
Database name (<i>database_name</i>)	The database alias (<i>database_alias</i>) of the <i>remote</i> database. When you create a database, it is automatically cataloged on the server with the database alias (<i>database_alias</i>) the same as the database name (<i>database_name</i>).	sample	
Database alias (<i>database_alias</i>)	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 (<i>database_name</i>). This is the name that you use when connecting to a database from a client.	tor1	
Node name (<i>node_name</i>)	The name of the node directory entry that describes where the database resides. Use the same value for node name (<i>node_name</i>) that you used to catalog the node in the previous step.	db2node	

3 Catalog the database by entering the following commands in the command line processor:

```
catalog database database_name as database_alias at node node_name
terminate
```

For example, to catalog a remote database called *sample* so that it has the alias *tor1*, on the node *db2node*, use:

```
catalog database sample as tor1 at node db2node
terminate
```



If you need to change values that were set with the **catalog database** command, first run the **uncatalog database** command in the command line processor as follows:

```
uncatalog database database_alias
```

Recatalog the database with the value that you want to use.

Step 3. Test the Client-to-Server Connection

When the configuration of the 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. Refer to the *Quick Beginnings* manual for more information.

- 1 Start the database manager by entering the **db2start** command on the server (if it was not automatically started at boot time).
- 2 Enter the following command in the client's Command Center or command line processor to connect the client to the remote database:

```
connect to database_alias user userid using password
```

The values for *userid* and *password* must be valid for the system on which they are authenticated. By default, authentication takes place on the SERVER. If the database manager is configured for CLIENT authentication, the *userid* and *password* must be valid on the client.

If the connection is successful, you will get a message showing the name of the database to which you have connected. You are now able to retrieve data from that database. For example, to retrieve a list of all the table names listed in the system catalog table, enter the following SQL command in the Command Center or command line processor:

```
"select tabname from syscat.tables"
```

When you are finished using the database connection, enter the **connect reset** command to end the database connection.



You are ready to start using the DB2 server. See Chapter 11, "Getting Started with DB2 Universal Database" on page 119 for details.

Troubleshooting the Client-to-Server Connection

If the connection fails, check the following items:

At the *server*:

- 1 The *db2comm* registry value includes the value *npipe*.



Check the settings for the *db2comm* registry value by entering the **db2set DB2COMM** command. For more information, see Chapter 15, “Controlling Your DB2 Environment” on page 157.

- 2 The security service was started (enter the **net start db2ntsecserver** command - for Windows NT servers only).
- 3 The database was created and cataloged properly.
- 4 The database manager was stopped and started again (enter the **db2stop** and **db2start** commands on the server).



If there are problems starting a protocol's connection managers, a warning message is displayed and the error messages are logged in the *db2diag.log* file.

Refer to the *Troubleshooting Guide* for information on the *db2diag.log* file.

At the *client*:

- 1 The node was cataloged with the correct computer name (*computer_name*) and instance name (*instance_name*), of the server, where the database resides.
- 2 The node name (*node_name*), specified in the database directory, points to the correct entry in the node directory.
- 3 The database was cataloged properly, using the *server's* database alias (*database_alias*) that was cataloged when the database was created on the server, as the database name (*database_name*) on the client.

After you verify these items, refer to the *Troubleshooting Guide* if the connection still fails.

Configuring TCP/IP on the Client

This section assumes that TCP/IP is functional on the client and DB2 server workstations. See “Software Requirements” on page 5 for the communication protocol requirements for your platform. See “Possible Client-to-Server Connectivity Scenarios” on page 9 for the supported communication protocols for your particular client and server.

The following steps are required to set up a DB2 client to use TCP/IP communications:

- 1 Identify and record parameter values.
- 2 Configure the client:

- a Resolve the server's host address.
 - b Update the `services` file.
 - c Catalog a TCP/IP node.
 - d Catalog the database.
- 3 Test the connection between the client and server.



Due to the characteristics of the TCP/IP protocol, the TCP/IP subsystem may not be immediately notified of the failure of a partner on another host. As a result, a client application accessing a remote DB2 server using TCP/IP, or the corresponding agent at the server, may sometimes appear to be hung. DB2 uses the TCP/IP SO_KEEPALIVE socket option to detect when there has been a failure and the TCP/IP connection has been broken.

If you are experiencing problems with your TCP/IP connection, refer to the *Troubleshooting Guide* for information on this parameter and other common TCP/IP problems.

Step 1. Identify and Record Parameter Values

As you proceed through the configuration steps, complete the *Your Value* column in the following table. You can fill in some of the values before you start configuring this protocol.

Table 11 (Page 1 of 2). TCP/IP Values Required at the Client			
Parameter	Description	Sample Value	Your Value
Hostname (<i>hostname</i>) or IP address (<i>ip_address</i>)	Use the <i>hostname</i> or <i>ip_address</i> of the remote server workstation. To resolve this parameter: Issue the hostname command at the server to obtain the <i>hostname</i> . Contact the server administrator to obtain the <i>ip_address</i> . (On AIX, you can enter the ping server_hostname command to obtain the <i>ip_address</i>).	serverhost or 9.21.15.235	

Table 11 (Page 2 of 2). TCP/IP Values Required at the Client

Parameter	Description	Sample Value	Your Value
Connection Port Connection Service name (<i>svcname</i>) Port number/Protocol (<i>port_number/tcp</i>)	Values required in the <i>services</i> file. The Connection Service name is an arbitrary name used to represent the port number (<i>port_number</i>) on the client. The port number for the client must be the same as the port number that the <i>svcname</i> parameter maps to in the <i>services</i> file at the server. (The <i>svcname</i> parameter is located in the database manager configuration file on the DB2 server.) This value must not be in use by any another applications, and must be unique within the <i>services</i> file. Contact your LAN administrator for the values used to configure the server.	server1 37 /tcp	
Node name (<i>node_name</i>)	A local alias, or nickname, that describes the node where the database resides. You can choose any name you want, however, all node name values within your local node directory must be unique.	db2node	

Step 2. Configure the Client

The following steps describe how to configure this protocol on the client. Replace the sample values with your worksheet values.

A. Resolve the Server's Host Address



If your network has a name server, or you are planning to directly specify the IP address (*ip_address*) of the server, skip this step and proceed to “Step B. Update the Services File” on page 64.

The client must know the address of the server to which it is attempting to establish communications. If a name server does not exist on your network, you may directly specify a hostname that maps to the IP address (*ip_address*) of the host in the local *hosts* file. See Table 12 on page 63 for the location of the *hosts* file for your particular platform.



If you are planning on supporting a UNIX client that is using Network Information Services (NIS), and you are not using a name server on your network, you must update the `hosts` file located on your NIS master server.

Table 12. Location of the Local Hosts and Services Files

Platform	Location
Macintosh	The <code>hosts</code> file is located in the folder called System Folder . Note: This operating system does not use a <code>services</code> file. You must catalog this node using the <code>port_number</code> parameter. See "Step C. Catalog a TCP/IP Node" on page 64 for more information.
OS/2	Specified by the <code>etc</code> environment variable. Enter the <code>set etc</code> command to determine the location of your local <code>hosts</code> or <code>services</code> files. Note: For DOS and WIN-OS2 sessions, you might need to update the <code>hosts</code> and <code>services</code> files located in the <code>tcpip_product\dos\etc</code> directory.
Windows 3.x	Typically in the <code>tcpip_product\etc</code> directory, but it depends on the products that you have installed. Refer to your TCP/IP documentation for more information.
Windows 95 and Windows 98	<code>windows</code> directory
Windows NT	<code>winnt\system32\drivers\etc</code> directory
UNIX	<code>/etc</code> directory

Using a text editor, add an entry to the client's `hosts` file for the server's hostname. For example:

```
9.21.15.235    serverhost    # host address for serverhost
```

where:

`9.21.15.235` is the *ip_address*

`serverhost` is the *hostname*

`#` is a comment describing the entry

Notes:

1. If the server is not in the same domain as the client, you must provide a fully qualified domain name such as `serverhost.vnet.ibm.com`, where `vnet.ibm.com` is the domain name.
2. For specific information on resolving host addresses, refer to your TCP/IP documentation.

Step B. Update the Services File



If you are planning to catalog a TCP/IP node using the port number (*port_number*), skip this step and go to “Step C. Catalog a TCP/IP Node.”

If you are configuring a DB2 for Macintosh client, you must catalog the TCP/IP node using the port number (*port_number*). Go to “Step C. Catalog a TCP/IP Node” to catalog the node.

Using a local text editor, add the Connection Service name and port number to the client's `services` file for TCP/IP support. For example:

```
server1 37 /tcp # DB2 connection service port
```

where:

server1 is the Connection Service name

3700 is the port number for the Connection Port

tcp is the communication protocol that you are using

is a comment describing the entry

The port number used on the client must match the port number used on the server.



If you are planning on supporting a UNIX client that is using Network Information Services (NIS), you must update the `services` file located on your NIS master server.

The file called `services` is located in the same directory as the local `hosts` file that you may have edited in “A. Resolve the Server's Host Address” on page 62.

See Table 12 on page 63 for the location of the `services` file for your particular platform.

Step C. Catalog a TCP/IP Node

You must add an entry to the client's node directory to describe the remote node.

This entry specifies the chosen alias (*node_name*), the *hostname* (or *ip_address*), and the *svcname* (or *port_number*) that the client will use to access the remote server.

To catalog a TCP/IP node, perform the following steps:

- 1 Log on to the system as a user with System Administrative (SYSADM) or System Controller (SYSCTRL) authority.



If you have trouble logging on to the system, see “Logging on to the System” on page 119.



If you are configuring communications for a Windows 3.x client, you can use the Client Setup tool to complete this task. Go to “Using the Client Setup Tool to Configure a Windows 3.x Client” on page 115 to catalog a node using the Client Setup tool.

- 2 If you are using a UNIX client, set up the instance environment and invoke the DB2 command line processor. Run the start-up script as follows:

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

where *INSTHOME* is the home directory of the instance.

- 3 Catalog the node by entering the following commands in the command line processor:

```
catalog tcpip node node_name remote [ hostname | ip_address ] server [ svcename | port_number ]
terminate
```

For example, to catalog the remote server *serverhost* on the node called *db2node*, using the service name *server1*, use:

```
catalog tcpip node db2node remote serverhost server server1
terminate
```

To catalog a remote server with the IP address *9.21.15.235* on the node called *db2node*, using the Port number *3700*, use:

```
catalog tcpip node db2node remote 9.21.15.235 server 37
terminate
```



If you need to change values that were set with the **catalog node** command, first run the **uncatalog node** command in the command line processor as follows:

```
uncatalog node node_name
```

Recatalog the node with the value that you want to use.

Step D. Catalog the Database

Before a client application can access a remote database, the database must be cataloged on the server node and on any client nodes that will connect to it. When you create a database, it is automatically cataloged on the server with the database alias (*database_alias*) the same as the database name (*database_name*). The information in the database directory, along with the information in the node directory, is used on the client to establish a connection to the remote database.

To catalog a database on the client, perform the following steps.

- 1 Log on to the system as a user with System Administrative (SYSADM) or System Controller (SYSCTRL) authority.



If you have trouble logging on to the system, see “Logging on to the System” on page 119.

- 2 Fill in the *Your Value* column in the following worksheet.

Table 13. Worksheet: Parameter Values for Cataloging Databases

Parameter	Description	Sample Value	Your Value
Database name (<i>database_name</i>)	The database alias (<i>database_alias</i>) of the <i>remote</i> database. When you create a database, it is automatically cataloged on the server with the database alias (<i>database_alias</i>) the same as the database name (<i>database_name</i>).	sample	
Database alias (<i>database_alias</i>)	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 (<i>database_name</i>). This is the name that you use when connecting to a database from a client.	tor1	
Node name (<i>node_name</i>)	The name of the node directory entry that describes where the database resides. Use the same value for node name (<i>node_name</i>) that you used to catalog the node in the previous step.	db2node	



If you are configuring communications for a Windows 3.x client, you can use the Client Setup tool to complete this task. Go to “Using the Client Setup Tool to Configure a Windows 3.x Client” on page 115 to catalog a database using the Client Setup tool.

- 3 If you are using a UNIX client, set up the instance environment and invoke the DB2 command line processor. Run the start-up script as follows:


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

 where *INSTHOME* is the home directory of the instance.
- 4 Catalog the database by entering the following commands in the command line processor:

```
catalog database database_name as database_alias at node node_name
terminate
```

For example, to catalog a remote database called *sample* so that it has the alias *tor1*, on the node *db2node*, use:

```
catalog database sample as tor1 at node db2node
terminate
```



If you need to change values that were set with the **catalog database** command, first run the **uncatalog database** command in the command line processor as follows:

```
uncatalog database database_alias
```

Recatalog the database with the value that you want to use.

Step 3. Test the Client-to-Server Connection

When the configuration of the 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. Refer to the *Quick Beginnings* manual for more information.

- 1 Start the database manager by entering the **db2start** command on the server (if it was not automatically started at boot time).
- 2 Enter the following command in the client's Command Center or command line processor to connect the client to the remote database:

```
connect to database_alias user userid using password
```

The values for *userid* and *password* must be valid for the system on which they are authenticated. By default, authentication takes place on the SERVER. If the database manager is configured for CLIENT authentication, the *userid* and *password* must be valid on the client.

If the connection is successful, you will get a message showing the name of the database to which you have connected. You are now able to retrieve data from that database. For example, to retrieve a list of all the table names listed in the system catalog table, enter the following SQL command in the Command Center or command line processor:

```
"select tabname from syscat.tables"
```

When you are finished using the database connection, enter the **connect reset** command to end the database connection.



You are ready to start using the DB2 server. See Chapter 11, "Getting Started with DB2 Universal Database" on page 119 for details.

Troubleshooting the Client-to-Server Connection

If the connection fails, check the following items:

At the *server*:

- 1 The *db2comm* registry value includes the value `tcPIP`.



Check the settings for the *db2comm* registry value by entering the **db2set DB2COMM** command. For more information, see Chapter 15, "Controlling Your DB2 Environment" on page 157.

- 2 The *services* file was updated correctly.
- 3 The service name (*svcename*) parameter was updated correctly in the database manager configuration file.
- 4 The security service was started (enter the **net start db2ntsecserver** command - for Windows NT servers only).
- 5 The database was created and cataloged properly.
- 6 The database manager was stopped and started again (enter the **db2stop** and **db2start** commands on the server).



If there are problems starting a protocol's connection managers, a warning message is displayed and the error messages are logged in the *db2diag.log* file.

Refer to the *Troubleshooting Guide* for information on the *db2diag.log* file.

At the *client*:

- 1 If used, the *services* and *hosts* files were updated correctly.
- 2 The node was cataloged with the correct hostname (*hostname*) or IP address (*ip_address*).
- 3 The port number must match, or the service name must map to, the port number used on the server.

- 4 The node name (*node_name*), specified in the database directory, points to the correct entry in the node directory.
- 5 The database was cataloged properly, using the *server's* database alias (*database_alias*) that was cataloged when the database was created on the server, as the database name (*database_name*) on the client.

After you verify these items, refer to the *Troubleshooting Guide* if the connection still fails.

Configuring NetBIOS on the Client

This section assumes that NetBIOS is functional on the client and DB2 server workstations. See “Software Requirements” on page 5 for the communication protocol requirements for your platform. See “Possible Client-to-Server Connectivity Scenarios” on page 9 for the supported communication protocols for your particular client and server.

The following steps are required to set up a client to use NetBIOS communications:

- 1 Identify and record parameter values.
- 2 Configure the client:
 - a Record the network route for the Logical adapter number.
 - b Update the database manager configuration file.
 - c Catalog the NetBIOS node.
 - d Catalog the database.
- 3 Test the connection between the client and server.

Step 1. Identify and Record Parameter Values

As you proceed through the configuration steps, complete the *Your Value* column in the following table. You can fill in some of the values before you start configuring this protocol.

<i>Table 14 (Page 1 of 2). NetBIOS Values Required at the Client</i>			
Parameter	Description	Sample Value	Your Value
Logical adapter number (<i>adapter_number</i>)	The local logical adapters that will be used for the NetBIOS connection.		

Parameter	Description	Sample Value	Your Value
Workstation name (<i>nname</i>) - on the client	The NetBIOS name of the <i>client</i> workstation. <i>nname</i> is chosen by the user and must be unique among all NetBIOS nodes in the network.	client1	
Workstation name (<i>nname</i>) - on the server	The NetBIOS name of the <i>server</i> workstation. Locate this parameter in the database manager configuration file on the server.	server1	
Node name (<i>node_name</i>)	A local alias, or nickname, that describes the node where the database resides. You can choose any name you want, however, all node name values within your local node directory must be unique.	db2node	

Step 2. Configure the Client

The following steps describe how to configure this protocol on the client. Replace the sample values with your worksheet values.

A. Record the Logical Adapter Number Used for the NetBIOS Connection

To view and record the logical adapter number (*adapter_number*) used for the NetBIOS connection, perform the following:

For OS/2:

- 1 Double-click on the **MPTS** icon.
- 2 Click on the **Configure** push button.
- 3 Select the **LAN adapters and protocols** radio button and click on the **Configure** push button.
- 4 Record the Logical adapter number associated with the **IBM OS/2 NETBIOS** entry in the Current Configuration window.
- 5 Click on the **Cancel** push button.
- 6 Click on the **Close** push button.
- 7 Click on the **Exit** push button.

For Windows 3.x

Record the logical adapter number *0*. If you experience problems making your NetBIOS connection, refer to the *Troubleshooting Guide* .

For Windows 95 and Windows 98:



There is no interface on the Windows 95 or Windows 98 operating system that you can use to record the adapter number used for the NetBIOS connection. Complete the following steps to have the NetBIOS protocol default to use the adapter number *0*.

- 1 Click on **Start** and select **Settings->Control Panel**.
- 2 Double-click on the **Network** icon.
- 3 Select the **NetBEUI** icon from the following Network components are installed window.
- 4 Click on the **Properties** push button.
- 5 Select the **Advanced** tab.
- 6 Select the **Set this protocol to be the default** check box.
- 7 Click on **OK** to exit this window.
- 8 Click on **OK**.
- 9 Record the value *0* as the logical adapter number on your worksheet.
- 10 Shut down and reboot your system for these changes to take effect.

For Windows NT:



The logical adapter number (*adapter_number*) that you use *must* be associated with the **Nbf** Network Route for native NetBIOS.

- 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** icon 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 Click on **OK**.
- 6 Click on the **Close** push button.

B. Update the Database Manager Configuration File

You must update the database manager configuration file with the *client's* workstation name (*nname*) parameter.

To update the database manager configuration file, perform the following steps:

- 1 Log on to the system with a user account that belongs to the local Administrators group on each machine in your partitioned database system.



If you have trouble logging on to the system, see "Logging on to the System" on page 119.

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

```
update database manager configuration using nname nname
terminate
```

For example, if the client's workstation name (*nname*) is `client1`, use:

```
update database manager configuration using nname client1
terminate
```

C. Catalog the NetBIOS Node

You must add an entry to the client's node directory to describe the remote node.

This entry specifies the chosen alias (*node_name*), the remote *server's* workstation name (*nname*), and the Logical adapter number (*adapter_number*) that the client will use to access the remote DB2 server.

To catalog the NetBIOS node, perform the following steps:

- 1 Log on to the system as a user with System Administrative (SYSADM) or System Controller (SYSCTRL) authority.



If you have trouble logging on to the system, see "Logging on to the System" on page 119.



If you are configuring communications for a Windows 3.x client, you can use the Client Setup tool to complete this task. Go to "Using the Client Setup Tool to Configure a Windows 3.x Client" on page 115 to catalog a node using the Client Setup tool.

- 2 Catalog the node by entering the following commands in the command line processor:

```
catalog netbios node node_name remote nname adapter adapter_number
terminate
```

For example, to catalog a remote database server *server1* on the node called *db2node*, using the logical adapter number *0*, use:

```
catalog netbios node db2node remote server1 adapter
terminate
```



If you need to change values that were set with the **catalog node** command, first run the **uncatalog node** command in the command line processor as follows:

```
uncatalog node node_name
```

Recatalog the node with the value that you want to use.

D. Catalog the Database

Before a client application can access a remote database, the database must be cataloged on the server node and on any client nodes that will connect to it. When you create a database, it is automatically cataloged on the server with the database alias (*database_alias*) the same as the database name (*database_name*). The information in the database directory, along with the information in the node directory, is used on the client to establish a connection to the remote database.

To catalog a database on the client, perform the following steps.

- 1 Log on to the system as a user with System Administrative (SYSADM) or System Controller (SYSCTRL) authority.



If you have trouble logging on to the system, see "Logging on to the System" on page 119.

- 2 Fill in the *Your Value* column in the following worksheet.

Table 15. Worksheet: Parameter Values for Cataloging Databases

Parameter	Description	Sample Value	Your Value
Database name (<i>database_name</i>)	The database alias (<i>database_alias</i>) of the <i>remote</i> database. When you create a database, it is automatically cataloged on the server with the database alias (<i>database_alias</i>) the same as the database name (<i>database_name</i>).	sample	
Database alias (<i>database_alias</i>)	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 (<i>database_name</i>). This is the name that you use when connecting to a database from a client.	tor1	
Node name (<i>node_name</i>)	The name of the node directory entry that describes where the database resides. Use the same value for node name (<i>node_name</i>) that you used to catalog the node in the previous step.	db2node	



If you are configuring communications for a Windows 3.x client, you can use the Client Setup tool to complete this task. Go to “Using the Client Setup Tool to Configure a Windows 3.x Client” on page 115 to catalog a database using the Client Setup tool.

3 Catalog the database by entering the following commands in the command line processor:

```
catalog database database_name as database_alias at node node_name
terminate
```

For example, to catalog a remote database called *sample* so that it has the alias *tor1*, on the node *db2node*, use:

```
catalog database sample as tor1 at node db2node
terminate
```



If you need to change values that were set with the **catalog database** command, first run the **uncatalog database** command in the command line processor as follows:

```
uncatalog database database_alias
```

Recatalog the database with the value that you want to use.

Step 3. Test the Client-to-Server Connection

When the configuration of the 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. Refer to the *Quick Beginnings* manual for more information.

- 1 Start the database manager by entering the **db2start** command on the server (if it was not automatically started at boot time).
- 2 Enter the following command in the client's Command Center or command line processor to connect the client to the remote database:

```
connect to database_alias user userid using password
```

The values for *userid* and *password* must be valid for the system on which they are authenticated. By default, authentication takes place on the SERVER. If the database manager is configured for CLIENT authentication, the *userid* and *password* must be valid on the client.

If the connection is successful, you will get a message showing the name of the database to which you have connected. You are now able to retrieve data from that database. For example, to retrieve a list of all the table names listed in the system catalog table, enter the following SQL command in the Command Center or command line processor:

```
"select tabname from syscat.tables"
```

When you are finished using the database connection, enter the **connect reset** command to end the database connection.



You are ready to start using the DB2 server. See Chapter 11, "Getting Started with DB2 Universal Database" on page 119 for details.

Troubleshooting the Client-to-Server Connection

If the connection fails, check the following items:

At the *server*:

- 1 The *db2comm* registry value includes the value *netbios*.

- 2 The logical adapter number is equal to 0 (or the *DB2NBADAPTERS* registry value was updated to override the default value).
- 3 The server's workstation name (*nname*) parameter was updated correctly in the database manager configuration file (or the admin configuration file, if you are setting up the Administration Server)
- 4 The network route associated with the logical adapter number is **Nbf** (for Windows NT servers only).
- 5 The security service was started (enter the **net start db2ntsecserver** command - for Windows NT servers only).
- 6 The database was created and cataloged properly.
- 7 The database manager was stopped and started again (enter the **db2stop** and **db2start** commands on the server).



If there are problems starting a protocol's connection managers, a warning message is displayed and the error messages are logged in the *db2diag.log* file.

Refer to the *Troubleshooting Guide* for information on the *db2diag.log* file.

At the *client*:

- 1 The client's workstation name (*nname*) parameter was updated correctly in the database manager configuration file
- 2 The node was cataloged with the correct *server's* workstation name (*nname*) and logical adapter number (*adapter_number*)
- 3 The node name (*node_name*), specified in the database directory, points to the correct entry in the node directory.
- 4 The database was cataloged properly, using the *server's* database alias (*database_alias*) that was cataloged when the database was created on the server, as the database name (*database_name*) on the client.

After you verify these items, refer to the *Troubleshooting Guide* if the connection still fails.

Configuring IPX/SPX on the Client

This section assumes that IPX/SPX is functional on the client and DB2 server workstations. See “Software Requirements” on page 5 for the communication protocol requirements for your platform. See “Possible Client-to-Server Connectivity Scenarios” on page 9 for the supported communication protocols for your particular client and server.

A client can access a DB2 server via Direct Addressing or File Server Addressing. See Table 16 for a list of the available IPX/SPX clients and their supported addressing methods. For a description of Direct Addressing or File Server Addressing, refer to the *Quick Beginnings* manual.



You need to know the IPX/SPX addressing method that was used to configure the server before you configure your DB2 client. If the DB2 server was configured to use Direct Addressing, you must configure your client to use Direct Addressing to communicate with the server. If the DB2 server was configured for File Server Addressing, you can choose to configure your client to use either Direct Addressing or File Server Addressing, provided that your client supports the method that you want to use. See Table 16 for a list of client specific IPX/SPX supported addressing methods.

Compare the IPX/SPX parameters in the database manager configuration file on the server machine, with those sample values in the "IPX/SPX Values Required at the Server" table of the *Quick Beginnings* manual in the "Configuring Client-to-Server Communications Using the Command Line Processor" chapter.

Table 16. IPX/SPX Supported Communication Methods for a DB2 Client

Client Platform	Direct Addressing	File Server Addressing
Macintosh	no support	
OS/2	*	*
SCO UnixWare 7	*	
UNIX (except SCO UnixWare 7)	no support	
SCO OpenServer	*	
Windows 3.x	*	*
Windows 95 and Windows 98	*	
Windows NT	*	

The following steps are required to set up a DB2 client to use IPX/SPX communications:

- 1 Identify and record parameter values.

- 2 Configure the client:
 - a Catalog the IPX/SPX node.
 - b Catalog the database.
- 3 Test the connection between the client and server.

Step 1. Identify and Record Parameter Values

As you proceed through the configuration steps, complete the *Your Value* column in the following table. You can fill in some of the values before you start configuring this protocol.

Table 17 (Page 1 of 2). IPX/SPX Values Required at the Client

Parameter	Description	Sample Value	Your Value
File server name (FILESERVER)	<p>Direct Addressing A * value indicates that you are using Direct Addressing.</p> <p>File Server Addressing The name of the NetWare file server where the database server instance is registered. This parameter must be entered in UPPERCASE.</p> <p>Locate this parameter in the database manager configuration file on the server.</p>	<p>Direct Addressing</p> <p>File Server Addressing NETWSRV</p>	

Table 17 (Page 2 of 2). IPX/SPX Values Required at the Client

Parameter	Description	Sample Value	Your Value
DB2 server object name (<i>OBJECTNAME</i>)	<p>Direct Addressing The server's IPX/SPX internetwork address of the form: netid (8 byte).nodeid (12 byte). socket # (4 byte)</p> <p>To resolve this parameter, enter the db2ipxad command at the server. See "A. Catalog the IPX/SPX Node" on page 79 for more information.</p> <p>File Server Addressing The database manager server instance, represented as the object <i>OBJECTNAME</i> on the NetWare file server. The server's IPX/SPX internetwork address is stored and retrieved from this object.</p> <p>This parameter must be entered in UPPERCASE and be unique on the NetWare file server system.</p> <p>Locate this parameter in the database manager configuration file on the server.</p>	<p>Direct Addressing 92127 .4 11527745.879E</p> <p>File Server Addressing DB2INST1</p>	
Node name (<i>node_name</i>)	A local alias, or nickname, that describes the node where the database resides. You can choose any name you want, however, all node name values within your local node directory must be unique.	db2node	

Step 2. Configure the Client

The following steps describe how to configure this protocol on the client. Replace the sample values with your worksheet values.

A. Catalog the IPX/SPX Node

You must add an entry to the client's node directory to describe the remote node.

This entry specifies the chosen alias (*node_name*), the file server name (*FILESERVER*), and the DB2 server object name (*OBJECTNAME*) that the client will use to access the remote DB2 server.

To catalog the IPX/SPX node, perform the following steps:

- 1 Log on to the system as a user with System Administrative (SYSADM) or System Controller (SYSCTRL) authority.



If you have trouble logging on to the system, see “Logging on to the System” on page 119.



If you are configuring communications for a Windows 3.x client, you can use the Client Setup tool to complete this task. Go to “Using the Client Setup Tool to Configure a Windows 3.x Client” on page 115 to catalog a node using the Client Setup tool.

- 2 If you are using a UNIX client, set up the instance environment and invoke the DB2 command line processor. Run the start-up script as follows:

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

where *INSTHOME* is the home directory of the instance.

- 3 Catalog the node by issuing the following commands in the command line processor:

```
catalog ipxspx node node_name remote FILESERVER server OBJECTNAME
terminate
```

Direct Addressing Example

You must assign a * to the *FILESERVER* parameter and specify the server's IPX/SPX internetwork address value as the *OBJECTNAME* parameter on the client.

To determine the value for the *OBJECTNAME* parameter, enter the **db2ipxad** command on the server. (This command is located in the *sqlllib/misc/* directory for UNIX servers, and the *sqlllib\misc* directory for all other DB2 servers.)

Make note of the output that is generated, and use that value in place of the sample value (92127 .4 11527745.879E) in the example that follows.

To catalog a remote node called *db2node*, using the IPX/SPX internetwork address *09212700.400011527745.879E* as the *OBJECTNAME*, use:

```
catalog ipxspx node db2node remote server 92127 .4 11527745.879E
terminate
```

File Server Addressing Example

To catalog a remote node called *db2node* at the file server *NETWSRV*, in the instance *DB2INST1*, use:

```
catalog ipxspx node db2node remote NETWSRV server DB2INST1
terminate
```




If you need to change values that were set with the **catalog node** command, first run the **uncatalog node** command in the command line processor as follows:

```
uncatalog node node_name
```

Recatalog the node with the value that you want to use.

B. Catalog the Database

Before a client application can access a remote database, the database must be cataloged on the server node and on any client nodes that will connect to it. When you create a database, it is automatically cataloged on the server with the database alias (*database_alias*) the same as the database name (*database_name*). The information in the database directory, along with the information in the node directory, is used on the client to establish a connection to the remote database.

To catalog a database on the client, perform the following steps.

- 1** Log on to the system as a user with System Administrative (SYSADM) or System Controller (SYSCTRL) authority.



If you have trouble logging on to the system, see “Logging on to the System” on page 119.

- 2** Fill in the *Your Value* column in the following worksheet.

Parameter	Description	Sample Value	Your Value
Database name (<i>database_name</i>)	The database alias (<i>database_alias</i>) of the <i>remote</i> database. When you create a database, it is automatically cataloged on the server with the database alias (<i>database_alias</i>) the same as the database name (<i>database_name</i>).	sample	
Database alias (<i>database_alias</i>)	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 (<i>database_name</i>). This is the name that you use when connecting to a database from a client.	tor1	
Node name (<i>node_name</i>)	The name of the node directory entry that describes where the database resides. Use the same value for node name (<i>node_name</i>) that you used to catalog the node in the previous step.	db2node	



If you are configuring communications for a Windows 3.x client, you can use the Client Setup tool to complete this task. Go to “Using the Client Setup Tool to Configure a Windows 3.x Client” on page 115 to catalog a database using the Client Setup tool.

- 3 If you are using a UNIX client, set up the instance environment and invoke the DB2 command line processor. Run the start-up script as follows:

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

where *INSTHOME* is the home directory of the instance.

- 4 Catalog the database by entering the following commands in the command line processor:

```
catalog database database_name as database_alias at node node_name
terminate
```

For example, to catalog a remote database called *sample* so that it has the alias *tor1*, on the node *db2node*, use:

```
catalog database sample as tor1 at node db2node
terminate
```



If you need to change values that were set with the **catalog database** command, first run the **uncatalog database** command in the command line processor as follows:

```
uncatalog database database_alias
```

Recatalog the database with the value that you want to use.

Step 3. Test the Client-to-Server Connection

When the configuration of the 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. Refer to the *Quick Beginnings* manual for more information.

- 1 Start the database manager by entering the **db2start** command on the server (if it was not automatically started at boot time).
- 2 Enter the following command in the client's Command Center or command line processor to connect the client to the remote database:

```
connect to database_alias user userid using password
```

The values for *userid* and *password* must be valid for the system on which they are authenticated. By default, authentication takes place on the SERVER. If the database manager is configured for CLIENT authentication, the *userid* and *password* must be valid on the client.

If the connection is successful, you will get a message showing the name of the database to which you have connected. You are now able to retrieve data from that database. For example, to retrieve a list of all the table names listed in the system catalog table, enter the following SQL command in the Command Center or command line processor:

```
"select tabname from syscat.tables"
```

When you are finished using the database connection, enter the **connect reset** command to end the database connection.



You are ready to start using the DB2 server. See Chapter 11, "Getting Started with DB2 Universal Database" on page 119 for details.

Troubleshooting the Client-to-Server Connection

If the connection fails, check the following items:

At the *server*:

- 1 The *db2comm* registry value includes the value *ipxspix*. script file.



Check the settings for the *db2comm* registry value by entering the **db2set DB2COMM** command. For more information, see Chapter 15, "Controlling Your DB2 Environment" on page 157.

- 2 The *FILESERVER*, *OBJECTNAME*, and *IPX_SOCKET* parameters were updated correctly in the database manager configuration file.
- 3 The database was created and cataloged properly.
- 4 The security service was started (enter the **net start db2ntsecserver** command - for Windows NT servers only).
- 5 If you are using File Server Addressing, ensure that the DB2 server was registered at the NetWare file server *after* the database manager configuration file was updated with the required IPX/SPX parameters. For more information, refer to the *Quick Beginnings* manual.
- 6 The database manager was stopped and started again (enter the **db2stop** and **db2start** commands on the server).



If there are problems starting a protocol's connection managers, a warning message is displayed and the error messages are logged in the *db2diag.log* file.

Refer to the *Troubleshooting Guide* for information on the *db2diag.log* file.

At the *client*:

- 1 If you are using Direct Addressing, check that the node was cataloged with a value of * for *FILESERVER*, and the correct IPX/SPX internetwork address value for the *OBJECTNAME* parameter.
- 2 If you are using File Serving Addressing, check that the *FILESERVER* and *OBJECTNAME* parameters, used to catalog the node, match those configured on the server.
- 3 The node name (*node_name*), specified in the database directory, points to the correct entry in the node directory.

- 4 The database was cataloged properly, using the *server's* database alias (*database_alias*) that was cataloged when the database was created on the server, as the database name (*database_name*) on the client.

After you verify these items, refer to the *Troubleshooting Guide* if the connection still fails.

Configuring APPC on the Client

This section describes how to configure a client workstation to communicate with a DB2 server using the APPC communication protocol, and assumes that APPC is functional on the client and DB2 server workstations. See “Software Requirements” on page 5 for the communication protocol requirements for your platform. See “Possible Client-to-Server Connectivity Scenarios” on page 9 for the supported communication protocols for your particular client and server.



You need to ensure that the DB2 server that you are attempting to establish communications with supports APPC clients. APPC client communications are supported by the following DB2 servers:

- AIX
- OS/2
- Solaris
- Windows NT

For more information, see “Software Requirements” on page 5.

The following steps are required to set up a client to use APPC communications:

- 1 Identify and record parameter values.
- 2 Configure the client:
 - a Update the APPC profiles.
 - b Catalog the APPC node.
 - c Catalog the database.
- 3 Test the connection between the client and server.

Step 1. Identify and Record Parameter Values

As you proceed through the configuration steps, complete the *Your Value* column in the following table. You can fill in some of the values before you start configuring this protocol.

Parameter	Description	Sample Value	Your Value
Symbolic destination name (<i>sym_dest_name</i>)	Specifies the name of the CPI-C profile defined in APPC on the client. This profile contains information that allows the client to communicate with the server. This profile name is used by DB2 as a link into the APPC communications subsystem on your client. Obtain this name from your APPC System Administrator or your APPC management tools.	NYSERVER	
Security type (<i>security_type</i>)	Specifies the type of APPC security to be used for the connection. It is recommended that you use the security level <i>none</i> for the APPC connection. If you do not specify this parameter in "B. Catalog the APPC Node" on page 111, the default <i>program</i> is used.	none	
Node name (<i>node_name</i>)	A local alias, or nickname, that describes the node where the database resides. You can choose any name you want, however, all node name values within your local node directory must be unique.	db2node	

Step 2. Configure the Client

The following steps describe how to configure this protocol on the client. Replace the sample values with your worksheet values.

A. Update the APPC Profiles: You must update the APPC communication profiles before your client can access a DB2 server. Use the provided worksheet to help you track values when updating the APPC profiles for your particular client.

<i>Table 20. APPC Values Required at the Client</i>			
No.	Description	Sample Value	Your Value
Network Connection Information at the DB2 Client			
13	Partner node name	NYX1	
14	Type	Token-ring	
15	Connection name	LINKNYX1	
16	LAN Destination Address	4 5AA8B532	
17	Partner network ID	SPIFNET	
18	Partner LU name	NYX1GW 1	
Network Elements at the DB2 Client			
19	Network name	SPIFNET	
2	Local node name (same as the Control Point name)	NYX2	
21	Local node ID	5D2751	
22	Partner LU fully-qualified LU name	SPIFNET.NYX1GW 1	
23	Symbolic destination name	NYSERVER	
24	Partner TP name	NYSERVER	
25	Security type	NONE	
26	Mode name	IBMRDB	
27	Local LU name	NYX2GW 1	
Note: Numbers 1 to 12 are reserved for server usage.			



Go to the section that describes that required steps to update the APPC profiles for the client that you want to use to communicate with you DB2 server.

“Updating the APPC Profiles for an AIX Client”

“Updating the APPC Profiles for an HP-UX Client” on page 90

“Updating the APPC Profiles for an OS/2 Client” on page 96

“Updating the APPC Profiles for a SINIX Client” on page 100

“Updating the APPC Profiles for a Solaris Client” on page 102

“Updating IBM Communications Server APPC Profiles for a Windows NT Client” on page 104

“Updating MS SNA Server APPC Profiles for a Windows NT Client” on page 108

Updating the APPC Profiles for an AIX Client: You need to customize the APPC configuration for the client. The following list gives the APPC profiles that you will need to update on your client workstation before you can access a DB2 server using APPC.

Control Point Profile

Token Ring SNA DLC Profile

Token Ring Link Station Profile

LU 6.2 Mode Profile

Partner LU 6.2 Location Profile
LU 6.2 Side Information Profile

Samples of the profiles are provided. A field that has been changed with the default value has been marked with a number in parentheses in each of the sample profiles.

A worksheet is provided on Table 20 on page 87. Each of the values marked with numbers in parentheses in the following samples are included in the corresponding worksheet. Use the worksheets to record the values of parameters that you need to provide when you create or modify a profile.

Configure your DB2 Connect system as follows:

- 1** Log on to the AIX machine as *root*.
- 2** On the command line, enter:

```
smit
```
- 3** Select panels in the following order:
 - a** Communications Applications and Services
 - b** SNA Server/6000
 - c** Configure SNA Profiles
 - d** Advanced Configuration.

The next several steps begin from the Advanced Configuration panel.

- 4** Update the Control Point Profile as follows:
 - a** Select the Control Point panel.
 - b** Select the Change/Show a Profile panel.
 - c** Between the square brackets [], enter the value of the XID node ID 21 .
The control point name was probably set when the SNA Server for AIX was installed. 2 .

To minimize confusion, use the same value for the control point alias as for the control point name.

For example, the panel could look like this:

Change / Show Control Point Profile

```
Profile name                node_cp
XID node ID                 [ 5D2751 ]    21
Network name                [ SPIFNET ]    19
Control Point (CP) name     [ NYX ]        2
Control Point alias         [ NYX2 ]        2
Control Point type          appn_end_node
Maximum number of cached routing trees [ 5 ]
Maximum number of nodes in the TRS database [ 5 ]
Route addition resistance   [ 128 ]
```

5 Check the Token Ring SNA DLC Profile as follows:

- a** Select the Links panel.
- b** Select the Token Ring panel.
- c** Select the Token Ring SNA DLC panel.
- d** Select the Change/Show a Profile panel.
- e** The profile name and data link device name were probably set when SNA Server for AIX was installed. If you have a reason to change these values, you can do so. If no values appear in the profile, talk to your LAN administrator.

Write down the profile name.

For example, the profile might look like this:

Add Token Ring SNA DLC Profile

```
Profile name                [ tok . 1 ]
Data link device name       [ tok ]
Force disconnect time-out (1-6 seconds) [ 12 ]
User-defined maximum I-Field size?     no
    If yes, Max. I-Field size (265-3 729) [ 3 729 ]
```

- f** If you will use APPC for connections from remote clients, make sure that the profile has the following value:

```
Dynamic link stations supported?      yes
```

6 Create a Mode Profile as follows:

- a** Select the Sessions panel.
- b** Select the LU 6.2 panel.
- c** Select the LU 6.2 Mode panel.
- d** Select the Add a Profile panel.

- e Define your mode profile. For profile name and mode name, use the value that you wrote in item 26 of the worksheet. On the other lines, specify values that match the mode profile defined on your DRDA server systems. If you are using the mode IBMRDB, specify the following values:

```

Profile name                               [IBMRDB] 26
Mode name                                  [IBMRDB] 26
Maximum number of sessions (1-5 )         [3 ]
Minimum contention winners ( -5 )         [15]
Minimum contention losers ( -5 )         [15]
Auto activate limit ( -5 )                [ ]
Upper bound for adaptive receive pacing window [16]
Receive pacing window ( -63)             [8]
Maximum RU size (128,...,32768: multiples of 32) [4 96]
Minimum RU size (128,...,32768: multiples of 32) [1 24]
Class of Service (COS) name                [#CONNECT]

```

7 Create a Local LU profile as follow:

- a Select the Sessions panel.
- b Select the LU 6.2 panel.
- c Select the LU 6.2 Local LU panel.
- d Select the Add a Profile panel.
- e Update the three lines with the following value: Use 27 for the all three lines. For example, the file could look like this:

Add LU 6.2 Local LU Profile

```

Profile name                               [NYX1GW 2] 27
Local LU name                              [NYX1GW 2] 27
Local LU alias                             [NYX1GW 2] 27
Local LU is dependent?                     no
  If yes,
    Local LU address (1-255)                [ ]
    System services control point
      (SSCP) ID ( , -65535)                 [ ]
    Link Station Profile name               [ ]
    Conversation Security Access List Profile name [ ]

```



After you have finished updating your APPC client profile, go to "B. Catalog the APPC Node" on page 111.

Updating the APPC Profiles for an HP-UX Client: You need to customize the APPC configuration for the client. The following list gives the APPC profiles that you will need to update on your client workstation before you can access a DB2 server using APPC.

Link Service
Connection Service
Local Node Service
APPC Mode
APPC Remote LU
APPC Local LU
CPI-C Symbolic Destination Name

To use APPC, SNAplus must be installed and configured on the client workstation, and you must create an entry in the node directory.

Configuring SNAplus for Client Workstation

Before configuring the connection between the DB2 Client Application Enabler for HP-UX workstation and the server, you will have to collect some information from the network administrator.

This section will give you an example of how to use those values to configure the SNA parameters on the DB2 Client Application Enabler for HP-UX workstation. This is only an example; you should be familiar with SNA and your own communications configuration to successfully configure SNAplus for DB2 Client Application Enabler for HP-UX.

Refer to the HP-UX SNAplus Installation Guide, the HP-UX SNAplusLink Administrator's Guide, the HP-UX SNAplus API Administrator's Guide, and the HP-UX SNAplusLink Diagnostics Guide for more information about:

Installing SNAplus
SNA terms and concepts
Configuring the SNA parameters
Problem resolution.

Description of the Example The following assumptions were made when configuring this system:

The basic installation of the SNAplus package has been completed, including the execution of the INSTALL_SNAP script.
DB2 Client Application Enabler for HP-UX has been installed.
The user is logged on as root.

Notes:

1. The values provided are for the example only; you must substitute values that are appropriate to your operating system.
2. Get the correct values from your network administrator.
3. Use the worksheet in Table 20 on page 87 to assist you in gathering the information.

The names and addresses in this example are:

CP Name	NYX2	2
Remote LINK ADDRESS	4 5AA8B532	16
XID Node ID	5D2751	21
Network Name	SPIFNET	19
Local LU Name	NYX2GW 1	27
Remote LU Name	NYX1GW 1	18
Remote Transaction Program Name	NYSERVER	24
Mode Name	IBMRDB	26

Configuring SNAplus

To configure SNAplus for DB2 Connect, log on as root and use either the **/opt/sna/bin/snapconfig** program or the **/opt/sna/bin/X11/xsnapconfig** program. Information about these programs can be found in the HP-UX documentation. The following steps describe how to use these programs to configure SNAplus for DB2 Connect.

Note: These steps contain suggested values for configuration parameters. For other parameters, use the SNAplus default values.

- 1 Open the configuration file:
 - a. Select **File->Open**
 - b. Select **Open running configuration file**, then click on **OK**.
- 2 Select the links:
 - a. Select **Services->Links**
 - b. Select the option marked **NEW** associated with the type of link you want to use: Token Ring (LAN), SDLC, or QLLC, then select **Add**. Enter a name for this link, for example, TR1.

For Token Ring links, use the default Device Name and Port Number. For SDLC and QLLC links, consult with your network administrator for the appropriate values. When you have filled in the information, or chosen to use the default values, click on **OK** followed by **Done** to get back to the main screen.
- 3 Select the connections:
 - a. Select **Services->Connections**. Select the option marked **NEW** corresponding to the value selected in the previous step. For example, for Token Ring, select **NEW** next to **LAN**. Enter a name for this connection. You may optionally enter a description.
 - b. Set **Remote End** to **Host System** and **Activation** to **On Demand**. If you are connecting to a machine other than a host system, select **Peer Node** instead of **Host System**.
 - c. Enter the first three characters of the XID Node ID in the first **Node ID to Send** field. For example, enter 5D.

- d. Enter the SNA Physical Unit Number associated with your system in the second **Node ID to Send** field. For this example, enter 2751 .
- e. Leave the "Node ID to Receive" fields blank.
- f. Enter your SNA Network Name in the first field of the **Full Control Point Name**. For this example, enter SPIFNET.
- g. Enter the Physical Unit Name associated with your HP workstation in the second field of the **Full Control Point Name**. For this example, enter NYX2.
- h. Move the link you configured in the previous step from the **Other Links** box to the **Links Usable** box.
- i. Select **Parameters** and enter any necessary connection information. For Token Ring, enter the SNA Destination Address assigned for the DB2RA server system in the **Remote Network Address** field and leave the other fields with their default values. For this example, use 4 115275 9. For SDLC and QLLC connections, additional information is required. Ask your network administrator for the required values.
- j. Click on **OK** to return to the connection configuration panel, then click on **OK** again, then **DONE**.

4 Select the local nodes:

- a. Select **Services->Local Nodes**.
- b. Select **NEW**.
- c. Enter a name for the node and click on **OK**. You may optionally enter a description for the node, but you must enter your network name in the **Name of Network** field. For this example, use SPIFNET for this value.
- d. Move the connection you created in the previous step from the **Available Connections** box to the **Connections Used** box using the **Move** button.
- e. Click on **OK**, then **DONE**.

5 Select the modes:

- a. Select **APPC->Modes**.
- b. Select **NEW**.
- c. Enter IBMRDB in the **Mode Name** field, and optionally enter a description.
- d. In the **Connection** box, highlight the connection you configured above. Make sure that **High priority mode** is not selected. The following configuration values are suggested:

Session limit = 030

Minimum contention winner limit = 015
Partner minimum contention winner limit = 015
Automatic activation limit = 001
Max send RU length = 4096
Max receive RU length = 4096
Send and Receive Pacing Counts = 07

Other values can be left as defaults.

e. Click on **OK**, then **DONE**.

6 Select Remote LUs:

- a. Select **APPC->Remote LUs**.
- b. Select **NEW**.
- c. Enter the Remote LU Name as the LU alias, and click on **OK**.
For this example, use `NYX1GW 1`. You may optionally enter a description of the remote LU.
- d. The **Full network name** field is made up of the SNA network name in the first field and the Remote LU Name associated with the DB2RA server database in the second field. For this example, use `SPIFNET.NYX1`.
- e. Select **Supports parallel sessions** and **No session level security**.
- f. Click on **OK**, then **DONE**.

7 Select Local LU:

- a. From the **APPC** menu, select **Local LUs**.
- b. In the **Local Nodes** box, select the node you configured above.
- c. In the **Configured LUs** box, select **NEW**.
- d. For the APPC LU alias, enter your Independent LU Name; for this example, it is `NYX2GW 1`. You may optionally enter a description for the Local LU.
- e. Make sure that the **Full Network Name** fields contain your Network Name and your Independent LU Name. The LU number should be `1` for an independent LU.
- f. Change the **Session Limit** parameter to `3`.
- g. Make sure that you have selected the radio buttons to indicate that this LU is in the pool of default LUs.
- h. Move the Remote LU you configured above from the **Other Remote LUs** box to the **Partner LUs** box.
- i. Select **Modes**.
- j. Move the `IBMRDB` mode from the **Other Modes** box to the **Associated Modes** box.

k. Click on **OK**, then **OK** again, then **DONE**.

8 Select the Symbolic Destination Name:

- a. Select **APPC->Sym Dest Name**.
- b. Select **NEW**.
- c. Enter the Symbolic Destination Name you want to associate with the DRDA server database and click on **OK**. You may optionally enter a description. This name must match the `sym_dest_name` value you use to catalog the APPC node.
- d. For the **Partner LU type**, make sure that **Alias** is selected.
- e. In the **Partner LU** box, ensure that the **Remote LU** you configured above is highlighted.
- f. In the **Mode Name** box, highlight `IBMRDB`.
- g. For the **Partner TP name**, use `NYSERVER`. In this example it is an Application TP.
- h. For the **Partner LU type**, make sure that `Alias` is selected.
- i. Click on **Security** and choose none (This does not mean that you will have no security; you can specify the security type later, in the node directory).
- j. When done, click on **OK**, then **OK**, then **DONE**.

9 Saving your configuration:

- a. From the **File** menu, select **Save**.
- b. Select **Update running configuration file**, then click on **OK**. This saves your new configuration as the default, running configuration.

Notes:

1. You may need to stop and restart SNA for your configuration changes to take effect.
2. You may also need to contact your database or network administrators to have your Local LU names added to the appropriate tables to access the DRDA server database.

Starting the SNAplus Subsystem

Use `/opt/sna/bin/snapstart` to start the SNAplus subsystem. You can use `/opt/sna/bin/snapstop` to stop the SNAplus subsystem first, if required.

To start the SNAplus connections you can use either the `/opt/sna/bin/snapmanage` program, or the `/opt/sna/bin/X11/xsnapmanage` program.

- 1** From the **Manage** menu, select **Services**.

- 2 Highlight the node you configured in the previous section, and select **start**.
- 3 Highlight the computer you want to start it on, then click on **OK**.
- 4 Highlight the connection you configured in the previous section, and select **start**.
- 5 Highlight the computer you want to start it on, then click on **OK**.

Updating the APPC Profiles for an OS/2 Client: You need to customize the APPC configuration for the client. The following list gives the APPC profiles that you will need to update on your client workstation before you can access a DB2 server using APPC.

- Local Node Characteristic Profile
- Token Ring or Other LAN Types DLC Adapter Parameters Profile
- Adapter List Profile
- Connection to a Peer Node Profile
- Partner LUs Profile
- Mode Definition Profile
- CPI Communications Side Information Profile


Begin to configure your DB2 Connect system with the following steps:

- 1 Double-click on the **Communications Server** icon.
- 2 Double-click on the **Communications Manager Setup** icon.
- 3 On the Communications Server Setup panel, click on the **Setup** push button.
- 4 On the Open Configuration panel:
 - a Specify the name of an existing configuration file if you have one, or provide a name for a new file to be created.
 - b Click on the **OK** push button to proceed to the Communications Server Configuration Definition panel.
- 5 The Configuration Definition panel opens.
 - a Select the **Commonly used definitions** radio button.
 - b In the Communications Definitions window, select the protocol that you want to use. These examples use APPC APIs over Token-Ring.
 - c Click on the **Configure** push button to proceed.

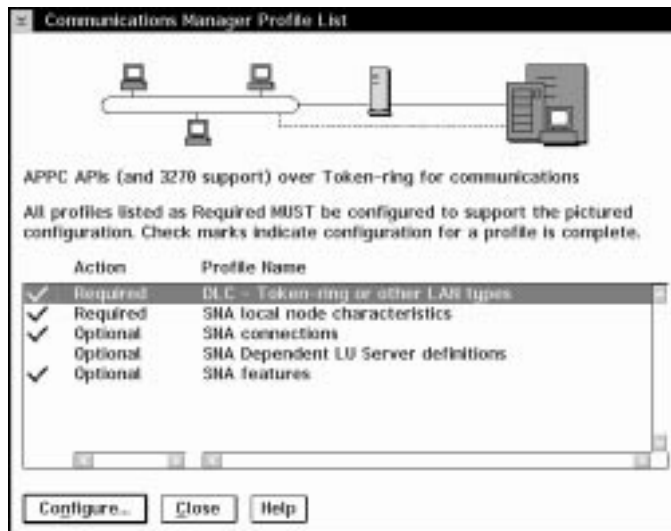
If this is a new configuration file, the next panel displayed will be the APPC APIs over Token-Ring panel. Otherwise the next panel displayed will be the Communications Server Profile List panel.
- 6 The APPC APIs over Token-Ring panel opens.
 - a Enter your values for:

Network ID 19
Local node name 2

- b** Click on the **End node** push button that your network administrator advises you to use.

 You can select either the **End node - to a network node server** radio button or the **End node - no network node server** radio button. A network node server is used when many users are routed through the same connection. The example here presumes no network node server is used.

- c** Click on the **Advanced** push button. The Communication Manager Profile List window opens.



Subsequent steps begin from this panel. You will return to this panel when each step is complete.

LAN DLC Profile

From the Communications Server Profile List panel, prepare a LAN DLC profile as follows:

- 1** Select **DLC - Token ring or other LAN types**->**Configure**.
- 2** Enter your value for **Network ID** or **C&SM LAN ID**.
- 3** Check that the other values are appropriate for your environment. The values in the example are default values.
- 4** Click on **OK** to return to the Communications Server Profile List panel.

Local Node Characteristic Profile

Update SNA Local Node Characteristics as follows:

- 1 On the Communications Server Profile List panel, select **SNA local node characteristics->Configure**. The Local Node Characteristic panel opens.
- 2 Enter your value for **Network ID** 19 .
- 3 The **Local node name** was probably set when Communications Server was installed. If you are not sure, consult your local network administrator.
- 4 Enter your value for **Local node ID (hex)** 21 .



The first part should be filled in for you already when you display the profile. You only need to complete the second part.

- 5 Click on the **Options** push button. The Local Node Options panel opens.
 - a Verify that the **Activate Attach Manager at startup** check box is selected.
 - b Click on **OK** to return to the Local Node Characteristics panel.
- 6 Click on **OK** to return to the Communications Server Profile List panel.

Adapter List Profile

Prepare the SNA connection profiles as follows:

- 1 From the Communications Server Profile List panel, select **SNA Connections->Configure**.
- 2 On the Connections List panel for **Partner Type**, select the **To peer node** radio button and click on the **Create** push button.
- 3 The Adapter List window opens.
- 4 Select the **Token-ring, or other LAN types** adapter type, and specify the same **adapter number** that you specified in the DLC profile.
- 5 Click on the **Continue** push button to proceed to the Connection to a Peer Node panel or the Connection to a Host panel.

Connection to a Peer Node Profile

The Connection to a Peer Node or Connection to a Host Node windows opens. At the Connection to a Peer Node panel or the Connection to Host panel:

- 1 Specify the **Link name** value that you wrote in item 15 of the worksheet.
- 2 On the Connection to Host panel, select **Additional parameters** push button and change the **Local PU name** field to your value for 2 .
- 3 Change the node ID field to your values for 21 .

- 4 Change **LAN destination address** field to your value for 16 .
- 5 Change **Partner network ID** field to your value for 17 .
- 6 Change **Partner node name** field to your value for 18 .
- 7 Select **Define Partner LUs** push button on either the Connection to a Peer Node panel or the Connection to Host panel to proceed.

Partner LUs Profile

Create a Partner LUs Profile as follows:

- 1 Enter your value for **Network ID** 17 .
- 2 Enter your value for **LU name** and **Alias** (use 18).
- 3 Click on the **Add** push button to add the partner LU profile to the connection profile.
- 4 Click on **OK** to return to the previous panel.
- 5 On the Connection to Host panel, click on the **Additional Parameters** push button. The Partner LU window opens

Mode Definition Profile

From the SNA Features List panel, prepare a Mode Definition as follows:

- 1 Select **Modes->Create**.
- 2 Define your mode profile as follows:
 - a Enter your value for **mode name** 26 .
 - b On the other lines, you can either specify values that match the mode profile defined on your DB2RA server systems, or tune the parameters.
- 3 Click on **OK** to finish the creation of the mode and to return to the SNA Features List panel.

CPI Communications Side Information Profile

On the SNA Features List panel, select **CPI Communications Side Information->Create**. The CPI Communications Side Information opens.

Complete the CPI Communications Side Information panel as follows:

- 1 Enter your value for **Symbolic destination name** 23 .
- 2 Select the **Alias** radio button. Click on the **Alias** drop down box to view a list of all defined aliases and choose the alias that corresponds to item 18 .
- 3 In the **Partner TP** box, specify the remote Transaction Program (TP) name 24 .
Select the **None** radio button as the **Security type**.



This does not mean that you will have no security, you will specify the security type later when you update the DB2 directories.

- 4 Enter your value for the **Mode name** field 26 .
- 5 Click on **OK** to save the CPI side information profile and return to the SNA Features List panel.
- 6 Click on **Close** to return to the Communications Server Profile List panel.



After you have finished updating your APPC client profile, go to "B. Catalog the APPC Node" on page 111.

Updating the APPC Profiles for a SINIX Client: You need to customize the APPC configuration for the client. The following list gives the APPC profiles that you will need to update on your client workstation before you can access a DB2 server using APPC.

- Link Profile
- Station Profile
- Local LU Profile
- Remote LU Profile
- Mode Profile
- Symbolic Destination Profile

The following section provides a working example of the SNA configuration on a DB2 for SINIX client to support communications with a server workstation.

Notes:

1. The values provided are for the example only; you must substitute values that are appropriate for your operating environment.
2. Get the correct values from your network administrator.
3. A worksheet is available in Table 20 on page 87 to assist you in gathering information.

There is one configuration file that is read by the TRANSIT Server on startup. This configuration file has a default name of `/opt/lib/transit/KOGS/conf.nuc`. The configuration file defines the SNA resources and operating characteristics through a set of configuration directives.

The following example is a configuration file that contains the configuration directives.

```

XLINK   tok . 1,                               //Link name
        ACT       = AUTO,                       //Activation
        TYP       = LAN,                        //Connection type
        XID       = 5D27946,                   //Node ID

```

```

CPNAME      = CAIBMOML.OMXTXL,           //Control point
CONFSTR     = /opt/lib/llc2/conf.str,    //Confstr
DEVICE      = tr ,                       //Device
SSAP        = 4                          //SSAP
XPU         JAGUAR,                      //Station name
           TYP      = PEER,              //Type
           CONNECT  = AUTO,              //Connection setup
           DISCNT   = AUTO,              //Connection shutdown
           LINK     = tok . 1,           //Link name
           NVSCONNECT = DYNAMIC,         //NVS connect
           MAXDATA  = 265,               //Segment size
           XID      = 5D27946,           //Node ID
           CPNAME   = CAIBMOML.OMXTP9 9, //Control point
           ROLE     = NEG,                //Station role
           PAUSE    = 3,                  //Pause
           RETRIES  = 1 ,                 //Retries
           DMAC     = 1 5AA8E2 8,        //DMAC
           DSAP     = 4,                  //DSAP
           RWINDOW  = 7                   //Window
XLU         OMXTXL,                      //Name
           TYP      = 6,                  //LU type
           PUCONNECT = APHSTART,          //Connection
           CTYP     = PUBLIC,             //Coordinate Type
           SESS-LMT = 3,                  //Session limit
           SESS-CTR = IND,                //Session control
           NETNAME  = CAIBMOML.OMXTXL,    //Network name
           PAIR     = OMXTP9 9 IBMRDB     //LU pair
XRLU        OMXTP9 9,                    //Remote LU name
           NETNAME  = CAIBMOML.OMXTP9 9, //Network name
           PU       = JAGUAR              //Station name
XMODE       IBMRDB,                      //Mode name
           SESS-MAX = 2 ,                 //Session Max
           SESS-LOS = 1 ,                 //Session Loser
           SESS-WIN = 1 ,                 //Session Winner
           SESS-AUTO = 5,                 //Session Auto
           SRU-MAX  = 87,                 //RU size Send
           RRU-MAX  = 87,                 //RU size Recv
           PAC-SEND = 4,                  //Pacing Send
           PAC-RCV  = 4,                  //Pacing Recv
           IDLETIME =                     //Idle time
XSYMDEST    CPICSD,                      //SD name
           RLU     = OMXTP9 9,            //RLU name
           MODE    = IBMRDB,              //Mode name
           TP      = CPICRCV,             //TP name
           TP-TYP  = USER,                //Type
           SEC-TYP = NONE                  //Security
XEND

```



After you have finished updating your APPC client profile, go to “B. Catalog the APPC Node” on page 111.

Updating the APPC Profiles for a Solaris Client: You need to customize the APPC configuration for the client. The following list gives the APPC profiles that you will need to update on your client workstation before you can access a DB2 server using APPC.

- PARTNER LU Profile
- MODE Profile
- CP Profile
- TRLINE Profile
- DLC Profile
- LU Profile

You also have to create a file containing the CPI-C side information representing the Symbolic Destination Name.

You can use or your favorite editor to create the configuration files.

CPIC Side File: This file must be placed in the application's path for a standalone DB2 Connect for Solaris system. However, for DB2 agents to be able to access it on DB2 Connect for Solaris, the file should be in either:

| INSTHOME/sqllib/adm or

| INSTHOME/sqllib/bin

Note: The name of the CPIC side file must be the same as the Symbolic destination name specified in the DB2 node directory on the DB2 Connect for Solaris system.

Sample CPIC side file

The file name must be the same as the Symbolic destination name specified in the DB2 node directory for the server.

```
# CPIC Side File information
#
PTNR_LU_NAME=NYX1GN 1
MODE_NAME=IBMRDB
TP_NAME= NYSERVER
SECURITY=NONE
```

This file is called `sunpu2.config`, and it must be placed in `/opt/SUNWpu21`, or the directory where SunLink SNA PU 2.1 Server is installed.

```

// SunLink SunLU6.2/SunPU2.1 SNA Server Sample Configuration
// Token Ring Peer-to-Peer System A @(#)sunlu62.a.tr
//
// The physical connection is a Token Ring interface adapter.

CP      NAME=NYX2                      // Local name (8 char max)
        NQ_CP_NAME=SPIFNET.NYX12      // Network Qualified Name
        ;

TRLINE  NAME=MAC1                      // SunLink specific name
        SOURCE_ADDRESS=x'4  115275 9' // sysA_mac_addr for Sun machine
        ;

DLC     NAME=HOSTLINK                  // User defined name (8 char max)
        LINK_NAME=MAC1                // Line name this station is on
        LCLLSAP=x' 4'                 // Local Link Service Access Point
        RMTLSAP=x' 4'                 // Remove Link Service Access Point
        RMTMACADDR=x'4  5AA8B532'    // sysB_mac_addr
        TERMID=x' 71275 9'           // XID negotiation
        ;

LU      NAME=NYX2GW 1                 // Local name (8 char max)
        NQ_LU_NAME=SPIFNET.NYX2GW 1  // Network Qualified Name
        SESS_LMT=5                     // Max LU sessions
        LUTYPE=6.2
        ;

PTNR_LU NAME=NYM2DB2                  // Partner LU name(8 char max)
        LOC_LU_NAME=NYX1GW 1          // Associated Local LU
        NQ_LU_NAME=SPIFNET.NYM2DB2   // Network Qualified Name
        ;

MODE    NAME=IBMRDB                   // Mode Name (8 char max)
        DLC_NAME=HOSTLINK             // Associated DLC
        PTNR_LU_NAME=                 // Associated Partner LU
        LCL_MAX_SESS_LMT=3            // Max Session Limit
        MIN_CW_SESS=15                // Min Conwinners
        MIN_CL_SESS=15                // Min Conlosers
        ;

```

Figure 1. Sample Solaris PU 2.1 SNA Client Configuration File



After you have finished updating your APPC client profile, go to “B. Catalog the APPC Node” on page 111.

Updating IBM Communications Server APPC Profiles for a Windows NT Client: You may need to create an entirely new configuration file or customize an existing one. This section assumes that you are creating a new configuration file, but if you are customizing an existing one the steps are similar. You need to perform the following configuration steps:

- Configure Node
- Configure Devices
- Configure Connections
- Configure Transaction Program
- Configure Partner LU
- Configure Mode
- Configure Local LU 6.2
- Configure CPI-C Side Information

To begin, perform the following steps:

- 1** Click on the **Start** push button, and select **Programs->IBM Communications Server->SNA Node Configuration**.
- 2** Select **File->New** to create a new configuration file, as used in this example, or you can select **File->Open** to select, open, and modify an existing configuration profile.
- 3** Select **Scenarios->CPI-C, APPC, or 5250 Emulation**, or you can select **Scenarios->Advanced** when creating a new configuration file or modifying an existing one.

Configure Node

To configure the node, do the following (this example is for an End Node):

- 1** Select the **Configure Node** configuration option, and click on **New**. The Define The Node notebook appears.
- 2** Complete the **Basic** notebook page as follows. Use the values that you recorded on your worksheet, for example:
 - Fully qualified CP name** SPIFNET.NYX2 19 and 2
 - CP alias** We suggest you use NYX2, the same value as 2
 - Local Node ID** 5D for the Block ID (the first three digits of 21), and 2751 for the Physical Unit ID (the last five digits of 21).
 - Node Type** End Node was specified in this example.
- 3** Click on **OK** to save the node definition and return to the Communications Server SNA Node Configuration window.

Configure Devices

To configure devices, do the following (this example used a LAN DLC):

- 1 Select the **Configure Devices** configuration option.
- 2 Select the appropriate DLC for your environment. In this example, **LAN** was used.
- 3 Click on **New**. The Define a LAN Device notebook appears.
- 4 All the fields on the **Basic** notebook page were allowed to default in this example. The same was true for the **Advanced** and **Performance** notebook pages
- 5 Click on **OK** to save the node definition and return to the Communications Server SNA Node Configuration window.

Configure Connections

To configure the LAN connection, do the following:

- 1 Select the **Configure Connections** configuration option, with your DLC also selected (**LAN** in this example).
- 2 Click on **New**. The Define a LAN Connection notebook appears.
- 3 Complete this notebook page as follows. Use the values that you recorded on your worksheet, for example:

Link Station Name	Use any unique and meaningful value, such as LINKNYX1.
Device name	Use the value that already appears. It was selected when you configured the device in the previous step.
Destination Address	4 5AA8B532 16
Remote SAP	Was left to default to 04 in this example.

- 4 Select the **Security** notebook tab, and set the **Adjacent CP name** to SPIFNET.NYX1, corresponding to worksheet items 17 and 13 .
- 5 Click on **OK** to save the connection definition and return to the Communications Server SNA Node Configuration window.

Configure Transaction Program

No action is required.

Configure Partner LU

To configure the partner LU, do the following:

- 1 Select the **Configure Partner LU** configuration option, and click on **New**. The Define a Partner LU 6.2 notebook appears.
- 2 Complete the **Basic** notebook page as follows. Use the values that you recorded on your worksheet, for example:

Partner LU Name	SPIFNET.NYX1GW 1 17 and 18
------------------------	----------------------------

Partner LU Alias We suggest you use NYX1GW 1, the same value as 18

Fully Qualified CP Name SPIFNET.NYX1 17 and 13

- 3 Click on **OK** to save the partner LU definition and return to the Communications Server SNA Node Configuration window.

Configure Mode

To configure the mode, do the following:

- 1 Select the **Configure Mode** configuration option, and click on **New**. The Define a Mode notebook appears.
- 2 Complete the **Basic** notebook page as follows. Use the values that you recorded on your worksheet, for example:

Mode name IBMRDB 26

PLU session mode limit Was allowed to default (32).

Minimum contention winner sessions Was allowed to default (16).

- 3 Select the **Advanced** notebook page. Usually all of these values can be allowed to default. However, you may need to verify or even change the Maximum PIU size for this workstation. For example, if the connection to the server will use an Ethernet adapter, the maximum Ethernet frame size is 1536 bytes, and this would require a maximum RU size no greater than 1529 bytes.
- 4 Click on **OK** to save the mode definition and return to the Communications Server SNA Node Configuration window.

Configure Local LU 6.2

To configure the local LU, do the following:

- 1 Select the **Configure Local LU 6.2** configuration option, and click on **New**. The Define a Local LU 6.2 notebook appears.
- 2 Complete the **Basic** notebook page as follows. Use the values that you recorded on your worksheet, for example:

Local LU Name NYX2GW 1 27

Local LU alias We suggest you use the same value as for item 27

LU session limit Leave this set to 0 (permits up to 64K sessions)

- 3 Click on **OK** to save the node definition and return to the Communications Server SNA Node Configuration window.

Configure CPI-C Side Information

To configure the CPI-C Side Information, do the following:

- 1 Select the **Configure CPI-C Side Information** configuration option, and click on **New**. The Define CPI-C Side Information notebook appears.
- 2 Complete the **Basic** notebook page as follows. Use the values that you recorded on your worksheet, for example:

Symbolic Destination Name NYSERVER 23

Mode name Select IBMRDB, the same value as 26

Use Partner LU Name or Use Partner LU Alias

Click the radio button that selects which of these you will use. Either specify the fully-qualified partner LU name SPIFNET.NYX1GW 1 (the same value as worksheet item 22), or select the Partner LU alias that you defined earlier (the same value as worksheet item 18 in this example).

TP Name NYSERVER 24

- 3 Complete the **Security** notebook page as follows. Use the values that you recorded on your worksheet, for example:
Conversation Security Specify NONE 25
- 4 Click on **OK** to save the CPI-C Side Information definition and return to the Communications Server SNA Node Configuration window.

To complete the configuration file, from the Communications Server SNA Node Configuration window:

- 1 Select **File->Save As** and specify the configuration file name of your choice, for example `nyserver.acg`. Indicate whether this configuration file is to become the default for this workstation.
- 2 Close the Communications Server SNA Node Configuration window.
- 3 Use **Start->Programs->IBM Communications Server->SNA Node Operations** to open the **Communications Server Node Operation** window.
- 4 Select **Operations->Start Node->Open** to start the node with the new configuration file.



After you have finished updating your APPC client profile, go to “B. Catalog the APPC Node” on page 111.

Updating MS SNA Server APPC Profiles for a Windows NT Client: You need to customize the APPC configuration for the client. The following list gives the APPC profiles that you will need to update on your client workstation before you can access a DB2 server using APPC.

- Server Properties
- Local LU Properties
- Mode Properties
- Connection Properties
- Remote LU Properties
- CPI-C Symbolic Destination Name Properties

To begin, perform the following steps:

- 1 Click on the **Start** push button, and select **Program->SNA**
- 2 Double-click on the **SNA Server Admin** icon. The SNA Server Admin window opens.



By default, the Servers and Connections window opens. If it does not, double-click on the **Servers and Connections** icon.

Server Properties

To specify parameters for an SNA Server that will be used for LU 6.2 (APPC), do the following:

- 1 In the Servers and Connections window, select the server (named SERVER in the example).
- 2 Select **Services->Properties**. The Server Properties window opens.
- 3 Enter the values you filled in on Table 20 on page 87 as follows:

Network Name	19
Control Point Name	2

Local LU Properties

To assign a local APPC LU to the Server, perform the following steps:

- 1 In the Servers and Connections window, select the server.
- 2 Select **Services->Assign LUs**. The Insert LU window opens.
- 3 In the Insert LU window, select **APPC (Local)**.
- 4 Click on **OK**. The New APPC LU Properties window opens.
- 5 Make sure that the **Independent** radio button is selected.
- 6 Enter the values you entered in on Table 20 on page 87 as follows:

LU Alias	27
Network Name	19
LU Name	27

- 7** Select the **Enable Automatic Partnering** check box, if it is not already selected.
- 8** Select the **Member of Default Outgoing Local APPC LU Pool** check box, if it is not already selected.
- 9** Click on **OK**. The new LU is added to the window.

Mode Properties

You can use an IBM-defined mode, or add a new mode that is appropriate for the application. Ensure that the values match between your client and server.

To configure a mode for the APPC LU, perform the following steps:

- 1** In the Servers and Connections window, select the local LU with which the mode will be associated.
- 2** Click on **Services->Properties**.
Based on whether the LU is local or remote, either the APPC LU Properties window or the New APPC LU Properties window opens.
- 3** Click on the **Partners...** push button. The LU 6.2 Partner LUs window opens.
- 4** Click on the **Modes...** push button. The APPC Mode Properties window opens.
- 5** Use the following suggested value:
Mode Name IBMRDB 26
- 6** Select the **Enable Automatic Partnering** check box, if it is not already selected.
- 7** Ensure that the **High Priority Mode** check box is *not* selected; if it is, clear the check box.



The rest of the values in our example are arbitrary. The values for items such as **Parallel Session Limit** and **Minimum Contention Winner Limit** are for tuning purposes. See the online help for SNA Server for guidance on setting these values.

- 8** Click on the **Add** push button to add the mode.
- 9** Click on the **Close** push buttons until you return to the APPC LU Properties window.
- 10** Click on the **Cancel** push button to return to the Servers and Connections window.

Connection Properties

To configure settings for a new connection, do the following:

- 1 In the Servers and Connections window, select the server that you want.
- 2 Select **Services->New Connection**. The Insert Connection window opens.
- 3 In the Insert Connection window, select the type of connection you want. Our example uses 8 2.2. See the online help for information on the other available choices.
- 4 Click on **OK**. The Connection Properties window opens. Use the following values:

Connection Name	LINKNYX1	15
Link Service	SnaDlc1	
Remote End	Peer System	radio button
Activation	On Demand	radio button
Allowed Directions	Outgoing Calls	check box.

- 5 Click on the **Setup** push button.
The 802.2 Setup window opens.
- 6 Use the following values:

Remote Network Address	4	5AA8B532	16
Local Node ID	5D	2751	21
Network Name	SPIFNET		19
Control Point Name	NYX1		13
- 7 Click on **OK** to save your work and return to the Connection Properties window.
- 8 Click on **OK**.

Remote LU Properties

To assign a remote APPC LU to a connection do the following:

- 1 In the Servers and Connections window, select the connection that you want.
- 2 Select **Services->Assign LUs**. The Insert LU window opens.
- 3 In the Insert LU window, select **APPC Remote**.
- 4 Click on **OK**. The New APPC Remote LU Properties window opens.
- 5 Fill in the values for your system. For example.

LU Alias	NYX1GW	1
Network name	SPIFNET	17
Remote LU name	NYX1GW	1 18

6 Make sure the **Supports Parallel Sessions** and the **Enable Automatic Partnering** check boxes are selected.

7 Click on **OK**.

CPI-C Symbolic Destination Name Properties

To configure the properties for CPI-C, do the following:

1 Select **Options->CPI-C**. The Configure CPI-C Names window opens.

2 Click on the **Add** push button. The CPI-C Symbolic Destination Name Properties window opens.

3 Use the values that you recorded on your worksheet. For example:

Name	NYSERVER	23
SNA Service TP	this field will be blank	
Fully Qualified	SPIFNET.NYX1GW 1 17 and 18	
Mode Name	IBMRDB	26
Conversation Security	By default, None is selected. You will specify the conversation security type later.	

4 Click on **OK** to save your work and return to the Connection Properties window.

5 Click on the **Close** push button to close the Configure CPI-C Names window and return to the Servers and Connections window.



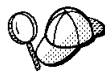
After you have finished updating your APPC client profile, go to “B. Catalog the APPC Node” on page 111.

B. Catalog the APPC Node: You must add an entry to the client's node directory to describe the remote node.

This entry specifies the chosen alias (*node_name*), Symbolic Destination Name (*sym_dest_name*), and the APPC security type (*security_type*) that the client will use for the APPC connection.

To catalog the APPC node, perform the following steps:

1 Log on to the system as a user with System Administrative (SYSADM) or System Controller (SYSCTRL) authority.



If you have trouble logging on to the system, see “Logging on to the System” on page 119.

2 If you are using a UNIX client, set up the instance environment and invoke the DB2 command line processor. Run the start-up script as follows:

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

where *INSTHOME* is the home directory of the instance.

- 3 Catalog the node by entering the following commands in the command line processor:

```
catalog appc node node_name remote sym_dest_name security security_type
terminate
```

Note: The *sym_dest_name* parameter is case-sensitive and *must* exactly match the value used in “A. Update the APPC Profiles” on page 86.

For example, to catalog a remote database server with the Symbolic Destination Name *NYSERVER* on the node called *db2node*, using APPC Security type *none*, use:

```
catalog appc node db2node remote NYSERVER security none
terminate
```



If you need to change values that were set with the **catalog node** command, first run the **uncatalog node** command in the command line processor as follows:

```
uncatalog node node_name
```

Recatalog the node with the value that you want to use.

C. Catalog the Database: Before a client application can access a remote database, the database must be cataloged on the server node and on any client nodes that will connect to it. When you create a database, it is automatically cataloged on the server with the database alias (*database_alias*) the same as the database name (*database_name*). The information in the database directory, along with the information in the node directory, is used on the client to establish a connection to the remote database.

To catalog a database on the client, perform the following steps.

- 1 Log on to the system as a user with System Administrative (SYSADM) or System Controller (SYSCTRL) authority.



If you have trouble logging on to the system, see “Logging on to the System” on page 119.

- 2 Fill in the *Your Value* column in the following worksheet.

Parameter	Description	Sample Value	Your Value
Database name (<i>database_name</i>)	The database alias (<i>database_alias</i>) of the remote database. When you create a database, it is automatically cataloged on the server with the database alias (<i>database_alias</i>) the same as the database name (<i>database_name</i>).	sample	
Database alias (<i>database_alias</i>)	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 (<i>database_name</i>). This is the name that you use when connecting to a database from a client.	tor1	
Node name (<i>node_name</i>)	The name of the node directory entry that describes where the database resides. Use the same value for node name (<i>node_name</i>) that you used to catalog the node in the previous step.	db2node	



If you are configuring communications for a Windows 3.x client, you can use the Client Setup tool to complete this task. Go to “Using the Client Setup Tool to Configure a Windows 3.x Client” on page 115 to catalog a database using the Client Setup tool.

- 3** If you are using a UNIX client, set up the instance environment and invoke the DB2 command line processor. Run the start-up script as follows:

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

where *INSTHOME* is the home directory of the instance.

- 4** Catalog the database by entering the following commands in the command line processor:

```
catalog database database_name as database_alias at node node_name
terminate
```

For example, to catalog a remote database called *sample* so that it has the alias *tor1*, on the node *db2node*, use:

```
catalog database sample as tor1 at node db2node
terminate
```



If you need to change values that were set with the **catalog database** command, first run the **uncatalog database** command in the command line processor as follows:

```
uncatalog database database_alias
```

Recatalog the database with the value that you want to use.

Step 3. Test the Client-to-Server Connection: When the configuration of the 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. Refer to the *Quick Beginnings* manual for more information.

- 1 Start the database manager by entering the **db2start** command on the server (if it was not automatically started at boot time).
- 2 Enter the following command in the client's Command Center or command line processor to connect the client to the remote database:

```
connect to database_alias user userid using password
```

The values for *userid* and *password* must be valid for the system on which they are authenticated. By default, authentication takes place on the SERVER. If the database manager is configured for CLIENT authentication, the *userid* and *password* must be valid on the client.

If the connection is successful, you will get a message showing the name of the database to which you have connected. You are now able to retrieve data from that database. For example, to retrieve a list of all the table names listed in the system catalog table, enter the following SQL command in the Command Center or command line processor:

```
"select tabname from syscat.tables"
```

When you are finished using the database connection, enter the **connect reset** command to end the database connection.



You are ready to start using the DB2 server. See Chapter 11, "Getting Started with DB2 Universal Database" on page 119 for details.

Troubleshooting the Client-to-Server Connection: If the connection fails, check the following items:

At the *server*:

1. The *db2comm* registry value includes the value *appc*.



Check the settings for the *db2comm* registry value by entering the **db2set DB2COMM** command. For more information, see Chapter 15, "Controlling Your DB2 Environment" on page 157.

2. The transaction program name (*tpname*) parameter was updated correctly in the database manager configuration file (or the admin server configuration file, if you are setting up the Administration Server).
3. The security service was started (enter the **net start db2ntsecserver** command - for Windows NT servers only).
4. The database was created and cataloged properly.
5. The database manager was stopped and started again (enter the **db2stop** and **db2start** commands on the server).



If there are problems starting a protocol's connection managers, a warning message is displayed and the error messages are logged in the *db2diag.log* file.

Refer to the *Troubleshooting Guide* for information on the *db2diag.log* file.

At the *client*:

1. The node was cataloged with the correct Symbolic Destination Name (*sym_dest_name*).
2. The node name (*node_name*), specified in the database directory, points to the correct entry in the node directory.
3. The database was cataloged properly, using the *server's* database alias (*database_alias*) that was cataloged when the database was created on the server, as the database name (*database_name*) on the client.

After you verify these items, refer to the *Troubleshooting Guide* if the connection still fails.

Using the Client Setup Tool to Configure a Windows 3.x Client

The Client Setup tool provides you with a graphical interface that you can use to complete the catalog node and database instructions in Chapter 10, "Configuring Client-to-Server Communications Using the Command Line Processor" on page 55. This section describes how to use the Client Setup tool to catalog a node or database for a Windows 3.x client.

Use the instructions in this section, along with the protocol-specific instructions in Chapter 10, "Configuring Client-to-Server Communications Using the Command Line Processor" on page 55, to configure your Windows 3.x client for communications with a DB2 server.

To invoke the Client Setup tool, double-click on the **Client Setup** icon, located in the DB2 product folder for your particular platform.

Cataloging the Node

- 1 Select **Node->New**, the New Node window opens.
- 2 Fill in the node name (*node_name*) parameter in the **Node** field and optionally add a comment describing the node in the **Comment** field.
- 3 Select the **Protocol** radio button for the protocol that you want the Windows 3.x client to use to communicate with the server.
- 4 Fill in the required parameters in the **Protocol Settings** box to catalog the node for the protocol that you selected.
- 5 Click on **OK**.

Cataloging the Database

- 1 Click on the **Databases** push button, the DB2 Client Setup - Databases window opens.
- 2 Select **Database->New**, the New Database window opens.
- 3 Fill in the **Name** (*database_name*) and **Alias** (*database_alias*) parameters and optionally add a comment in the **Comment** field describing the database.
- 4 Select the **Location** radio button that describes the location of the database that you are trying to connect to.

If the database resides on a remote DB2 server, select the **Other Node** radio button and select the name of the Node that represents the remote server from the **At Node** drop down box.

If the database resides locally, select the **This node** radio button and select the drive where the database is located from the **On** drop down box.
- 5 Select the **Authentication Type (DB2 1.x Database)** check box if you are configuring communications to a DB2 Version 1.x database.



After you have cataloged the node and database for the Windows 3.x client, and completed the other specific steps for the protocol that you are using in Chapter 10, "Configuring Client-to-Server Communications Using the Command Line Processor" on page 55, you can test the database connection by clicking on the **Test Database Connection** push button.

If the connection is not successful, see the "Troubleshooting the Client to DB2 Server Connection" section for the protocol that you are trying to configure.

Part 3. Using DB2 Universal Database

Chapter 11. 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.”

Logging on to the System

To use DB2, log on with a valid DB2 user ID that has the appropriate authority level for the commands you want to execute. If the *sysadm_group* parameter is defined in the database manager configuration file, the user performing administration tasks (such as cataloging) must belong to the System Administrative (SYSDAM) group.

The *sysadm_group* parameter is not set when DB2 is first installed; therefore, the following criteria are used to determine default SYSADM authority:

OS/2

Any valid DB2 user ID which belongs to the UPM Administrator or Local Administrator group.

UNIX

Any valid DB2 username that belongs to the primary group of the instance owner's user ID.

Windows 95 and Windows 98

Any Windows 95 or Windows 98 username that is a valid DB2 user ID.

Windows NT

Any valid DB2 user account which belongs to the local Administrators group.

See Appendix C, “Naming Rules” on page 279 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.

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. While DCE authentication is an option, you cannot specify it when you create an instance. Refer to *Administration Guide* for more information on all authentication types.

Connecting to a local database

To connect to a local database called `SAMPLE`, enter 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, enter 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.)



UNIX passwords and userids are case sensitive.

To connect a client to a remote database called `SAMPLE`, using `CLIENT` authentication, enter 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, enter the **net start db2ntsecserver** command.

When a successful connection is made, a message similar to the following is displayed:

```
Database product      = DB2/NT 5.2.  
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, refer to the *Quick Beginnings* manual for your operating system.

Refer to the *Administration Guide* for more information on authentication.

Chapter 12. Working with Instances

The information in this section describes DB2 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 failure. In the event of an instance failure, only one instance is affected. Other instances can continue to function normally.

Each additional instance requires additional system resources (virtual storage and disk space) and more administration.

The Default Instance

On OS/2 and Windows 32-bit operating systems:

A default instance called DB2 is created when you install DB2. The instance name is used to set up the directory structure.

The following are set during installation: the environment variable *DB2INSTANCE=DB2*, and the DB2 registry value *DB2INSTDEF=DB2*. These settings establish DB2 as the default instance. You can change the instance that is used by default, but first you have to create an additional instance.

On UNIX:

There is no default instance on UNIX.

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 directory
- The DB2 diagnostic file (db2diag.log)
- Any other files that contain debugging information, such as the exception/register dump or the call stack for the DB2 processes.

On OS/2 and Windows 32-bit operating systems:

The instance directory is located in the `x:\sqlllib` directory, where `x:` is the drive where you installed DB2.

On UNIX operating systems:

The instance directory is located in the `INSTHOME/sqlllib` directory, where `INSTHOME` is the home directory of the instance owner.

Creating Instances

On OS/2 and Windows 32-bit operating systems:

The command syntax to create an instance is:

```
db2icrt instance_name
```

where *instance_name* is an alphanumeric string up to eight characters long. See Appendix C, “Naming Rules” on page 279 for more information about valid names.



You can change the location of the instance directory from `DB2PATH` using the `DB2INSTPROF` environment variable. You require write access for the instance directory. If you want the directories created in a path other than `DB2PATH`, you have to set the `DB2INSTPROF` environment variable before entering the **db2icrt** command. See Chapter 15, “Controlling Your DB2 Environment” on page 157 for further information.

On UNIX operating systems:

The instance owner and the group that is the System Administrative (SYSADM) group are associated with every instance. The instance owner and the SYSADM group are assigned during the process of creating the instance. One user ID can be used for only one instance. That user ID is also referred to as the *instance owner*.

Each instance owner must have a unique home directory. All of the files necessary to run the instance are created in the home directory of the instance owner's user ID. If it becomes necessary to remove the instance owner's user ID from the system, you could potentially lose files associated with the instance and lose access to data stored in this instance. For this reason, it is recommended that you dedicate an instance owner user ID to be used exclusively to run DB2.

The primary group of the instance owner is also important. This primary group automatically becomes the system administrative group for the instance and gains SYSADM authority over the instance. Other user IDs that are members of the primary group of the instance owner also gain this level of authority. For this reason, you may want to assign the instance owner's user ID to a primary group that is reserved for the administration of instances. (Also make sure that you assign a primary group to the instance owner user ID; otherwise, the system-default primary group is used.)

If you already have a group that you want to make the system administrative group for the instance, you can simply assign this group as the primary group when you create the instance owner user ID. To give other users administrative authority on the instance, add them to the group that is assigned as the system administrative group.

To separate SYSADM authority between instances, ensure that each instance owner user ID uses a different primary group. However, if you choose to have common SYSADM authority over multiple instances, you can use the same primary group for multiple instances.

To create a DB2 instance, use the **db2icrt** command as follows:

```
DB2DIR/instance/db2icrt instance_name
```

where DB2DIR	= /usr/lpp/db2_05_00	on AIX
	= /opt/IBMdb2/V5.0	HP-UX, SCO UnixWare 7, Solaris, or SGI IRIX
	= /usr/IBMdb2/V2.1	on SCO OpenServer
	= /opt/IBMdb2/V2.1	on SINIX

and where *instance_name* is an alphanumeric string up to eight characters long (refer to the Appendix C, "Naming Rules" on page 279 for more information about naming rules).

For example, to create an instance, enter the following command:

```
db2icrt db2inst1
```

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

Listing Instances

On OS/2 and Windows 32-bit operating systems:

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:

```
db2 get instance
```

On UNIX Systems:

To get a list of all the instances that are available on a system, enter the **db2ilist** command. The **db2ilist** command is located in:

```
DB2DIR/bin/db2ilist
```

where DB2DIR	= /usr/lpp/db2_05_00	on AIX
	= /opt/IBMDB2/V5.0	HP-UX, SCO UnixWare 7, Solaris, or SGI IRIX
	= /usr/IBMDB2/V2.1	on SCO OpenServer
	= /opt/IBMDB2/V2.1	on SINIX

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

```
db2 get instance
```

Note: If you have changed to another instance, run the **db2 terminate** command before checking the instance.

Setting the Current Instance

On OS/2 and Windows 32-bit operating systems:

When you enter 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:

Read the setting for the *DB2INSTANCE* environment variable for the current session. To set the *DB2INSTANCE* environment variable, enter:

```
set db2instance=new_instance_name
```

where *new_instance_name* represents the instance that you want to work with.

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 32-Bit Operating Systems” on page 160 for instructions on setting *DB2INSTANCE*.
- On Windows 95 and Windows 98, system environment variables are set in *autoexec.bat* or other batch file. See “Setting Your Environment on Windows 32-Bit Operating Systems” on page 160 for instructions on setting *DB2INSTANCE*.
- On OS/2, system environment variables are set in *config.sys* or other command file. See “Setting Your Environment on OS/2” on page 159 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 globally, enter:

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

where *new_instance_name* represents the instance that you want to be the default for your system.

For more information on setting registry values, refer to Chapter 15, “Controlling Your DB2 Environment” on page 157.

On UNIX Systems:

To use a specific instance, you need to execute the startup script for that instance. Enter the following command:

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

where *INSTHOME* is the home directory of the instance.

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

```
db2 get instance
```

Updating Instances on UNIX Systems

Existing instances are designed to be as independent as possible from the effects of subsequent installation and removal of products.

In most cases, existing instances will automatically inherit or lose access to the function of the product that is being installed or removed. However, if certain executables or components are installed or removed, existing instances do not automatically inherit the new system configuration parameters or gain access to all the additional function. The instance must be updated.

If DB2 is updated by installing a PTF or a patch, all the existing instances should be updated using the **db2iupdt** command. You can run the **db2iupdt** command as follows:

```
DB2DIR/instance/db2iupdt instance_name
```

where DB2DIR	= /usr/lpp/db2_05_00	on AIX
	= /opt/IBMDB2/V5.0	HP-UX, SCO UnixWare 7, Solaris, or SGI IRIX
	= /usr/IBMDB2/V2.1	on SCO OpenServer
	= /opt/IBMDB2/V2.1	on SINIX

For example, to update an instance, enter the following command:

```
db2iupdt db2inst1
```

The **db2iupdt** command updates the files in the `INSTHOME/sqllib` directory, where `INSTHOME` is the home directory of the instance owner.

Removing Instances

On OS/2 and Windows 32-bit operating systems:



Ensure that there are no applications using the instance before removing it.

To remove an instance, perform the following steps:

- 1 Close all applications that are currently using the instance.
- 2 Stop the Command Line Processor by entering the **db2 terminate** command in each DB2 command window.

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

4 Back up the files in the `x:\sqllib\instance_name` directory (where `x:` is the drive where DB2 is installed) to any directory outside of this path. Once you enter the **db2idrop** command, the `x:\sqllib\instance_name` directory is removed and cannot be recovered.

Note: If the `DB2INSTPROF` registry value is set, these files will be in a different location than the one used in this example.

5 Enter the **db2idrop** command as follows:

```
db2idrop instance_name
```

where *instance_name* represents the instance you want to drop.

On UNIX Systems:

To remove an instance, perform the following steps:

1 Log in as the instance owner.

2 Back up files in the `INSTHOME/sqllib` directory, if needed.

You might want to save the database manager configuration file, `db2system`, the `db2nodes.cfg` file, or user defined function or fenced stored procedure applications in `INSTHOME/sqllib/function`, where `INSTHOME` is the home directory of the instance owner.

3 Log off as the instance owner.

4 Log in as user with root authority.

5 Remove the instance using the **db2idrop** command. You can enter the `DB2IDROP` command as follows:

```
DB2DIR/instance/db2idrop InstName
```

InstName is the login name of the instance.

where DB2DIR	= /usr/lpp/db2_05_00	on AIX
	= /opt/IBMDB2/V5.0	HP-UX, SCO UnixWare 7, Solaris, or SGI IRIX
	= /usr/IBMDB2/V2.1	on SCO OpenServer
	= /opt/IBMDB2/V2.1	on SINIX

6 Optionally, as a user with root authority, remove the instance owner's user ID and group (if used only for that instance). Do not remove these if you are planning to re-create the instance.

This step is optional since the instance owner and the instance owner group may be used for other purposes.

The **db2idrop** command removes the instance entry from the list of instances and removes the `INSTHOME/sqllib` directory.

Chapter 13. 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 bind files 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 or service levels of DB2, you must bind the utilities once for each operating system and 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 sql1lib\win on Windows 3.x systems and sql1lib for OS/2, and Windows 32-bit operating systems). 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 and Windows 32-bit 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:

For UNIX platforms INSTHOME/sql1lib/bnd

For all other platforms x:\sql1lib\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 Enter 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 BINDADD privileges are granted to *public*.

- 4 Reset the connection to the database by entering 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 BINDADD 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 131 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 database (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.x, 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.x 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. A unique index should be created for each DB2 table that is to be updated by an ODBC application. Refer to the **CREATE INDEX** statement in the *SQL Reference*. Using the Control Center you would alter the settings of the table, then select the **Primary Key** tab and move one or more columns from the available columns list over to the primary key columns list. Any column you select as part of the primary key must be defined as NOT NULL.
- 4 Various CLI/ODBC Configuration Keywords can be set to modify the behavior of DB2 CLI/ODBC and the applications using it.
- 5 If you followed the above steps to install ODBC support, and added DB2 databases as ODBC data sources, your ODBC applications will now be able to access them.

After the platform specific instructions there are further details on the following topics:

“How to Bind the DB2 CLI/ODBC Driver to the Database” on page 144

“How to Set CLI/ODBC Configuration Keywords” on page 144

“Configuring db2cli.ini” on page 145

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.x, Windows 95, Windows 98, and Windows NT Client Access to DB2 using CLI/ODBC” on page 135

“OS/2 Client Access to DB2 using CLI/ODBC” on page 137

“UNIX Client Access to DB2 using CLI/ODBC” on page 139

Windows 3.x, Windows 95, Windows 98, 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 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 251 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 32-bit operating systems they are both installed with DB2 unless the ODBC component is manually unselected during the install. On Windows 3.x 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 32-bit operating systems, **odbcadm.exe** on Windows 3.x.
- 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 on Windows 32-bit operating systems, or double-click on the **Install ODBC Driver** icon in the DB2 program group in Windows 3.x.

- 3** Register the DB2 database with the ODBC driver manager as a *data source*. On Windows 32-bit operating systems 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 32-bit operating systems you can use the radio buttons to add the data source as either a user or system data source.

Using the Microsoft **32-bit 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 32-bit operating systems 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 32-bit operating systems 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 “Configuration Keyword Descriptions” on page 173.

For information on manually editing this file (`db2cli.ini`), see “Configuring `db2cli.ini`” on page 145.

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 32-bit operating systems.

Both 16- and 32-bit applications use the same 32-bit IBM DB2 CLI/ODBC Driver. The Microsoft ODBC Driver Manager takes care of the conversion between the 16-bit application and the 32-bit ODBC driver.

Some 16-bit applications were shipped with an old set of 16-bit Driver Manager DLLs (that is, before 1995). These DLLs do not function well in the 32-bit environment. The following symptoms may occur when running the old DLLs:

Traps occur in `odbc.dll` or other ODBC related DLLs.

16-bit applications do not see data sources cataloged using the 32-bit ODBC Administrator (from the Control Panel).

If (and only if) you are experiencing these problems you may want to update the Microsoft 16-bit ODBC Driver Manager DLLs. These files are provided with DB2 in the `SQLLIB\MSODBC16` subdirectory. Use these new DLLs to replace the older versions that are located in the `system` subdirectory of the Windows operating system directory.

- 6 If you have installed ODBC access (as described above), you can now access DB2 data using ODBC applications. Start the ODBC application and go to the Open window. Select the **ODBC databases** file type. The DB2 databases that you added as ODBC data sources will be selectable from the list. Many ODBC applications will open the table as read-only unless a unique index exists.



If you require additional information at this point you can refer to the following topics in "Detailed Configuration Information" on page 143:

"How to Bind the DB2 CLI/ODBC Driver to the Database" on page 144

"How to Set CLI/ODBC Configuration Keywords" on page 144

"Configuring `db2cli.ini`" on page 145

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 251.

- 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 "Configuration Keyword Descriptions" on page 173.

For information on manually editing this file (`db2cli.ini`), see "Configuring db2cli.ini" on page 145.

- 4** If you have installed ODBC access (as described above), you can now access DB2 data using ODBC applications. Start the ODBC application and go to the Open window. Select the **ODBC databases** file type. The DB2 databases that

you added as ODBC data sources will be selectable from the list. Many ODBC applications will open the table as read-only unless a unique index exists.



If you require additional information at this point you can refer to the following topics in “Detailed Configuration Information” on page 143:

- “How to Bind the DB2 CLI/ODBC Driver to the Database” on page 144
 - “How to Set CLI/ODBC Configuration Keywords” on page 144
 - “Configuring db2cli.ini” on page 145
-

UNIX Client Access to DB2 using CLI/ODBC

Before DB2 CLI and ODBC applications can successfully access a DB2 database from a UNIX client, perform the following steps on the client system:

- 1** The DB2 database (and node if the database is remote) must be cataloged. To do so, use the command line processor.

For more information see Chapter 10, “Configuring Client-to-Server Communications Using the Command Line Processor” on page 55 or the **CATALOG DATABASE** and **CATALOG NODE** commands in the *Command Reference*.

- 2** The DB2 CLI/ODBC driver is automatically installed when DB2 is installed, and therefore nothing needs to be done.

- 3** If you are using ODBC applications to access DB2 data, perform the following steps. (If you are using only CLI applications, skip this step and go to the next step.)

- a** When using an ODBC application you must ensure that the Visigenic or Intersolv ODBC Driver Manager components are installed and that each user that will use ODBC has access to it.

If DB2 installed the Driver Manager it would be located in the `sqllib/odbc1ib` subdirectory.

If the Driver Manager was installed by another program, the file `.odbc.ini` (begins with a period) would be located in the home directory of the user ID that runs the ODBC application.

DB2 does not install an ODBC Driver Manager on the SCO UnixWare operating system. You must use the ODBC Driver Manager that was supplied with your ODBC client application or ODBC SDK in order to access DB2 data using that application.

- b** The Driver Manager uses two initialization files. A sample template of these files are provided in the `sqllib/odbc1ib` subdirectory.

`odbcinst.ini` ODBC Driver Manager's configuration file indicating which database drivers are installed. Each user that will use ODBC must have access to this file.

`.odbc.ini` End-user's data source configuration. Each user has a separate copy of this file. The sample template file is called `odbc.ini` (without the first dot).

Setting up `odbcinst.ini`

The settings in this file impact all of the ODBC drivers on the machine.

Use an ASCII editor to update this file. It must have a stanza (section) called `[IBM DB2 ODBC DRIVER]`, with a line starting with "Driver" indicating the full path to the DB2 ODBC driver (`db2.o`). For example, if the home directory of your end user is `/u/thisuser/` and the `sqllib` directory is installed there, then the correct entry would be:

```
[IBM DB2 ODBC DRIVER]
Driver=/u/thisuser/sqllib/lib/db2.o
```

See the sample file in the `sqllib/odbc1ib` subdirectory for an example.

Setting up `odbc.ini`

The settings in this file are associated with a particular user on the machine; different users can have different `odbc.ini` files.

The `odbc.ini` file must be copied into the end user's home directory and called `.odbc.ini`. Update this file, using an ASCII editor, to reflect the appropriate data source configuration information. To register a DB2 database as an ODBC data source there must be one stanza (section) for each DB2 database.

The sample `odbc.ini` template shows:

line 2 (part of the `[ODBC Data Source]` stanza):

```
SAMPLE=IBM DB2 ODBC DRIVER
```

Indicates that there is a data source called `SAMPLE` that used the `IBM DB2 ODBC DRIVER`.

lines 4-6 (part of the `[SAMPLE]` stanza):

```
[SAMPLE]
Driver=/u/thisuser/sqllib/lib/db2.o
Description=Sample DB2 ODBC Database
```

Indicates that the `SAMPLE` database is part of the DB2 instance located in the directory `/u/thisuser`.

line 11 (part of the `[ODBC]` stanza):

```
InstallDir=/u/thisuser/sqllib/odbc1ib
```

Indicates that `/u/thisuser/sqllib/odbc1ib` should be treated as the location where ODBC is installed.

Note: If your application specifically installed the ODBC components (such as the Driver Manager) elsewhere, you must update the line starting with `InstallDir` under the `[ODBC]` stanza to reflect this new location.

For example, if ODBC has been installed in `/opt/odbc`, the [ODBC] stanza would look like:

```
[ODBC]
Trace=
TraceFile=odbctrace.out
InstallDir=/opt/odbc
```

See the sample file in the `sql1lib/odbc1lib` subdirectory for an example. You can also see “How to Configure ODBC.INI” on page 145 for more detailed information.

Once the .ini files are set up you can run your ODBC application and access DB2 databases. Refer to the documentation that comes with your ODBC application for additional help and information.

4 Configure the DB2 CLI/ODBC driver (optional).

There are various keywords and values that can be used to modify the behavior of DB2 CLI/ODBC and the applications using it. The keywords are associated with the database *alias name*, and affect all DB2 CLI/ODBC applications that access the database.

For information on manually editing this file (`db2cli.ini`), see “Configuring `db2cli.ini`” on page 145. For information about the specific keywords see the *CLI Guide and Reference*.



If you require additional information at this point you can refer to the following topics in “Detailed Configuration Information” on page 143:

- “How to Bind the DB2 CLI/ODBC Driver to the Database” on page 144
 - “How to Set CLI/ODBC Configuration Keywords” on page 144
 - “Configuring `db2cli.ini`” on page 145
-

Macintosh Client Access to DB2 using CLI/ODBC

Before DB2 CLI and ODBC applications can successfully access a DB2 database from a Macintosh client, perform the following steps on the client system:

- 1 The DB2 database (and node if the database is remote) must be cataloged. To do so, use the command line processor.
For more information see “Configuring Database Connections” on page 251.
- 2 If you are using ODBC applications to access DB2 data, perform the following steps. (If you are using only CLI applications, skip to step #4.)
 - a An ODBC driver manager must be installed to handle the communications between the ODBC applications and the DB2 ODBC driver. DB2 does not supply a Macintosh ODBC driver manager, only the DB2 ODBC driver itself.

If your ODBC application does not supply an ODBC driver manager then you can obtain the MacODBC driver manager directly from Apple.

- b** Register the DB2 database with the ODBC driver manager list of data sources (contained in the ODBC Preferences file) as a *data source* using either of these methods:

- Running **ODBC Setup** (in a 68K environment) or **ODBC Setup PPC** (in a PowerMacintosh environment) from the Control Panel

- Editing the initialization (ODBC Preferences or ODBC Preferences PPC) file directly; refer to the next step in this process for more information.

- 3** Configuring ODBC Preferences or ODBC Preferences PPC: The ODBC initialization file is used to record information such as the available drivers and data sources. See the documentation for your driver manager for procedures on updating this file.

The MacODBC driver manager uses the ODBC Preferences file or the ODBC Preferences PPC file to record information about the available drivers and data sources.

It is also possible to modify these files manually. Do not change any of the existing entries in the files.

- a** Use an ASCII editor to edit the ODBC Preferences file or the ODBC Preferences PPC file.

68K Environment

The following is a sample ODBC Preferences file:

```
[ODBC Data Sources]
GLOBALDB=IBM ODBC DB2 for Macintosh
SAMPLE=IBM ODBC DB2 for Macintosh

[GLOBALDB]
Driver=appl:ODBC$DB2DriverFunctionSet
Description=My GLOBAL database

[SAMPLE]
Driver=appl:ODBC$DB2DriverFunctionSet
Description=My SAMPLE database
```

PowerMacintosh Environment

The following is a sample ODBC Preferences PPC:

```
[ODBC Data Sources]
SAMPLE=IBM ODBC DB2 for PPC

[SAMPLE]
Driver=DB2ODBCDriverPPC
Description=My SAMPLE database
```

- b** The [ODBC Data Source] section lists the name of each available data source and the description of the associated driver.

For each data source listed in the [ODBC Data Source] section, there is a section that lists additional information about that data source. These are called the *Data Source Specification* sections.

Under the [ODBC DATA SOURCE] entry, add the following line:

68K Environment

```
database_alias=IBM ODBC DB2 for Macintosh
```

PowerMacintosh Environment

```
database_alias=IBM ODBC DB2 for PPC
```

where *database_alias* is the alias of the database cataloged in the database directory (the database name used by the Command Line Processor CONNECT TO statement).

- c** Add a new entry in the Data Source Specification section to associate the data source with the driver:

68K Environment

```
[database_alias]  
Driver=appl:ODBC$DB2DriverFunctionSet
```

PowerMacintosh Environment

```
[database_alias]  
Driver=DB2ODBCDriverPPC
```

Where *database_alias* is the alias of the database cataloged in the database directory, and listed under the Data Source Specification section.

- 4** The DB2 CLI/ODBC driver can be configured further by editing the `db2cli.ini` file. This file contains various keywords and values that can be used to modify the behavior of DB2 CLI and the applications using it. The keywords are associated with the database *alias name*, and affect all DB2 CLI applications that access the database. For a complete description of all the keywords and their usage, refer to “Configuration Keyword Descriptions” on page 173.
- 5** If you have installed ODBC access (as described above), you can now access DB2 data using ODBC applications. Start the ODBC application and go to the Open window. Select the **ODBC databases** file type. The DB2 databases that you added as ODBC data sources will be selectable from the list. Many ODBC applications will open the table as read-only unless a unique index exists.

Detailed Configuration Information

The section “Platform Specific Details for CLI/ODBC Access” on page 134 should provide you with all of the information you require. The following additional information is useful where DB2 tool support is not available, and for administrators who require more detailed information.



The following topics are covered in this section:

“How to Bind the DB2 CLI/ODBC Driver to the Database” on page 144

“How to Set CLI/ODBC Configuration Keywords” on page 144

“Configuring db2cli.ini” on page 145

How to Bind the DB2 CLI/ODBC Driver to the Database

The CLI/ODBC driver will autobind on the first connection to the database, provided the user has the appropriate privilege or authorization. The administrator may want to perform the first connect or explicitly bind the required files.

See “Binding Database Utilities” on page 131 for more information.

How to Set CLI/ODBC Configuration Keywords

DB2 CLI can be configured further by using either the CCA or the DB2 Client Setup administration tool, whichever is applicable for your platform, or by manually editing the `db2cli.ini` file.

This file contains various keywords and values that can be used to modify the behavior of DB2 CLI and the applications using it. The keywords are associated with the database *alias name*, and affect all DB2 CLI and ODBC applications that access the database.

By default, the location of the CLI/ODBC configuration keyword file is as follows:

Platform:	Location:
OS/2	<code>sqllib</code> directory
Windows NT	<code>sqllib</code> directory
Windows 95	<code>sqllib</code> directory
Windows 98	<code>sqllib</code> directory
Windows 3.x	<code>sqllib\win</code> directory
UNIX	<code>sqllib/cfg</code> directory of the database instance running the CLI/ODBC applications

The environment variable `DB2CLIINIPATH` can also be used to override the default and specify a different location for the file.

The configuration keywords enable you to:

Configure general features such as data source name, user name, and password.
Set options that will affect performance.
Indicate query parameters such as wild card characters.
Set patches or work-arounds for various ODBC applications.
Set other, more specific features associated with the connection, such as code pages and IBM Graphic data types.

For a complete description of all the keywords and their usage, refer to “Configuration Keyword Descriptions” on page 173.

Configuring db2cli.ini: The `db2cli.ini` initialization file is an ASCII file which stores values for the DB2 CLI configuration options. A sample file is shipped to help you get started. Refer to the *CLI Guide and Reference* for information on each keyword.

See “Platform Specific Details for CLI/ODBC Access” on page 134 for more information on how to modify this file on your platform.

How to Configure ODBC.INI

Microsoft's 16-bit ODBC Driver Manager and Visigenic's ODBC Driver Manager use the `odbc.ini` file to record information about the available drivers and data sources. Visigenic's ODBC Driver Manager also uses the `odbcinst.ini` file on UNIX platforms. Although the necessary files are updated automatically by the tools on most platforms, users of ODBC on UNIX platforms will have to edit them manually. The file `odbc.ini` (and `odbcinst.ini` where required) are located:

Platform: **Location:**

Windows `x:\windows` (where *x* is the drive where Windows is installed)

WIN-OS/2 `x:\os2\mdos\winos2` (where *x* is the drive where OS/2 is installed)

UNIX Home directory of user ID running ODBC application (on UNIX the `odbc.ini` file name has a dot before it: `.odbc.ini`)

It is also possible to modify this file manually. Do not change any of the existing entries in the file.

- 1 Use an ASCII editor to edit the `odbc.ini` file.

The following is an example `odbc.ini` file:

```
[ODBC Data Sources]
MS Access Databases=Access Data ( .mdb)

[MS Access Databases]
Driver=D:\WINDOWS\SYSTEM\simba.dll
FileType=RedISAM
SingleUser=False
UseSystemDB=False
```

The [ODBC Data Sources] section lists the name of each available data source and the description of the associated driver.

For each data source listed in the [ODBC Data Sources] section, there is a section that lists additional information about that data source. These are called the *Data Source Specification* sections.

- 2 Under the [ODBC DATA SOURCE] entry, add the following line:

```
database_alias=IBM DB2 ODBC DRIVER
```

where *database_alias* is the alias of the database cataloged in the database directory (the database name used by the command line processor CONNECT TO statement).

- 3 Add a new entry in the Data Source Specification section to associate the data source with the driver:

```
[database_alias]  
Driver=x:\windows\system\db2cliw.dll
```

Where:

database_alias is the alias of the database cataloged in the database directory, and listed under the Data Source Specification section.
x is the drive where Windows is installed.

The following shows the example file with the IBM data source entries added:

```
[ODBC Data Sources]  
MS Access Databases=Access Data ( .mdb)  
SAMPLE=IBM DB2 ODBC DRIVER  
  
[MS Access Databases]  
Driver=D:\WINDOWS\SYSTEM\simba.dll  
FileType=RedISAM  
SingleUser=False  
UseSystemDB=False  
  
[SAMPLE]  
Driver=D:\WINDOWS\SYSTEM\db2cliw.dll  
Description=Sample DB2 Client/Server database
```

Note: If you are running the ODBC application under WIN-OS/2, specify the equivalent path \OS2\MDOS\WINOS2\SYSTEM in place of the \WINDOWS\SYSTEM path.

UNIX Configuration of .ini files

The section "UNIX Client Access to DB2 using CLI/ODBC" on page 139 contains detailed steps on how to update both the `odbc.ini` and `odbcinst.ini` files.

Running Java Programs

You can develop Java programs to access DB2 databases with the appropriate Java Development Kit (JDK) on AIX, HP-UX, OS/2, SCO UnixWare, Solaris, or Windows 32-bit operating systems. The JDK includes Java Database Connectivity (JDBC), a dynamic SQL API for Java.

DB2 JDBC support is provided by the DB2 Client Application Enabler (DB2 CAE). With DB2 JDBC support you can build and run JDBC applications and applets. These contain dynamic SQL only, and use a Java call interface to pass SQL statements to DB2.

The DB2 Software Developer's Kit (DB2 SDK) provides support for Java embedded SQL (SQLJ). With DB2 SQLJ support and DB2 JDBC support you can build and run SQLJ applications and applets. These contain static SQL and use embedded SQL statements that are bound to the DB2 database.

Java can also be used on the server to create JDBC and SQLJ stored procedures and user-defined functions (UDFs).

Building and running different types of Java programs requires support from different components of DB2:

- JDBC applications do not require any other component of DB2 on the client to be built. To run JDBC applications requires the DB2 Client Application Enabler (CAE) to connect to DB2.

- To build SQLJ applications requires the DB2 SDK. To run SQLJ applications requires the DB2 CAE to connect to DB2.

- JDBC applets do not require any other DB2 component on the client to be built or run.

- SQLJ applets require the DB2 SDK to be built. They do not require any other DB2 component to be run on the client.

For detailed information on building and running JDBC and SQLJ programs on UNIX platforms see *Building Applications for UNIX Environments*. Detailed information on building and running JDBC programs on Windows and OS/2 can be found in *Building Applications for Windows and OS/2 Environments*. Building and running SQLJ programs on Windows and OS/2 is explained in the *What's New* book. For more information on DB2 programming in Java, refer to the *Embedded SQL Programming Guide*, chapter 15, "Programming in Java". This covers creating and running JDBC applications, applets, stored procedures and UDFs. Information on SQLJ applications, applets, stored procedures and UDFs can be found in the *What's New* book.

For the latest, updated DB2 Java information, visit the Web Page at:

<http://www.software.ibm.com/data/db2/java>

Configuring the Environment

To build and run DB2 Java programs, you need to install and configure the appropriate version of the Java Development Kit (JDK) on your development machine:

AIX:	The Java Development Kit (JDK) Version 1.1 for AIX from IBM.
HP-UX:	The HP-UX Developer's Kit for Java Release 1.1 from Hewlett-Packard.
OS/2:	The Java Development Kit (JDK) Version 1.1 for OS/2 from IBM.
SCO UnixWare:	No installation necessary: Java Development Kit (JDK) Version 1.1.3 for SCO UnixWare is already installed in <code>/usr/java</code> .
Solaris:	The Java Development Kit (JDK) Version 1.1.4 for Solaris, and the Solaris Native Thread pack, from Sun Microsystems.
Windows 32-bit operating systems:	The Java Development Kit (JDK) Version 1.1 for Win32 from Sun Microsystems.

For information on installing and configuring any of the above JDKs, please refer to:

<http://www.software.ibm.com/data/db2/java>

For all supported platforms, you must also install and configure the DB2 Client Application Enabler for your platform from the DB2 Client Pack. It must be Version 2.1.2 or later, except for SCO UnixWare which must be Version 5.2 or later.

To run DB2 Java stored procedures or UDFs, you also need to update the DB2 database manager configuration to include the path where the JDK version 1.1 is installed on your development machine. You can do this by entering the following on the command line:

```
db2 update dbm cfg using JDK11_PATH /home/smith/jdk11
```

where `/home/smith/jdk11` is the path where the JDK version 1.1 is installed.

You can check the DB2 database manager configuration to verify the correct value for the `JDK11_PATH` field by entering the following command:

```
db2 get dbm cfg
```

You may want to pipe the output to a file for easier viewing. The `JDK11_PATH` field appears near the beginning of the output. For more information on these commands, see the *Command Reference*.

Note: On Solaris, some Java Virtual Machine implementations do not work well in programs that run in a "setuid" environment. The shared library that contains the Java interpreter, `libjava.so`, may fail to load. As a workaround, you can create

symbolic links for all needed JVM shared libraries in `/usr/lib`, with a command similar to the following (depending on where Java is installed on your machine):

```
ln -s /opt/jdk1.1.3/lib/sparc/native_threads/.so /usr/lib
```

For more information on this and other workarounds available, please visit:

<http://www.software.ibm.com/data/db2/java/v5/faq.html>

To run Java programs, the following environment variables are automatically updated during DB2 installation on Windows and OS/2, and during instance creation on UNIX platforms.

On UNIX platforms:

CLASSPATH includes "." and the file `sqlllib/java/db2java.zip`

On AIX, SCO UnixWare, and Solaris: LD_LIBRARY_PATH includes the directory `sqlllib/lib`; on HP-UX: SHLIB_PATH includes the directory `sqlllib/lib`

On Solaris only: THREADS_FLAG is set to "native"

On Windows and OS/2 platforms:

CLASSPATH includes "." and the file `%DB2PATH%\java\db2java.zip`

In order to build and run SQLJ programs, CLASSPATH is also automatically updated to include these files:

On UNIX platforms:

`sqlllib/java/sqlj.zip` (required to build SQLJ programs)

`sqlllib/java/runtime.zip` (required to run SQLJ programs)

On Windows and OS/2 platforms:

`%DB2PATH%\java\sqlj.zip` (required to build SQLJ programs)

`%DB2PATH%\java\runtime.zip` (required to run SQLJ programs)

Java Applications

Start your application from the desktop or command line by running the Java interpreter on the executable program with this command:

```
java prog_name
```

where `prog_name` is the name of the program.

The DB2 JDBC driver handles the JDBC API calls from your application and uses the DB2 CAE to communicate the requests to the server and receive the results.

Note: An SQLJ application must be bound to the database before it is run.

Java Applets

Because Java applets are delivered over the web, a web server must be installed on your DB2 machine (server or client).

To run your applet, make sure your `.html` file is properly configured. Start the JDBC applet server on the TCP/IP port specified in the `.html` file. For example, if you specified:

```
param name=port value='6789'
```

then you would enter:

```
db2jstrt 6789
```

You must ensure that your working directory is accessible to your web browser. If it is not, copy your applet's `.class` and `.html` files into a directory that is accessible. For SQLJ applets, you must also copy the profile `.class` and `.ser` files as well.

Copy the `sqliib/java/db2java.zip` file into the same directory as these other files. For SQLJ applets, also copy the `sqliib/java/runtime.zip` file into this directory.

Then on your client machine start your web browser (which supports JDK 1.1) and load the `.html` file.

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.

Note: An SQLJ applet must be bound to the database before it is run.

Chapter 14. 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 OS/2 or Windows 32-bit operating systems. Otherwise, you can enter commands in the command line processor or at a system command prompt.

Task: Entering DB2 commands, SQL statements, and operating system commands	
Operating System	Tools
Macintosh	command line processor
OS/2	Command Center command line processor command prompt
UNIX	command line processor command line prompt
Windows 3.x	command line processor
Windows 32-bit operating systems	Command Center command line processor command window

Using the Command Center

On OS/2 and Windows 32-bit 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 SQL statements and DB2 commands in a results 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 on the **DB2 for OS/2** folder.
- 2 Double-click on the **Command Center** icon.

To invoke the Command Center on Windows 32-bit operating systems:

- 1 Click on **Start**, and 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 toolbar, or use CTRL+Enter.



It is no longer necessary to enter the commands with a db2 prefix; instead, enter the command as follows:

```
list node directory
```

To enter operating system commands in interactive mode, precede the operating-system command with an exclamation mark (!). For example:

```
!dir
```

If you want to enter multiple commands, you must end each command with the termination character, then press Enter to start the next command on a new line. (The default termination character is a semicolon(;).) To specify a different termination character, click on the **Tools Settings** icon in the menu toolbar.

After you enter a command, the Command Center displays the Results window, which informs you how the commands are proceeding.

To recall commands that you have entered, select the **Script** Tab, click on the drop down box, and select a command. To save commands 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 Center** icon from the main tool bar.

Using the Command Line Processor

You can use the command line processor to enter SQL statements, DB2 commands, and operating system commands. It operates in the following modes:

Interactive Input Mode

You can enter operating systems commands, 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 operating system commands, 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 on the **DB2 for OS/2** folder.
- 2 Double-click on the **Command Line Processor** icon.

On Windows 32-bit operating systems:

- 1 Click on **Start** and 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 on the **DB2** folder.
- 2 Double-click on the **Command Line Processor** icon.

On Macintosh:

- 1 Double-click on 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 **db2cmd** command followed by the **db2** command at a command prompt. In interactive input mode, the prompt looks like this:

```
db2 =>
```

In interactive mode, you do not have to enter DB2 commands with a **db2** prefix; instead, you just enter the DB2 command. For example:

```
db2 => list node directory
```

To enter operating system commands in interactive mode, precede the operating-system command with an exclamation mark (!). For example:

```
db2 => !dir
```



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 empno, lastname, birthdate, from \  
db2 (cont.) => employee where sex='F' order by empno desc
```

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, enter the **quit** command.

For more information on advanced topics using the command line processor, refer the *Command Reference*.

Command Line Mode

On OS/2 and UNIX operating systems, you can enter 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 32-bit 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 meaning 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 the information from the *employee* table, even if the * character has a special meaning on the operating system:

```
db2 "select  from employee"
```

2. The command line mode is not available on Windows 3.x systems.

|
|
For more information on using the command line processor, refer to the *Command Reference*.

Chapter 15. 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 158.

To set DB2 environment variables, see:

- “Setting Your Environment on OS/2” on page 159.
- “Setting Your Environment on Windows 32-Bit Operating Systems” on page 160.
- “Setting Your Environment on UNIX Systems” on page 161.
- “Setting Your Environment on Windows 3.x” on page 161.

To set database manager configuration parameters on a client instance, see “Configuration Parameters” on page 169.

Prior to the introduction of the DB2 profile registry in Version 5, changing your DB2 environment on OS/2 or Windows 32-bit operating systems required you to change your system's environment variables. On UNIX platforms, changing your DB2 environment required you to change the script files, `db2profile` or `db2cshrc` in the home directory of the instance.

With Version 5, almost all of the environment variables have been moved to the DB2 profile registry. Users with system administrative (SYSADM) authority for a given instance can update registry values for that instance. Use the **db2set** command to update DB2 registry values without rebooting your system. The DB2 registry applies the updated information to DB2 server instances and DB2 applications started after the changes are made.

Note: There is no DB2 profile registry on Windows 3.x. See “Setting Your Environment on Windows 3.x” on page 161 for more information.

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

The environment variable settings, or a variable set in the script files, `db2profile` or `db2cshrc`.

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 163 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 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.

To modify registry variable values, enter the **db2set** command.

The syntax of the **db2set** command is as follows:

To set a parameter, enter:

```
db2set parameter=value
```

This command will set the parameter that you specify at the default level for that parameter

To set a parameter's value for a specific instance, enter:

```
db2set parameter=value -i instance_name
```

To set a parameter at the global profile level, enter:

```
db2set parameter=value -g
```

Notes:

1. Some parameters will always default to the global level profile. They cannot be set at the instance or node level profiles; for example, *DB2SYSTEM* and *DB2INSTDEF*.
2. On UNIX, you must have system administrative (SYSADM) authority to change registry values for an instance. Only users with root authority can change parameters in global-level registries.

To delete a parameter's value at a specified level, you can use the same command syntax to set the parameter, but specify nothing for the parameter value. For example, to delete the parameter's setting at the instance level, enter:

```
db2set parameter= -g instance
```

To delete a parameter's setting and to restrict its use, if it is defined at a higher profile level, enter:

```
db2set parameter= -null -i instance
```

This command will delete the setting for the parameter you specify and restrict high level profiles from setting this parameter (in this case, the DB2 global-level profile).

However, the parameter you specify could still be set by a lower level profile (in this case, the DB2 node-level profile).

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 the administration 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; this environment variable 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 determine the setting of an environment variable, enter:

```
set variable
```

To change the setting of an environment variable, enter the following command:

```
set variable=value
```

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 32-Bit Operating Systems

It is strongly recommended that all DB2 specific registry values be defined in the DB2 profile registry. If DB2 variables are set outside of the registry, remote administration of those variables will not be possible, and the workstation will have to be rebooted in order for the variable values to take effect.

Windows 32-bit operating systems have one system environment variable, *DB2INSTANCE*, that can only be set outside the profile registry; however, you are not required to set *DB2INSTANCE*. The DB2 profile registry variable *DB2INSTDEF* may be set in the global level profile to specify the instance name to use if *DB2INSTANCE* is not defined.

To determine the setting of an environment variable, enter:

```
set variable
```

To set system environment variables, do the following:

On Windows 95 and Windows 98:

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 value that you would like this parameter to take, 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 change or append, for example *DB2INSTANCE*.

2 Change the **Value** field to the value that you would like this parameter to take, 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.

Setting Your Environment on UNIX Systems

It is strongly recommended that all DB2 specific registry values be defined in the DB2 profile registry. If DB2 variables are set outside of the registry, remote administration of those variables will not be possible.

On UNIX, you must set the system environment variable *DB2INSTANCE*.

The sample script files, *db2profile* (for Bourne or Korn shell) and *db2cshrc* (for C shell), are provided to help you set the database environment. These scripts are available in the *INSTHOME/sql1lib* directory, where *INSTHOME* is the home directory of the instance owner.

Note: Except for *PATH* and *DB2INSTANCE*, all other DB2-supported variables must be set in the DB2 profile registry. To set variables that are not supported by DB2, define them in your script files, *db2profile* or *db2cshrc*.

An instance owner or SYSADM user may customize these script files for all users of an instance. Alternatively, you can copy and customize a script file, then invoke a script directly or add it to your *.profile* or *.login* file.

To work with another instance for the current session, execute the *db2profile* script file (for Bourne shell or Korn shell) or the *db2cshrc* script file (for C shell) in the *sql1lib* directory of the instance's home directory.

Setting Your Environment on Windows 3.x

The DB2 environment on Windows 3.x is not controlled by profile registries. Instead, Windows 3.x clients define environment keywords in the file *db2.ini* file (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 this 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
```

The following is a sample `db2.ini` file:

```
[DB2 Client Application Enabler]
; This is for my Windows 3.x Client
DB2PATH=C:\SQLLIB\WIN
DB2INSTANCE=DB2
DB2INSTPROF=C:\SQLLIB
DB2TRACEON=N
```

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).

This file is located in the Windows product directory.

On Windows 3.x, the Client Application Enabler for DB2 Version 2.x, Version 4, 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.x, enter the following command:

```
db2set -lr
```

Note: Windows 3.x clients define environment variables in the file `db2.ini`. See “Setting Your Environment on Windows 3.x” on page 161 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 <i>DB2 Connect User's Guide</i> for details.
DB2CODEPAGE	All	Default: derived from the language ID, as specified by the operating system.	Specifies the code page of the data presented to DB2 for database client application. The user should not set <i>db2codepage</i> unless explicitly stated in DB2 documents, or asked to do so by DB2 service. Setting <i>DB2CODEPAGE</i> to a value not supported by the operating system can produce unexpected results. Normally, you do not need to set <i>DB2CODEPAGE</i> 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 applications, 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.

Parameter	Operating System	Values	Description
DB2DBMSADDR	Windows 32-bit operating systems	Default= (not set), value for Windows 95 and Windows 98: (90000000), value for Windows NT: (20000000)	Specifies the default database manager shared memory address in hexadecimal format. If a shared memory address collision occurs, this value can be modified to force the database manager instance to allocate its shared memory at a different address. Use the DB2DMSADDR registry profile variable to assign an address value in the range of x'20000000' to x'B0000000' on Windows NT or x'80000000' to x'B0000000' on Windows 95 using increments of x'40000000'. Use the db2set command to change the shared memory address. For example, to set DB2DBMSADDR to x'84000000, use the following command: db2set DB2DBMSADDR=84
DB2DISCOVERYTIME	OS/2 and Windows 32-bit operating systems	Default: (default time of 40 seconds is used when this parameter is not set)	Specifies the amount of time that a SEARCH discovery will search for DB2 systems.
DB2DMNBCKCTRL	Windows NT	Default: (not set)	If DB2 is installed on a backup domain controller, setting DB2DMNBCKCTRL=DOMAIN_NAME allows DB2 to use the security database on the backup domain controller, thereby reducing LAN traffic. (DOMAIN_NAME must be in upper case.) 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 <i>db2envlist</i> 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 <i>db2envlist</i> , 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: (not set)	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 <i>Embedded SQL Programming Guide</i> for descriptions of how <i>db2include</i> is used in the different precompiled languages.
DB2INSTDEF	OS/2 and Windows 32-bit operating systems	Default=DB2 (on OS/2 and Windows 32-bit operating systems)	Sets the default instance to be used if <i>DB2INSTANCE</i> is not defined.
DB2LIBPATH	UNIX	Default: (not set)	Specifies the value of LIBPATH in the <i>db2libpath</i> 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 <i>db2envlist</i> environment variable, you must also specify the value of LIBPATH in the <i>db2libpath</i> registry value. The db2start executable will then read the value of <i>db2libpath</i> and append this value to the end of the DB2-constructed LIBPATH.
DB2OPTIONS	All	Default: (not set)	Sets command line processor options.

Parameter	Operating System	Values	Description
DB2SLOGON	Windows 3.x	Default (not set), Values: Y, N	Enables a secure logon in DB2 for Windows 3.x . If <i>db2slogon=YES</i> DB2 does not write user IDs and passwords to a file, but instead uses a segment of memory to maintain them. When <i>db2slogon</i> is enabled, the user must logon each time Windows 3.x is started.
DB2TIMEOUT	Windows 3.x and Macintosh	Disabled	Used to control the timeout period for Windows 3.x 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 <i>db2timeout</i> is set to a value between 1 and 30, the default minimum value will be used. If <i>db2timeout</i> is set to a value of 0, or a negative value, the timeout feature is disabled. This feature is disabled by default.
DB2TRACENAME	Windows 3.x and Macintosh	Default (not set)	On Windows 3.x and Macintosh, specifies the name of the file where trace information is stored. The default on Windows 3.x is <i>db2tracename=DB2WIN.TRC</i> , and is saved in your current instance directory (for example, <code>\sql1lib\db2</code>). The default on Macintosh is <i>db2tracename=DB2MAC.TRC</i> . We strongly recommend that you specify the full path name when naming the trace file.
DB2TRACEON	Windows 3.x and Macintosh	Default (not set), Values: Y, N	On Windows 3.x and Macintosh, turn 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 <i>Troubleshooting Guide</i> for information on using the trace facility with DB2 Client Application Enabler.

Parameter	Operating System	Values	Description
DB2TRCFLUSH	Windows 3.x and Macintosh	Default (not set), Values: Y, N	On Windows 3.x and Macintosh, <i>db2trcflush</i> can be used in conjunction with <i>db2traceon=Y</i> . <i>db2trcflush=Y</i> 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 <i>db2trcflush=N</i> . 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	Windows 3.x and Macintosh	Default: (not set), Values: Y, N	Specifies the number of system errors to trace before the client turns off tracing. The default value traces one system error, after which, trace is turned off.
DB2YIELD	Windows 3.x	Default (not set), Values: Y, N	Specifies the behavior of the Windows 3.x client while communicating with a remote server. When <i>N</i> is set, the client will not yield the CPU to other Windows 3.x 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 <i>Y</i> , your system functions as normal. It is recommended that you try to run your application with <i>db2yield=YES</i> . If your system crashes, you will need to set <i>db2yield=NO</i> . 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= <i>DB2INSTDEF</i> on OS/2 and Windows 32-bit operating systems	The environment variable used to specify the instance that is active by default. On UNIX, users must set the environment variable <i>DB2INSTANCE</i> .
DB2INSTPROF	OS/2, Windows 3.x, and Windows 32-bit operating systems	Default (not set)	The environment variable used to specify the location of the instance directory on OS/2 and Windows 32-bit operating systems if different than <i>DB2PATH</i> .

Parameter	Operating System	Values	Description
DB2PATH	OS/2, Windows 3.x, and Windows 32-bit operating systems		The environment variable used to specify the directory where the product is installed on OS/2 and Windows 32-bit operating systems. By default it is set to <code>x:\sql1lib\win</code> on Windows 32-bit operating systems; where <code>x:</code> is the install drive.
DCE Directories			
DB2DIRPATHNAME	OS/2 and UNIX	Default: (not set)	Specifies a temporary override of the <code>DIR_PATH_NAME</code> parameter value in the database manager configuration file. If a directory server is used and the target of a <code>CONNECT</code> statement or <code>ATTACH</code> command is not explicitly cataloged, then the target is concatenated with <i>db2dirpathname</i> (if specified) to form the fully qualified DCE name. Note: The <i>db2dirpathname</i> value has no effect on the instance's global name, which is always identified by the database manager configuration parameters <code>DIR_PATH_NAME</code> and <code>DIR_OBJ_NAME</code> .
DB2CLIENTCOMM	OS/2 and UNIX	Default: (not set)	Specifies a temporary override of the <code>DFT_CLIENT_COMM</code> parameter value in the database manager configuration file. If both <code>DFT_CLIENT_COMM</code> and <i>db2clientcomm</i> are not specified, then the first protocol found in the object is used. If either one or both of them are specified, then only the first matching protocol will be used. In either case, no retry is attempted if the first connect fails.
DB2CLIENTADPT	OS/2	Default: (not set), range: 0-15	Specifies the client adapter number for NETBIOS protocol on OS/2 operating systems. The <i>db2clientadpt</i> value overrides the <code>DFT_CLIENT_ADPT</code> parameter value in the database manager configuration file.
DB2ROUTE	OS/2 and UNIX	Default: (not set)	Specifies the name of the Routing Information Object the client uses when it connects to a database with a different database protocol. The <i>db2route</i> value overrides the <code>ROUTE_OBJ_NAME</code> parameter value in the database manager configuration file.

Note: There are three registry values that are related to ADSM:

DSMI_CONFIG

DSMI_DIR

DSMI_LOG

Refer to the *Administration Guide* for information about ADSM and these values.

Configuration Parameters

Configuration parameters are values that affect the operating characteristics of a database or database management system.

Database manager configuration parameters exist on servers and clients; however, only certain database manager configuration parameters can be set on the client. These parameters are a subset of the database management configuration parameters that can be set on the server.

Database configuration parameters can be set only on the server, or remotely from the Control Center.

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 or Windows 32-bit operating systems, 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 32-bit operating systems:

- 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.

To modify a configuration parameter:

- 1 Select the tab that corresponds to the type of configuration parameter that you want to modify.
- 2 Select the configuration parameter that you want to modify from the **Parameter** window.
- 3 Select the value for this parameter from the **Value** window.
- 4 Click on **OK**.

For more information, refer to the online help.

To control database management configuration parameters on a client instance on UNIX, use the command line processor.



For information on using Client Configuration Assistant, see Chapter 19, "Using the Client Configuration Assistant" on page 251.

For information on using the command line processor, see "Using the Command Line Processor" on page 153.

DB2 CLI/ODBC Configuration Keyword Listing

The keywords are listed in alphabetical order starting with "APPENDAPINAME." They are also divided into categories. In the ODBC Data Source Configuration tool (not available on UNIX platforms) each of these categories is presented on a separate tab in the notebook.

For further information about DB2 CLI/ODBC applications, see the information on your specific operating system in "Platform Specific Details for CLI/ODBC Access" on page 134.

Configuration Keywords by Category

CLI/ODBC Settings General Configuration Keywords

“DBALIAS” on page 180

“PWD” on page 192

“UID” on page 201

Compatibility Configuration Keywords

The **Compatibility** set of options are used to define DB2 behavior. They can be set to ensure that other applications are compatible with DB2.

“DEFERREDPREPARE” on page 182

“DISABLEMULTITHREAD” on page 183

“EARLYCLOSE” on page 183

“TRANSLATEDLL” on page 199

“TRANSLATEOPTION” on page 200

Data Type Configuration Keywords

The **Data Type** set of options are used to define how DB2 reports and handles various data types.

“BITDATA” on page 174

“GRAPHIC” on page 185

“LOBMAXCOLUMNSIZE” on page 188

“LONGDATACOMPAT” on page 188

Enterprise Configuration Keywords

The **Enterprise** set of options are used to maximize the efficiency of connections to large databases.

“CURRENTPACKAGESET” on page 175

“CURRENTSQLID” on page 176

“DBNAME” on page 181

“GATEWAYVERSION” on page 184

“GRANTEELIST” on page 184

“GRANTORLIST” on page 185

“SCHEMALIST” on page 193

“SYSSHEMA” on page 194

“TABLETYPE” on page 195

Environment Configuration Keywords

The **Environment** set of options are used to define the location of various files on the server and client machines.

“CURRENTFUNCTIONPATH” on page 175

“DEFAULTPROCLIBRARY” on page 181

“TEMPDIR” on page 196

Optimization Configuration Keywords

The **Optimization** set of options are used to speed up and reduce the amount of network flow between the CLI/ODBC Driver and the server.

“DB2DEGREE” on page 177

“DB2ESTIMATE” on page 178

“DB2EXPLAIN” on page 179

“DB2OPTIMIZATION” on page 180

“KEEPSTATEMENT” on page 187

“OPTIMIZEFORNROWS” on page 190

“UNDERSCORE” on page 201

Service Configuration Keywords

The **Service** set of options are used to help in troubleshooting problems with CLI/ODBC connections. Some options can also be used by programmers to gain a better understanding of how their CLI programs are translated into calls to the server.

“APPENDAPINAME” on page 173

“IGNOREWARNINGS” on page 186

“PATCH1” on page 191

“PATCH2” on page 191

“POPUPMESSAGE” on page 192

“SQLSTATEFILTER” on page 193

“TRACE” on page 197

“TRACEFILENAME” on page 197

“TRACEFLUSH” on page 198

“TRACEPATHNAME” on page 198

“WARNINGLIST” on page 202

Transaction Configuration Keywords

The **Transaction** set of options are used to control and speed up SQL statements used in the application.

“ASYNCENABLE” on page 173

"CONNECTTYPE" on page 174
"CURSORHOLD" on page 177
"KEEPCONNECT" on page 187
"MAXCONN" on page 189
"MODE" on page 189
"MULTICONNECT" on page 190
"SYNCPOINT" on page 194
"TXNISOLATION" on page 200

Configuration Keyword Descriptions

APPENDAPINAME

Keyword Description: Append the CLI/ODBC function name which generated an error to the error message.

db2cli.ini Keyword Syntax: APPENDAPINAME = 0 | 1

Default Setting: Do NOT display DB2 CLI function name.

DB2 CLI/ODBC Settings Tab: Service

Usage Notes:

The DB2 CLI function (API) name that generated an error is appended to the error message retrieved using `SQLGetDiagRec()` or `SQLError()`. The function name is enclosed in curly braces { }.

For example,

```
[IBM][CLI Driver]" CLIxxxx: < text >  
SQLSTATE=XXXXX {SQLGetData}"
```

0 = do NOT append DB2 CLI function name (default)

1 = append the DB2 CLI function name

This keyword is only useful for debugging.

ASYNCEENABLE

Keyword Description: Enable or disable the ability to execute queries asynchronously.

db2cli.ini Keyword Syntax: ASYNCEENABLE = 1 | 0

Default Setting: Execute queries asynchronously.

DB2 CLI/ODBC Settings Tab: Transaction

Usage Notes:

This option allows you to enable or disable the ability to execute queries asynchronously. This only benefits applications that were written to take advantage of this feature. Disable it only if your application does not function properly when enabled. It is placed in the data source specific section of the `db2cli.ini` file.

- 1 = Execute queries asynchronously (default)
- 0 = Queries not executed asynchronously

Note: The CLI/ODBC driver will act as it did with previous versions of DB2 that did not support asynchronous ODBC.

BITDATA

Keyword Description: Specify whether binary data types are reported as binary or character data types.

db2cli.ini Keyword Syntax: BITDATA = 1 | 0

Default Setting: Report FOR BIT DATA and BLOB data types as binary data types.

DB2 CLI/ODBC Settings Tab: Data Type

Usage Notes:

This option allows you to specify whether ODBC binary data types (SQL_BINARY, SQL_VARBINARY, SQL_LONGVARBINARY, and SQL_BLOB), are reported as binary type data. IBM DBMSs support columns with binary data types by defining CHAR, VARCHAR, and LONG VARCHAR columns with the FOR BIT DATA attribute. DB2 Universal Database will also support binary data via the BLOB data type (in this case it is mapped to a CLOB data type).

Users may also need to set this option if they are using a DB2 Version 1 application that retrieves (LONG) (VAR)CHAR data into SQL_C_CHAR buffer. In DB2 Version 1, data is moved into the SQL_C_CHAR buffer unchanged; starting in DB2 Version 2, the data is converted into the ASCII representation of each hexadecimal nibble.

Only set BITDATA = 0 if you are sure that all columns defined as FOR BIT DATA or BLOB contain only character data, and the application is incapable of displaying binary data columns.

- 1 = report FOR BIT DATA and BLOB data types as binary data types (default).
- 0 = report FOR BIT DATA and BLOB data types as character data types.

CONNECTTYPE

Keyword Description: Remote or Distributed unit of work.

db2cli.ini Keyword Syntax: CONNECTTYPE = 1 | 2

Default Setting: Remote unit of work

DB2 CLI/ODBC Settings Tab: Transaction

See Also: "SYNCPOINT" on page 194

Usage Notes:

This option allows you to specify the default connect type.

- 1 = Remote unit of work. Multiple concurrent connections, each with its own commit scope. The concurrent transactions are not coordinated. (default)
- 2 = Distributed unit of work. Coordinated connections where multiple databases participate under the same distributed unit of work. This setting works in conjunction with the SYNCPOINT setting to determine if a Transaction Manager should be used.

CURRENTFUNCTIONPATH

Keyword Description: Specify the schema used to resolve function references and data type references in dynamic SQL statements.

db2cli.ini Keyword Syntax: CURRENTFUNCTIONPATH = *current_function_path*

Default Setting: See description below.

DB2 CLI/ODBC Settings Tab: Environment

Usage Notes:

This keyword defines the path used to resolve function references and data type references that are used in dynamic SQL statements. It contains a list of one or more schema-names, where schema-names are enclosed in double quotes and separated by commas.

The default value is "SYSIBM","SYSFUN",X where X is the value of the USER special register delimited by double quotes. The schema SYSIBM does not need to be specified. If it is not included in the function path, then it is implicitly assumed as the first schema.

This keyword is used as part of the process for resolving unqualified function references that may have been defined in a schema name other than the current user's schema. The order of the schema names determines the order in which the function names will be resolved. For more information on function resolution, refer to the *SQL Reference*.

CURRENTPACKAGESET

Keyword Description: Issue "SET CURRENT PACKAGESET schema" after every connect.

db2cli.ini Keyword Syntax: CURRENTPACKAGESET = *schema name*

Default Setting: The clause is not appended.

DB2 CLI/ODBC Settings Tab: This keyword cannot be set using the CLI/ODBC Settings notebook. The db2cli.ini file must be modified directly to make use of this keyword.

Only Applicable when: connecting to a DB2 for MVS/ESA v4.1 or later database.

Usage Notes:

This option will issue the command "SET CURRENT PACKAGESET schema" after every connect to a DB2 for MVS/ESA v4.1 or later database. By default this clause is not appended.

This statement sets the schema name (collection identifier) that will be used to select the package to use for subsequent SQL statements.

CLI/ODBC applications issue dynamic SQL statements. Using this option you can control the privileges used to run these statements:

- Choose a schema to use when running SQL statements from CLI/ODBC applications.

- Ensure the objects in the schema have the desired privileges and then rebind accordingly.

- Set the CURRENTPACKAGESET option to this schema.

The SQL statements from the CLI/ODBC applications will now run under the specified schema and use the privileges defined there.

Refer to the *SQL Reference* for more information on the SET CURRENT PACKAGESET command.

CURRENTSQLID

Keyword Description: Specify the ID used in a SET CURRENT SQLID statement sent to the DBMS upon a successful connect.

db2cli.ini Keyword Syntax: CURRENTSQLID = *current_sqlid*

Default Setting: No statement is issued.

DB2 CLI/ODBC Settings Tab: Enterprise

Only Applicable when: connecting to those DB2 DBMS's where SET CURRENT SQLID is supported (such as DB2 for MVS/ESA).

Usage Notes:

Upon a successful connect, if this option is set, a SET CURRENT SQLID statement is sent to the DBMS. This allows the end user and the application to name SQL objects without having to qualify them by schema name.

CURSORHOLD

Keyword Description: Effect of a transaction completion on open cursors.

db2cli.ini Keyword Syntax: CURSORHOLD = 1 | 0

Default Setting: Selected--Cursors are not destroyed.

DB2 CLI/ODBC Settings Tab: Transaction

Usage Notes:

This option controls the effect of a transaction completion on open cursors.

1 = cursor hold, the cursors are not destroyed when the transaction is committed (default).

0 = cursor no hold, the cursors are destroyed when the transaction is committed.

Note: Cursors are always destroyed when transactions are rolled back.

This option affects the result returned by `SQLGetInfo()` when called with `SQL_CURSOR_COMMIT_BEHAVIOR` or `SQL_CURSOR_ROLLBACK_BEHAVIOR`. The value of `CURSORHOLD` is ignored if connecting to DB2 for VSE & VM where cursor `with hold` is not supported.

You can use this option to tune performance. It can be set to cursor no hold (0) if you are sure that your application:

1. Does not have behavior that is dependent on the `SQL_CURSOR_COMMIT_BEHAVIOR` or the `SQL_CURSOR_ROLLBACK_BEHAVIOR` information returned via `SQLGetInfo()`, and
2. Does not require cursors to be preserved from one transaction to the next.

The DBMS will operate more efficiently, as resources no longer need to be maintained after the end of a transaction.

DB2DEGREE

Keyword Description: Set the degree of parallelism for the execution of SQL statements.

db2cli.ini Keyword Syntax: DB2DEGREE = 0 | integer value from 1 to 32767 | ANY

Default Setting: No SET CURRENT DEGREE statement is issued.

DB2 CLI/ODBC Settings Tab: Optimization

Only Applicable when: connecting to a cluster database system.

Usage Notes:

This option only applies to a DB2 Version 5.2 or later server. If the value specified is anything other than 0 (the default) then DB2 CLI will issue the following SQL statement after a successful connection:

```
SET CURRENT DEGREE value
```

This specifies the degree of parallelism for the execution of the SQL statements. The database manager will determine the degree of parallelism if you specify ANY.

For more information, see the SET CURRENT DEGREE statement in the *SQL Reference*.

DB2ESTIMATE

Keyword Description: Threshold for displaying CLI optimizer estimates after SQL query statement preparation.

db2cli.ini Keyword Syntax: DB2ESTIMATE = 0 | large positive number

Default Setting: Estimates are not returned.

DB2 CLI/ODBC Settings Tab: Optimization

Only Applicable when: a GUI application accesses a DB2 Version 2 or later server.

Usage Notes:

This option determines whether DB2 CLI will display a dialog box to report estimates returned by the DB2 optimizer at the end of SQL query statement preparation.

0 = Estimates are not returned (default).

large positive number = The threshold above which DB2 CLI will display the window to report estimates. This value is compared against the SQLERRD(4) field in the SQLCA associated with the PREPARE. If the value in SQLERRD(4) is greater than DB2ESTIMATE, the estimates window will appear.

The graphic window will display the optimizer estimates, along with push buttons to allow users to choose whether they wish to continue with subsequent execution of this query or cancel it.

The recommended value for DB2ESTIMATE is 60000.

This option is only relevant when connecting to a DB2 version 2 or later database. In order for the window to appear, the application must have a graphical interface.

If this option is used then the DB2 CLI/ODBC option DEFERREDPREPARE will be considered off.

DB2EXPLAIN

Keyword Description: Determines whether Explain snapshot and/or Explain table information will be generated by the server.

db2cli.ini Keyword Syntax: DB2EXPLAIN = 0 | 1 | 2 | 3

Default Setting: Neither Explain snapshot nor Explain table information will be generated by the server.

DB2 CLI/ODBC Settings Tab: Optimization

Usage Notes:

This keyword determines whether Explain snapshot and/or Explain table information will be generated by the server.

0 = both off (default)

A 'SET CURRENT EXPLAIN SNAPSHOT=NO' and a 'SET CURRENT EXPLAIN MODE=NO' statement will be sent to the server to disable both the Explain snapshot and the Explain table information capture facilities.

1 = Only Explain snapshot facility on

A 'SET CURRENT EXPLAIN SNAPSHOT=YES' and a 'SET CURRENT EXPLAIN MODE=NO' statement will be sent to the server to enable the Explain snapshot facility, and disable the Explain table information capture facility.

2 = Only Explain table information capture facility on

A 'SET CURRENT EXPLAIN MODE=YES' and a 'SET CURRENT EXPLAIN SNAPSHOT=NO' will be sent to the server to enable the Explain table information capture facility and disable the Explain snapshot facility.

3 = Both on

A 'SET CURRENT EXPLAIN MODE=YES' and a 'SET CURRENT EXPLAIN SNAPSHOT=YES' will be sent to the server to enable both the Explain snapshot and the Explain table information capture facilities.

Explain information is inserted into Explain tables, which must be created before the Explain information can be generated. For more information on these tables, refer to the *SQL Reference*.

The current authorization ID must have INSERT privilege for the Explain tables.

Option 1 is only valid when connecting to a DB2 Common Server version 2.1.0 or later database; options 2 and 3 when connecting to a DB2 Common Server version 2.1.1 or later database.

DB2OPTIMIZATION

Keyword Description: Set the query optimization level.

db2cli.ini Keyword Syntax: DB2OPTIMIZATION = *integer value from 0 to 9*

Default Setting: No SET CURRENT QUERY OPTIMIZATION statement issued.

DB2 CLI/ODBC Settings Tab: Optimization

Only Applicable when: when connecting to a DB2 Version 2 server or later.

Usage Notes:

If this option is set then DB2 CLI will issue the following SQL statement after a successful connection:

```
SET CURRENT QUERY OPTIMIZATION positive number
```

This specifies the query optimization level at which the optimizer should operate the SQL queries. Refer to the *SQL Reference* for the allowable optimization levels.

DBALIAS

Keyword Description: Enables Data Source Names greater than 8 characters.

db2cli.ini Keyword Syntax: DBALIAS = *dbalias*

Default Setting: Use the DB2 database alias as the ODBC Data Source Name.

DB2 CLI/ODBC Settings Tab: CLI/ODBC Settings General

Usage Notes:

This keyword allows for Data Source Names of greater than 8 single byte characters. The Data Source Name (DSN) is the name, enclosed in square brackets, that denotes the section header in the `db2cli.ini` file (on platforms where this is an ASCII file). Typically, this section header is the database alias name which has a maximum length of 8 bytes. A user who wishes to refer to the data source with a longer, more meaningful name, can place the longer name in the section header, and set this keyword value to the database alias used on the CATALOG command. Here is an example:

```
; The much longer name maps to an 8 single byte character dbalias  
[MyMeaningfulName]  
DBALIAS=DB2DBT1
```

The end user can specify `[MyMeaningfulName]` as the name of the data source on connect while the actual database alias is `DB2DBT1` .

In a 16-bit Windows ODBC environment, under the `[ODBC DATA SOURCES]` entry in the `ODBC.INI` file, the following line must also be updated with the long alias name (*dbname*).

< alias > IBM DB2 ODBC DRIVER

DBNAME

Keyword Description: Specify the database name to reduce the time it takes for the application to query MVS table information.

db2cli.ini Keyword Syntax: DBNAME = *dbname*

Default Setting: Don't filter on the DBNAME column.

DB2 CLI/ODBC Settings Tab: Enterprise

Only Applicable when: connecting to DB2 for MVS/ESA.

See Also: "SCHEMALIST" on page 193, "TABLETYPE" on page 195

Usage Notes:

This option is only used when connecting to DB2 for MVS/ESA, and only if (*base*) table catalog information is requested by the application. If a large number of tables exist in the DB2 for MVS/ESA subsystem, a *dbname* can be specified to reduce the time it takes for the application to query table information, and reduce the number of tables listed by the application.

This value maps to the `DBNAME` column in the DB2 for MVS/ESA system catalog tables. If no value is specified, or if views, synonyms, system tables, or aliases are also specified via `TABLETYPE`, only table information will be restricted; views, aliases, and synonyms are not restricted with `DBNAME`. It can be used in conjunction with `SCHEMALIST`, and `TABLETYPE` to further limit the number of tables for which information will be returned.

DEFAULTPROCLIBRARY

Keyword Description: Set default stored procedure library.

db2cli.ini Keyword Syntax: DEFAULTPROCLIBRARY = < *full path name* >

Default Setting: Do not add a default stored procedure library to stored procedure calls.

DB2 CLI/ODBC Settings Tab: Environment

Only Applicable when: application is not using the stored procedure catalog table.

Usage Notes:

This option should only be used on a temporary basis; the stored procedure catalog table should be used instead. See the *SQL Reference* for more information.

The library pointed to by this option will be used in all stored procedure calls that do not already explicitly specify a library. Because you are specifying a location on the server

machine, you must use the path format of that operating system, not of the client. For more information, see the CALL statement in the *SQL Reference*.

For instance, if the stored procedures are located on the server in the library file `d:\terry\proclib\comstor`, you could set `DEFAULTPROCLIBRARY` to `d:\terry\proclib\comstor`, then call the stored procedure `func` without specifying a library. The resulting SQL statement sent would be:

```
CALL d:\terry\proclib\comstor!func
```

DEFERREDPREPARE

Keyword Description: Minimize network flow by combining the PREPARE request with the corresponding execute request.

db2cli.ini Keyword Syntax: DEFERREDPREPARE = 0 | 1

Default Setting: The prepare request will be delayed until the execute request is sent.

DB2 CLI/ODBC Settings Tab: Compatibility

Not Applicable when: DB2ESTIMATE is set.

Usage Notes:

Defers sending the PREPARE request until the corresponding execute request is issued. The two requests are then combined into one command/reply flow (instead of two) to minimize network flow and to improve performance.

The default behavior has changed from DB2 version 2. Deferred prepare is now the default and must be explicitly turned off if required.

0 = Disable deferred prepare. The PREPARE request will be executed the moment it is issued.

1 (default) = Enable deferred prepare. Defer the execution of the PREPARE request until the corresponding execute request is issued.

If the target DB2 Common Server database or the DDCS gateway does not support deferred prepare, the client disables deferred prepare for that connection.

Note: When deferred prepare is enabled, the row and cost estimates normally returned in the `SQLERRD(3)` and `SQLERRD(4)` of the `SQLCA` of a PREPARE statement may become zeros. This may be of concern to users who want to use these values to decide whether or not to continue the SQL statement.

This option is turned off if the CLI/ODBC option `DB2ESTIMATE` is set to a value other than zero.

DISABLEMULTITHREAD

Keyword Description: Disable Multithreading.

db2cli.ini Keyword Syntax: DISABLEMULTITHREAD = 0 | 1

Default Setting: Multithreading is enabled.

DB2 CLI/ODBC Settings Tab: Compatibility

Usage Notes:

The CLI/ODBC driver is capable of supporting multiple concurrent threads.

This option is used to enable or disable multi-thread support.

0 = Multithreading is enabled (default).

1 = Disable Multithreading.

If multithreading is disabled then all calls for all threads will be serialized at the process level. Use this setting for multithreaded applications that require the serialized behavior of DB2 Version 2.

(This option is contained in the Common section of the initialization file and therefore applies to all connections to DB2.)

EARLYCLOSE

Keyword Description: Should the cursor associated with the connection be closed early by the DB2 server when it encounters the end of the result set?

db2cli.ini Keyword Syntax: EARLYCLOSE = 1 | 0

Default Setting: EARLYCLOSE behavior is on.

DB2 CLI/ODBC Settings Tab: Compatibility

Usage Notes:

This option specifies whether or not the temporary cursor on the server can be automatically closed, without closing the cursor on the client, when the last record is sent to the client.

0 = Do not close the temporary cursor on the server early.

1 = Close the temporary cursor on the server early (default).

This saves the CLI/ODBC driver a network request by not issuing the statement to explicitly close the cursor because it knows that it has already been closed.

Having this option on will speed up applications that make use of many small result sets.

The EARLYCLOSE feature is not used if either:
The statement disqualifies for blocking.

The cursor type is anything other than SQL_CURSOR_FORWARD_ONLY.

Note: Although this option can be set at any time, the option value used is the one that exists when the statement is executed (when the cursor is opened).

GATEWAYVERSION

Keyword Description: Specify DB2 Connect or DB2 DDCS gateway version being used.

db2cli.ini Keyword Syntax: GATEWAYVERSION = *gateway version*

Default Setting: 5

DB2 CLI/ODBC Settings Tab: This keyword cannot be set using the CLI/ODBC Settings notebook. The db2cli.ini file must be modified directly to make use of this keyword.

Only Applicable when: connecting to a data source through a DB2 Connect or DB2 DDCS gateway.

Usage Notes:

This option is used to indicate to the DB2 CLI driver which version of a DB2 Connect or DB2 DDCS gateway is being used. The CLI driver can then use this information to maximize its interaction with the data source (supporting stored procedures that return multiple result sets, for instance).

5 = Indicates that a version 5 DB2 Connect gateway is being used (default).

2 = Indicates that a version 2 DB2 DDCS gateway is being used.

GRANTEELIST

Keyword Description: Reduce the amount of information returned when the application gets a list of table or column privileges.

db2cli.ini Keyword Syntax: GRANTEELIST = " 'userID1', 'userID2',... 'userIDn' "

Default Setting: Do not filter the results.

DB2 CLI/ODBC Settings Tab: Enterprise

See Also: "GRANTORLIST" on page 185

Usage Notes:

This option can be used to reduce the amount of information returned when the application gets a list of privileges for tables in a database, or columns in a table. The list of authorization IDs specified is used as a filter; the only tables or columns that are returned are those with privileges that have been granted *TO* those IDs.

Set this option to a list of one or more authorization IDs that have been granted privileges, delimited with single quotes, and separated by commas. The entire string must also be enclosed in double quotes. For example:

```
GRANTEELIST=" 'USER1' , 'USER2' , 'USER8' "
```

In the above example, if the application gets a list of privileges for a specific table, only those columns that have a privilege granted *TO* USER1, USER2, or USER8 would be returned.

GRANTORLIST

Keyword Description: Reduce the amount of information returned when the application gets a list of table or column privileges.

db2cli.ini Keyword Syntax: GRANTORLIST = " 'userID1', 'userID2',... 'userIDn' "

Default Setting: Do not filter the results.

DB2 CLI/ODBC Settings Tab: Enterprise

See Also: "GRANTEELIST" on page 184

Usage Notes:

This option can be used to reduce the amount of information returned when the application gets a list of privileges for tables in a database, or columns in a table. The list of authorization IDs specified is used as a filter; the only tables or columns that are returned are those with privileges that have been granted *BY* those IDs.

Set this option to a list of one or more authorization IDs that have granted privileges, delimited with single quotes, and separated by commas. The entire string must also be enclosed in double quotes. For example:

```
GRANTORLIST=" 'USER1' , 'USER2' , 'USER8' "
```

In the above example, if the application gets a list of privileges for a specific table, only those columns that have a privilege granted *BY* USER1, USER2, or USER8 would be returned.

GRAPHIC

Keyword Description: Controls whether DB2 CLI reports the IBM GRAPHIC (double byte character support) as one of the supported data types.

db2cli.ini Keyword Syntax: GRAPHIC = 0 | 1 | 2 | 3

Default Setting: GRAPHIC is not returned as a supported data type.

DB2 CLI/ODBC Settings Tab: Data Type

Usage Notes:

This option controls how two related pieces of information are returned by the application:

Whether DB2 CLI reports the IBM GRAPHIC (double byte character support) as one of the supported data types when `SQLGetTypeInfo()` is called.

`SQLGetTypeInfo()` lists the data types supported by the DB2 database in the current connection.

What unit is used to report the length of graphic columns. This applies to all DB2 CLI/ODBC functions that return length/precision either on the output argument or as part of the result set.

0 = Do not report IBM GRAPHIC data type as a supported type. Length of graphic columns returned as number of DBCS characters. (default)

1 = Report IBM GRAPHIC data type as supported. Length of graphic columns returned as number of DBCS characters.

2 = Do not report IBM GRAPHIC data type as a supported type. Length of graphic columns returned as number of bytes. (This is needed for **Microsoft Access** 1.1-J** and **Microsoft Query**-J**.)

3 = Settings 1 and 2 combined. IBM GRAPHIC data type reported as supported. Length of graphic columns returned as number of bytes.

The default is that GRAPHIC is not returned since many off the shelf applications do not recognize this data type and cannot provide proper handling.

IGNOREWARNINGS

Keyword Description: Ignore Warnings.

db2cli.ini Keyword Syntax: IGNOREWARNINGS = 0 | 1

Default Setting: Warnings are returned as normal.

DB2 CLI/ODBC Settings Tab: Service

See Also: "WARNINGLIST" on page 202

Usage Notes:

On rare occasions an application will not correctly handle warning messages. This option can be used to indicate that warnings from the database manager are not to be passed on to the application.

0 = Warnings reported as usual (default).

1 = Database manager warnings are ignored, `SQL_SUCCESS` is returned.

Warnings from the DB2 CLI/ODBC driver are still returned; many are required for normal operation.

Although this option can be used on its own, it can also be used in conjunction with the WARNINGLIST CLI/ODBC configuration keyword.

KEEPCONNECT

Keyword Description: Number of connections to cache.

db2cli.ini Keyword Syntax: KEEPCONNECT = 0 | **positive integer**

Default Setting: Do not cache connections.

DB2 CLI/ODBC Settings Tab: Transaction

Usage Notes:

0 = Do not cache database connections (default).

Setting this option to a value greater than zero can speed up applications that constantly connect to and disconnect from the same database using the same connection information.

Instead of closing the connection each time, then re-opening it again, the CLI/ODBC driver will keep the connection open and cache the connection information. When the request to connect to the same database occurs a second time, the existing connection is used. This saves the time, resources, and network flow to close the first connection, as well as to re-open the second connection.

The value set for this option indicates the number of database connections to cache. Although the maximum is limited only by system resources, usually a value of 1 or 2 is sufficient for applications that will benefit at all from this behavior.

KEEPSTATEMENT

Keyword Description: Number of statement handles to cache.

db2cli.ini Keyword Syntax: KEEPSTATEMENT = 5 | **positive integer**

Default Setting: Cache 5 statement handles.

DB2 CLI/ODBC Settings Tab: Optimization

Usage Notes:

By default, the memory required for 5 statement handles is cached. When a statement handle is closed, the memory used for that handle is not deallocated but is instead used when the next statement handle is allocated.

The value set for this option determines how many statement handles are cached. It can be set to less than 5 to explicitly reduce the amount of memory used by the statement cache. It can be increased above 5 to improve performance for applications that open, close, and then re-open large sets of statements.

The maximum number of cached statement handles is determined by system resources.

LOBMAXCOLUMNSIZE

Keyword Description: Override default COLUMN_SIZE for LOB data types.

db2cli.ini Keyword Syntax: LOBMAXCOLUMNSIZE = *integer greater than zero*

Default Setting: 2 Gigabytes (1G for DBCLOB)

DB2 CLI/ODBC Settings Tab: Data Type

Only Applicable when: LONGDATACOMPAT option is used.

See Also: “LONGDATACOMPAT”

Usage Notes:

This will override the 2 Gigabyte (1G for DBCLOB) value that is returned by `SQLGetTypeInfo()` for the COLUMN_SIZE column for SQL_CLOB, SQL_BLOB, and SQL_DBCLOB SQL data types. Subsequent CREATE TABLE statements that contain LOB columns will use the column size value you set here instead of the default.

LONGDATACOMPAT

Keyword Description: Report LOBs as long data types or as large object types.

db2cli.ini Keyword Syntax: LONGDATACOMPAT = 0 | 1

Default Setting: Reference LOB data types as large object types.

DB2 CLI/ODBC Settings Tab: Data Type

See Also: “LOBMAXCOLUMNSIZE”

Usage Notes:

This option indicates to DB2 CLI what data type the application expects when working with a database with large object (LOB) columns.

Database data type	Large Objects (0--Default)	Long Data Types (1)
CLOB	SQL_CLOB	SQL_LONGVARCHAR
BLOB	SQL_BLOB	SQL_LONGVARBINARY
DBCLOB	SQL_DBCLOB	SQL_LONGVARGRAPHIC

This option is useful when running ODBC applications that cannot handle the large object data types.

The DB2 CLI/ODBC option LOBMAXCOLUMNSIZE can be used in conjunction with this option to reduce the default size declared for the data.

MAXCONN

Keyword Description: Maximum number of connections allowed for each application.

db2cli.ini Keyword Syntax: MAXCONN = 0 | **positive number**

Default Setting: As many connections as permitted by system resources.

DB2 CLI/ODBC Settings Tab: Transaction

Usage Notes:

This option is used to specify the maximum number of connections allowed for each CLI/ODBC application. This can be used as a governor for the maximum number of connections an administrator may wish to restrict each application to open. A value of 0 may be used to represent *no limit*; that is, an application is allowed to open up as many connections as permitted by the system resources.

On OS/2 and WIN32 platforms (Windows NT and Windows 95), if the NetBIOS protocol is in use, this value corresponds to the number of connections (NetBIOS sessions) that will be concurrently set up by the application. The range of values for OS/2 NetBIOS is 1 to 254. Specifying 0 (the default) will result in 5 *reserved* connections. *Reserved NetBIOS sessions* cannot be used by other applications. The number of connections specified by this parameter will be applied to any adapter that the DB2 NetBIOS protocol uses to connect to the remote server (adapter number is specified in the node directory for a NetBIOS node).

MODE

Keyword Description: Default connect mode.

db2cli.ini Keyword Syntax: MODE = SHARE | **EXCLUSIVE**

Default Setting: SHARE

DB2 CLI/ODBC Settings Tab: Transaction

Not Applicable when: connecting to a DRDA database.

Usage Notes:

Sets the CONNECT mode to either SHARE or EXCLUSIVE. If a mode is set by the application at connect time, this value is ignored. The default is SHARE.

Note: EXCLUSIVE is not permitted for DRDA connections. Refer to the *SQL Reference* for more information on the CONNECT statement.

MULTICONNECT

Keyword Description: How SQLConnect() requests are mapped to physical database connections.

db2cli.ini Keyword Syntax: MULTICONNECT = 0 | 1

Default Setting: Each SQLConnect () request by the application will result in a physical database connection.

DB2 CLI/ODBC Settings Tab: Transaction

Usage Notes:

This option is used to specify how SQLConnect () requests are mapped to physical database connections.

1 = Connections are not shared, multiple connections are used (default) -- Each SQLConnect () request by the application will result in a physical database connection.

0 = Connections are mapped to one physical connection, one connection is used -- All connections for the application are mapped to one physical connection. This may be used if the ODBC application runs out of file handles because it uses so many connections. It can also be useful for some applications that only read data from the database, and for some applications that use autocommit.

Note: If MULTICONNECT is set off then all statements are executed on the same connection and therefore in the same transaction. This means that a rollback will roll back ALL statements on all connections. Be sure that the application is designed to work with MULTICONNECT off before doing so or the application may not operate correctly.

OPTIMIZEFORNROWS

Keyword Description: Append "OPTIMIZE FOR n ROWS" clause to every select statement.

db2cli.ini Keyword Syntax: OPTIMIZEFORNROWS = *integer*

Default Setting: The clause is not appended.

DB2 CLI/ODBC Settings Tab: Optimization

Usage Notes:

This option will append the "OPTIMIZE FOR n ROWS" clause to every select statement, where n is an integer larger than 0. If set to 0 (the default) this clause will not be appended.

For more information on the effect of the OPTIMIZE FOR n ROWS clause, refer to the *Administration Guide*.

PATCH1

Keyword Description: Use work-arounds for known problems with ODBC applications.

db2cli.ini Keyword Syntax: PATCH1 = { 0 | 1 | 2 | 4 | 8 | 16 | ... }

Default Setting: Use no work-arounds.

DB2 CLI/ODBC Settings Tab: Service

See Also: "PATCH2"

Usage Notes:

This keyword is used to specify a work-around for known problems with ODBC applications. The value specified can be for none, one, or multiple work-arounds. The patch values specified here are used in conjunction with any PATCH2 values that may also be set.

Using the DB2 CLI/ODBC Settings notebook you can select one or more patches to use. If you set the values in the `db2cli.ini` file itself and want to use multiple patch values then simply add the values together to form the keyword value. For example, if you want the patches 1, 4, and 8, then specify `PATCH1=13`.

0 = No work around (default)

The DB2 CLI/ODBC Settings notebook has a list of values. Select the Service folder in the DB2 folder for information on how to update this list of values. This information is also contained in the `README` file (there will be no such section in the `README` if there are no current patch values for that platform).

PATCH2

Keyword Description: Use work-arounds for known problems with CLI/ODBC applications.

db2cli.ini Keyword Syntax: PATCH2 = "*patch value 1, patch value 2, patch value 3, ...*"

Default Setting: Use no work-arounds

DB2 CLI/ODBC Settings Tab: This keyword cannot be set using the CLI/ODBC Settings notebook. The `db2cli.ini` file must be modified directly to make use of this keyword.

See Also: "PATCH1"

Usage Notes:

This keyword is used to specify a work-around for known problems with CLI/ODBC applications. The value specified can be for none, one, or multiple work-arounds. The patch values specified here are used in conjunction with any PATCH1 values that may also be set.

When specifying multiple patches, the values are specified in a comma delimited string (unlike the PATCH1 option where the values are added together and the sum is used).

0 = No work around (default)

To set PATCH2 values 3, 4 and 8 you would specify:

```
PATCH2=" 3, 4, 8"
```

The PATCH2 values are contained in the README file (there will be no such section in the README if there are no current patch values for that platform).

POPUPMESSAGE

Keyword Description: Pop up a message box every time CLI/ODBC generates an error.

db2cli.ini Keyword Syntax: POPUPMESSAGE = 0 | 1

Default Setting: Do not display message box.

DB2 CLI/ODBC Settings Tab: Service

Only Applicable when: running OS/2 or Windows applications.

See Also: "SQLSTATEFILTER" on page 193

Usage Notes:

Pops up a message box every time DB2 CLI generates an error that can be retrieved using `SQLGetDiagRec()` or `SQLERROR()`. Useful for debugging applications that do not report messages to users.

0 = do NOT display message box (default)

1 = display message box

PWD

Keyword Description: Define default password.

db2cli.ini Keyword Syntax: PWD = *password*

Default Setting: None

DB2 CLI/ODBC Settings Tab: CLI/ODBC Settings General

Usage Notes:

This *password* value is used if a password is not provided by the application at connect time.

It is stored as plain text and is therefore not secure.

SCHEMALIST

Keyword Description: Restrict schemas used to query table information.

db2cli.ini Keyword Syntax: SCHEMALIST = " 'schema1', 'schema2',... 'schemaN' "

Default Setting: None

DB2 CLI/ODBC Settings Tab: Enterprise

Usage Notes:

SCHEMALIST is used to provide a more restrictive default, and therefore improve performance, for those applications that list every table in the DBMS.

If there are a large number of tables defined in the database, a schema list can be specified to reduce the time it takes for the application to query table information, and reduce the number of tables listed by the application. Each schema name is case-sensitive, must be delimited with single quotes, and separated by commas. The entire string must also be enclosed in double quotes. For example:

```
SCHEMALIST=" 'USER1 ' , 'USER2 ' , 'USER3 ' "
```

For DB2 for MVS/ESA, CURRENT SQLID can also be included in this list, but without the single quotes, for example:

```
SCHEMALIST=" 'USER1 ' , CURRENT SQLID , 'USER3 ' "
```

The maximum length of the string is 256 characters.

This option can be used in conjunction with DBNAME and TABLETYPE to further limit the number of tables for which information will be returned.

SQLSTATEFILTER

Keyword Description: Do not pop up an error message for the defined SQLSTATES.

db2cli.ini Keyword Syntax: SQLSTATEFILTER = " 'XXXXX', 'YYYYY', ... "

Default Setting: None

DB2 CLI/ODBC Settings Tab: Service

Only Applicable when: POPUPMESSAGE option is turned on.

See Also: "POPUPMESSAGE" on page 192

Usage Notes:

Use in conjunction with the POPUPMESSAGE option. This prevents DB2 CLI from displaying errors that are associated with the defined states.

Each SQLSTATE must be in upper case, delimited with single quotes and separated by commas. The entire string must also be enclosed in double quotes. For example:

```
SQLSTATEFILTER=" 'HY1 9 ', ' 15 4', ' 15 8' "
```

SYNCPPOINT

Keyword Description: How commits and rollbacks are coordinated among multiple database (DUOW) connections.

db2cli.ini Keyword Syntax: SYNCPPOINT = 1 | 2

Default Setting: 1 Phase commit.

DB2 CLI/ODBC Settings Tab: Transaction

Only Applicable when: default connect type set to Coordinated Connections (CONNECTTYPE=2)

See Also: "CONNECTTYPE" on page 174

Usage Notes:

Use this option to specify how commits and rollbacks will be coordinated among multiple database (DUOW) connections. It is only relevant when the default connect type is set to Coordinated connections (CONNECTTYPE = 2).

1 = ONEPHASE (default)

A Transaction Manager is not used to perform two phase commit but one phase commit is used to commit the work done by each database in a multiple database transaction.

2 = TWOPHASE

A Transaction Manager is required to coordinate two phase commits among those databases that support this.

SYSSCHEMA

Keyword Description: Indicates an alternative schema to be searched in place of the SYSIBM (or SYSTEM, QSYS2) schemas.

db2cli.ini Keyword Syntax: SYSSCHEMA = *syschema*

Default Setting: No alternatives specified.

DB2 CLI/ODBC Settings Tab: Enterprise

Usage Notes:

This option indicates an alternative schema to be searched in place of the SYSIBM (or SYSTEM, QSYS2) schemas when the DB2 CLI and ODBC Catalog Function calls are issued to obtain system catalog information.

Using this schema name the system administrator can define a set of views consisting of a subset of the rows for each of the following system catalog tables:

DB2 Universal Database	DB2 for MVS/ESA	DB2 for VSE & VM	OS/400	DB2 for AS/400
SYSTABLES	SYSTABLES	SYSCATALOG	SYSTABLES	SYSTABLES
SYSCOLUMNS	SYSCOLUMNS	SYSCOLUMNS	SYSCOLUMNS	SYSCOLUMNS
SYSINDEXES	SYSINDEXES	SYSINDEXES	SYSINDEXES	SYSINDEXES
SYSTABAUTH	SYSTABAUTH	SYSTABAUTH		SYSCST
SYSRELS	SYSRELS	SYSKEYCOLS		SYSKEYCST
SYSDATATYPES	SYSSYNONYMS	SYSSYNONYMS		SYSCSTCOL
SYS PROCEDURES	SYSKEYS	SYSKEYS		SYSKEYS
SYS PROC PARMS	SYSCOLAUTH	SYSCOLAUTH		SYSREFCST
	SYSFOREIGNKEYS			
	SYS PROCEDURES 1			
	SYSDATABASE			

1 DB2 for MVS/ESA 4.1 only.

For example, if the set of views for the system catalog tables is in the ACME schema, then the view for SYSIBM.SYSTABLES is ACME.SYSTABLES; and SYSSCHEMA should then be set to ACME.

Defining and using limited views of the system catalog tables reduces the number of tables listed by the application, which reduces the time it takes for the application to query table information.

If no value is specified, the default is:

SYSCAT or SYSIBM on DB2 Universal Database

SYSIBM on DB2 for common server versions prior to 2.1, DB2 for MVS/ESA and OS/400

SYSTEM on DB2 for VSE & VM

QSYS2 on DB2 for AS/400

This keyword can be used in conjunction with SCHEMALIST and TABLETYPE (and DBNAME on DB2 for MVS/ESA) to further limit the number of tables for which information will be returned.

TABLETYPE

Keyword Description: Define a default list of TABLETYPES returned when querying table information.

db2cli.ini Keyword Syntax: TABLETYPE = " 'TABLE' | ',ALIAS' | ',VIEW' | ',
'INOPERATIVE VIEW' | ',SYSTEM TABLE' | ',SYNONYM' "

Default Setting: No default list of TABLETYPES is defined.

DB2 CLI/ODBC Settings Tab: Enterprise

Usage Notes:

If there is a large number of tables defined in the database, a `tabletype` string can be specified to reduce the time it takes for the application to query table information, and reduce the number of tables listed by the application.

Any number of the values can be specified. Each type must be delimited with single quotes, separated by commas, and in uppercase. The entire string must also be enclosed in double quotes. For example:

```
TABLETYPE=" 'TABLE' , 'VIEW' "
```

This option can be used in conjunction with `DBNAME` and `SCHEMALIST` to further limit the number of tables for which information will be returned.

`TABLETYPE` is used to provide a default for the DB2 CLI function that retrieves the list of tables, views, aliases, and synonyms in the database. If the application does not specify a table type on the function call, and this keyword is not used, information about all table types is returned. If the application does supply a value for the *tabletype* on the function call, then that argument value will override this keyword value.

If `TABLETYPE` includes any value other than `TABLE`, then the `DBNAME` keyword setting cannot be used to restrict information to a particular DB2 for MVS/ESA database.

TEMPDIR

Keyword Description: Define the directory used for temporary files associated with LOB fields.

db2cli.ini Keyword Syntax: `TEMPDIR = < full path name >`

Default Setting: Use the system temporary directory.

DB2 CLI/ODBC Settings Tab: Environment

Usage Notes:

When working with Large Objects (CLOBS, BLOBS, etc...), a temporary file is often created on the client machine to store the information. Using this option you can specify a location for these temporary files. The system temporary directory will be used if nothing is specified.

The keyword is placed in the data source specific section of the `db2cli.ini` file, and has the following syntax:

```
TempDir= F:\DB2TEMP
```

When a Large Object is accessed, an `SQLSTATE` of `HY507` will be returned if the path name is invalid, or if the temporary files cannot be created in the directory specified.

TRACE

Keyword Description: Turn on the DB2 CLI/ODBC trace facility.

db2cli.ini Keyword Syntax: TRACE = 0 | 1

Default Setting: No trace information is captured.

DB2 CLI/ODBC Settings Tab: Service

See Also: “TRACEFILENAME,” “TRACEFLUSH” on page 198 , “TRACEPATHNAME” on page 198

Usage Notes:

When this option is on (1), CLI/ODBC trace records are appended to the file indicated by the TRACEFILENAME configuration parameter or to files in the subdirectory indicated by the TRACEPATHNAME configuration parameter.

For example, to set up a CLI/ODBC trace file that is written to disk after each trace entry:

```
[COMMON]
TRACE=1
TRACEFILENAME=E:\TRACES\CLI\MONDAY.CLI
TRACEFLUSH=1
```

(This option is contained in the Common section of the initialization file and therefore applies to all connections to DB2.)

TRACEFILENAME

Keyword Description: File used to store the DB2 CLI/ODBC trace information.

db2cli.ini Keyword Syntax: TRACEFILENAME = < Full file name >

Default Setting: None

DB2 CLI/ODBC Settings Tab: Service

Only Applicable when: the TRACE option is turned on.

See Also: “TRACE,” “TRACEFLUSH” on page 198 , “TRACEPATHNAME” on page 198

Usage Notes:

If the file specified does not exist, then it will be created; otherwise, the new trace information will be appended to the end of the file.

If the filename given is invalid or if the file cannot be created or written to, no trace will occur and no error message will be returned.

This option is only used when the TRACE option is turned on. This will be done automatically when you set this option in the CLI/ODBC Configuration utility.

See the TRACE option for an example of using the various trace settings. The TRACEPATHNAME option will be ignored if this option is set.

DB2 CLI trace should only be used for debugging purposes. It will slow down the execution of the CLI/ODBC driver, and the trace information can grow quite large if it is left on for extended periods of time.

(This option is contained in the Common section of the initialization file and therefore applies to all connections to DB2.)

TRACEFLUSH

Keyword Description: Force a write to disk after each CLI/ODBC trace entry.

db2cli.ini Keyword Syntax: TRACEFLUSH = 0 | 1

Default Setting: Do not write after every entry.

DB2 CLI/ODBC Settings Tab: Service

Only Applicable when: the CLI/ODBC TRACE option option is turned on.

See Also: "TRACE" on page 197, "TRACEFILENAME" on page 197, "TRACEPATHNAME"

Usage Notes:

Set this option on (TRACEFLUSH = 1) to force a write to disk after each trace entry. This will slow down the trace process, but will ensure that each entry is written to disk before the application continues to the next statement.

This option is only used when the TRACE CLI/ODBC option is turned on. See the TRACE option for an example.

(This option is contained in the Common section of the initialization file and therefore applies to all connections to DB2.)

TRACEPATHNAME

Keyword Description: Subdirectory used to store individual DB2 CLI/ODBC trace files.

db2cli.ini Keyword Syntax: TRACEPATHNAME = < **Full subdirectory name** >

Default Setting: None

DB2 CLI/ODBC Settings Tab: Service

Only Applicable when: the TRACE option is turned on.

Not Applicable when: the TRACEFILENAME option is turned on.

See Also: “TRACE” on page 197, “TRACEFILENAME” on page 197,
“TRACEFLUSH” on page 198

Usage Notes:

Each thread or process that uses the same DLL or shared library will have a separate DB2 CLI/ODBC trace file created in the specified directory.

No trace will occur, and no error message will be returned, if the subdirectory given is invalid or if it cannot be written to.

This option is only used when the TRACE option is turned on. This will be done automatically when you set this option in the CLI/ODBC Configuration utility.

See the TRACE option for an example of using the various trace settings. It will be ignored if the DB2 CLI/ODBC option TRACEFILENAME is used.

DB2 CLI trace should only be used for debugging purposes. It will slow down the execution of the CLI/ODBC driver, and the trace information can grow quite large if it is left on for extended periods of time.

(This option is contained in the Common section of the initialization file and therefore applies to all connections to DB2.)

TRANSLATEDLL

Keyword Description: Indicate the location of DB2TRANS.DLL (which contains codepage mapping tables).

db2cli.ini Keyword Syntax: TRANSLATEDLL = X:\PATH\DB2TRANS.DLL

Default Setting: No character set translation takes place.

DB2 CLI/ODBC Settings Tab: Compatibility

Only Applicable when: a Windows application connects to a DB2 Version 1 server.

See Also: “TRANSLATEOPTION” on page 200

Usage Notes:

Indicate the directory where the DB2 Client Application Enabler for Windows or the Software Developers Kit for Windows has been installed. DB2TRANS.DLL is the DLL that contains codepage mapping tables.

This keyword is used on 16-bit versions of Windows when connecting to DB2 for OS/2 Version 1, or when using a version of DB2 Connect for OS/2 prior to Version 2.3 in conjunction with the TRANSLATEOPTION, to provide proper mapping of NLS SBCS characters (such as the umlaut character in German) to the corresponding characters in the Windows codepage 1004.

Note: This option is useful when a Windows application connects to a downlevel server that does not support unequal codepage conversion (such as DB2 Version 1).

TRANSLATEOPTION

Keyword Description: Define the codepage number of the database in DB2 Version 1.

db2cli.ini Keyword Syntax: TRANSLATEOPTION = *database codepage number*

Default Setting: None. Must be set if the TRANSLATEDLL option is specified.

DB2 CLI/ODBC Settings Tab: Compatibility

Only Applicable when: TRANSLATEDLL is set, and a Windows application connects to DB2 Version 1.

See Also: “TRANSLATEDLL” on page 199

Usage Notes:

Defines the codepage number of the database in DB2 Version 1 (it can be obtained by querying the database configuration parameters). Specifying TRANSLATEDLL and TRANSLATEOPTION enables the translation of characters from codepage number *database codepage number* to the Windows 1004 codepage.

There are two supported values for *database codepage number*: 437 and 850. If you specify any other values, a warning is returned on the connect request indicating that translation is not possible.

Note: This option is useful when a Windows application connects to a downlevel server that does not support unequal codepage conversion (such as DB2 Version 1).

TXNISOLATION

Keyword Description: Set the default isolation level.

db2cli.ini Keyword Syntax: TXNISOLATION = 1 | 2 | 4 | 8 | 32

Default Setting: Read Committed (Cursor Stability)

DB2 CLI/ODBC Settings Tab: Transaction

Usage Notes:

Sets the isolation level to:

- 1 = Read Uncommitted (Uncommitted read)
- 2 = Read Committed (Cursor stability) (default)
- 4 = Repeatable Read (Read Stability)
- 8 = Serializable (Repeatable read)
- 32 = (No Commit, DATABASE 2 for AS/400 only; this is similar to autocommit)

The words in parentheses are IBM's terminology for the equivalent SQL92 isolation levels. Note that *no commit* is not an SQL92 isolation level and is supported only on DB2 for AS/400. Refer to the *SQL Reference* for more information on isolation levels.

UID

Keyword Description: Define default user ID.

db2cli.ini Keyword Syntax: UID = *userid*

Default Setting: None

DB2 CLI/ODBC Settings Tab: CLI/ODBC Settings General

Usage Notes:

The specified *userid* value is used if a *userid* is not provided by the application at connect time.

UNDERSCORE

Keyword Description: Specify whether or not the underscore character "_" is to be used as a wildcard character.

db2cli.ini Keyword Syntax: UNDERSCORE = **1** | **0**

Default Setting: "_" acts as a wildcard.

DB2 CLI/ODBC Settings Tab: Optimization

Usage Notes:

This option allows you to specify whether the underscore character "_" is to be used as a wildcard character (matching any one character, including no character), or to be used as itself. This option only affects catalog function calls that accept search pattern strings.

1 = "_" acts as a wildcard (default)

The underscore is treated as a wildcard matching any one character or none. For example, if two tables are defined as follows:

```
CREATE TABLE "OWNER"."KEY_WORDS" (COL1 INT)
CREATE TABLE "OWNER"."KEYWORDS" (COL1 INT)
```

The DB2 CLI catalog function call that returns table information (`SQLTables()`) will return both of these entries if "KEY_WORDS" is specified in the table name search pattern argument.

0 = "_" acts as itself

The underscore is treated as itself. If two tables are defined as the example above, `SQLTables()` will return only the "KEY_WORDS" entry if "KEY_WORDS" is specified in the table name search pattern argument.

Setting this keyword to 0 can result in performance improvement in those cases where object names (owner, table, column) in the database contain underscores.

Note: This keyword only has an effect on DB2 common server versions prior to Version 2.1. The ESCAPE clause for the LIKE predicate can be used for subsequent versions and all other DB2 servers. For more information on the ESCAPE clause, refer to the *SQL Reference*.

WARNINGLIST

Keyword Description: Specify which errors to downgrade to warnings.

db2cli.ini Keyword Syntax: WARNINGLIST = " 'xxxx', 'yyyy', ..."

Default Setting: Do not downgrade any SQLSTATEs.

DB2 CLI/ODBC Settings Tab: Service

See Also: "IGNOREWARNINGS" on page 186

Usage Notes:

Any number of SQLSTATEs returned as errors can be downgraded to warnings. Each must be delimited with single quotes, separated by commas, and in uppercase. The entire string must also be enclosed in double quotes. For example:

```
WARNINGLIST=" ' 1S 2', 'HY 9 ' "
```

This option can be used in conjunction with the IGNOREWARNINGS CLI/ODBC configuration keyword. If you also set IGNOREWARNINGS on then any errors you downgrade to warnings will not be reported at all.

Part 4. Advanced Installation and Configuration

Chapter 16. Unattended DB2 Installation on OS/2 or Windows 3.x Operating Systems

During a regular installation, you interact with the install program, filling in a series of panels to set up and configure the DB2 product.

In an unattended installation, these tasks run automatically without any intervention. The setup and configuration data is supplied by response files you set in place beforehand.

You can use this feature to install DB2 products on OS/2 or Windows 3.x operating systems:

Locally, from a CD-ROM (refer to "Installing DB2 Products from a Hard Disk or CD-ROM").

Remotely, across a network connection from a hard disk/CD-ROM (refer to "Installing DB2 Products from a Hard Disk or CD-ROM").

Remotely, using the Netfinity server, if it is installed on your system (refer to "Installing DB2 Products Remotely Using TME Netfinity Server" on page 212).



If you need to install the same DB2 product on several machines, you can use unattended installation's "response file" feature to create a customized configuration, save the customized response file, then reuse it as a template for each installation.

Installing DB2 Products from a Hard Disk or CD-ROM

To install DB2 products on local or remote workstations, from a hard disk or CD-ROM, perform the steps in the following sections:

"Step 1. Make DB2 Files Available for Installation" on page 206

"Step 2. Create a Response File for Unattended Installation" on page 207

"Step 3. Run the CMD/BAT File from the Remote Workstation" on page 209



You can also use the following procedure to *remove* all DB2 products from a target workstation. Simply follow the instructions below, but when editing the command file, specify D (Delete) instead of I (Install).

Step 1. Make DB2 Files Available for Installation

In this section, you make the DB2 install files accessible to the client machine where DB2 will be installed.

If performing *remote* installs directly from the CD-ROM:

- 1 Insert the appropriate Client Pack CD-ROM into the drive.
- 2 Ensure that the CD-ROM will remain in the drive for as long as it is needed for the install. If the drive is often used for other tasks as well, consider installing from the hard disk instead.

If installing from a hard disk, you must copy the required files from the CD-ROM directories to the hard disk. At the command prompt, use the **xcopy** command with the **/s** option.

- For OS/2, the syntax for the two commands required is:

```
xcopy x:\db2\os2\language e:\clients\os2\language /s
xcopy x:\db2\os2\common e:\clients\os2\common /s
```

where:

- *x*: represents the CD-ROM drive
- *language* is the two-character code that represents your language (for example, **EN** for English). Table 23 on page 278 lists the code for each available language.
- *e*: represents the destination drive

- For Windows 3.x, the command syntax is:

```
xcopy x:\db2\windows\language\client\ . e:\clients\windows /s
```

where:

- *x*: represents the CD-ROM drive
- *language* is the two-character code that represents your language (for example, **EN** for English). Table 23 on page 278 lists the code for each available language.
- *e*: represents the destination drive

If performing a remote installation, you must enable your network clients to access the DB2 install files.

Grant *shared access* authority to the directory that you just created, or to the CD-ROM drive if installing directly from CD-ROM.

Step 2. Create a Response File for Unattended Installation

During an interactive installation, you provide the information needed to install DB2 and configure its environment. During an unattended installation, this information is provided as keywords and values in a response file. DB2's install package includes a ready-to-use sample response file, with default entries already in place.



If you intend to use the sample response file provided, without making any changes to its values, you can skip this step and go to "Step 3. Run the CMD/BAT File from the Remote Workstation" on page 209.

To edit the sample response file:

- 1 Find the correct sample response files for the platform you are installing to, and for the product you want to install. Copy the files to a local directory.

If installing directly from the CD-ROM, substitute `x:\os2` or `x:\windows` (depending on the target workstation's operating system), where `x`: represents the CD-ROM drive for `e:\clients` in the directory names below.

For OS/2, edit

- `e:\clients\os2\language\db2cae.rsp` for DB2 Client Application Enabler
- `e:\clients\os2\language\db2sdk.rsp` for DB2 Software Developer's Kit.

where *language* is the two-character country code that represents your language (for example, `EN` for English). Table 23 on page 278 lists the codes for each available language.

For Windows 3.x, use one of:

- `e:\clients\windows\db2caew.rsp` for DB2 Client Application Enabler
- `e:\clients\windows\db2sdkw.rsp` for DB2 Software Developer's Kit.

- 2 The response file contains:

Keywords unique to installation

Registry value/environment variable settings

Database manager configuration parameter settings.



For more information on:

DB2 environment variables and registry values, see "DB2 Registry Values and Environment Variables" on page 163, or refer to the *Administration Guide*.

Database manager configuration parameters, refer to the *Administration Guide*.

- 3** To activate an item in the response file, remove the asterisk (*) to the left of the keyword/environment variable, erase the current setting to the right of the value and type in a new setting. The possible settings are listed to the right of the current setting.

An example section of a response file appears below:

```
...
FILE                = c:\sqllib
COMP                = Client Application Enabler
...
DB2ACCOUNT         = BLANK or char(199)
DB2BQTIME          = BLANK or 1 - MAX
...
```

Keywords unique to installation are parameters that are only specified in a response file during an unattended installation. The following is a list of keywords unique to OS/2 and Windows 3.x unattended installation:

FILE

Specifies the destination directory for DB2. If you do not change this value, it defaults to `x:\sqllib` where `x`: is the drive on which your operating system is installed.



Install DB2 products only on a drive which is local to the target workstation. Installing on a non-local drive can cause performance and availability problems.

COMP

Specifies the components you want installed. By default, all components of the DB2 product are installed. If you do not want to install a component, add an asterisk (*) to the left of the *COMP* line for that component in the response file.

OVERWRITE

Specifies whether to automatically overwrite files during installation. Valid values for this keyword are `YES` and `NO`.

CFGUPDATE

Specifies whether the `config.sys` file is automatically updated. Valid values for this keyword are:

– `AUTO`

Automatically updates `CONFIG.SYS`.

– `MANUAL`

Does not update `CONFIG.SYS`.

SAVEBACKUP

Specifies whether to save a backup version of the product when it is updated. Valid values for this keyword are YES and NO.

DB2SYSTEM

On OS/2, specifies a name for the system which is unique within a network.

Note: This parameter must be specified.

AUTOSTART_CONTROL_CENTER

On OS/2, specifies whether or not to automatically start the Control Center each time the system is rebooted.

Note: By default the Control Center will be automatically started unless this parameter is set to NO.

DB2.AUTOSTART

On OS/2, specifies whether or not to automatically start the DB2 instance each time the system is rebooted.

Note: By default the DB2 instance will be automatically started unless this parameter is set to NO.

- 4 Exit the file. If you have made any changes, save the file under a new file name to preserve the original sample response file.

If you are installing directly from the CD-ROM, you must store the renamed response file on a local drive.

Step 3. Run the CMD/BAT File from the Remote Workstation

A command (CMD) or batch (BAT) file contains the commands which will start the installation program. You must edit this file before running the installation.

- 1 Find and open the correct CMD or BAT file.

If installing directly from the CD-ROM, substitute `x:\os2` or `x:\windows` (depending on the target workstation's operating system) for `e:\clients\os2\` in the directory names below.

For OS/2, the CMD files are available in the `e:\clients\os2\language` directory where *language* is the two-character country code that represents your language (for example, `EN` for English). Table 23 on page 278 lists the codes for each available language. For example:

- `db2cae.cmd` for DB2 Client Application Enabler
- `db2sdk.cmd` for DB2 Software Developer's Kit.

For Windows 3.x, the BAT files are:

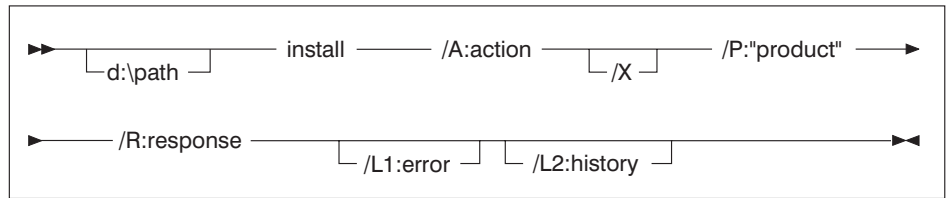
- `e:\clients\windows\db2caew.bat` for DB2 Client Application Enabler

- e:\clients\windows\db2sdkw.bat for DB2 Software Developer's Kit.

2 Edit the command in the file, to fill in the information needed for the installation.

For OS/2:

- a** You must specify the complete install command. A complete command requires the following syntax:



The following is an example of a complete command in the sample command files:

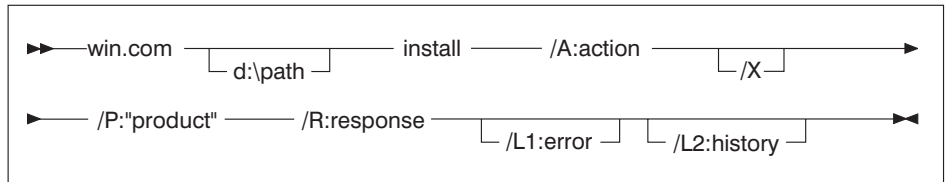
```
e:\clients\os2\language\install\install /A:I /X
/P:"IBM DB2 Client Application Enabler"
/R:e:\clients\os2\language\db2cae.rsp /L1:d:\error.log
/L2:d:\history.log
```

where:

- /d:\path** Specifies the location of the install files. If installing from the hard drive, specify the directory created in Step 1.
- /A** Specifies the action to be performed.
 - I** Specifies install.
 - Note:** If you want to *remove* all DB2 products from a target workstation, specify D (Delete) instead of I (Install).
- /X** Specifies that the installation will run in unattended mode.
- /P** Specifies the name of the product you want to install.
- /R** Specifies the fully qualified response file name. If you changed and renamed the sample response file provided, make sure this parameter matches the new name.
- /L1** Specifies the fully qualified log file name, where setup information and any errors occurring during setup are logged.
- /L2** Specifies the fully qualified history log name, where all the files processed by the installation program are listed.

On Windows 3.1:

A complete command requires the following syntax:



The following is an example of a complete command in the sample command files:

```
win e:\clients\windows\install\install /A:I /X
/P:"IBM Client Application Enabler V5"
/R:e:\clients\windows\db2caew.rsp /L1:d:\error.log
/L2:d:\history.log
```

where:

- /d:\path** Specifies the location of the install files. If installing from the hard drive, specify the directory created in Step 1.
- /R** Specifies the fully qualified response file name. If you changed and renamed the sample response file provided, make sure this parameter matches the new name.
- /A** Specifies the action to be performed.
 - I** Specifies install.

Note: If you want to *remove* all DB2 products from a target workstation, specify D (Delete) instead of I (Install).
- /X** Specifies that the installation will run in unattended mode.
- /P** Specifies the name of the product you want to install.
- /L1** Specifies the fully qualified log file name, where setup information and any errors occurring during setup are logged.
- /L2** Specifies the fully qualified history log name, where all the files processed by the installation program are listed.

3 Save and exit the CMD or BAT file.

If you are installing directly from the CD-ROM, you must rename the CMD or BAT file, store it on a local drive, and use that file name in the next step.

4 Start the installation by entering the CMD or BAT file's name on the command line. The DB2 Client Application Enabler is now ready to install on the target workstation.

5 When installation is complete, check the error and history logs for any errors or problems.



To configure your client to access remote servers, go to Chapter 9, "Configuring Client-to-Server Communications Using the Client Configuration Assistant" on page 49.

Installing DB2 Products Remotely Using TME Netfinity Server

To install a DB2 product in a Netfinity Server environment, perform the steps in the following sections:

“Step 1. Copy the DB2 Install Files to the Code Server”

“Step 2. Add the DB2 Product to the Software Library” on page 213

“Step 3. Configure the DB2 Product” on page 214

“Step 4. Install DB2 Products on the Remote Workstation” on page 214

Note: The following are the required levels of Netfinity:

Netfinity Server Product: IBM TME 10 Netfinity Server - Manager Version 4
Product Number: 5697-288, Feature 0000
Must have component: Software Distribution Server at service level 00000003
Netfinity Server cannot be used for Windows 3.x workstations.

Netfinity Client Product: IBM TME 10 Netfinity Server - Client Version 4
Product Number: 5697-288, Feature 0000
Must have component: Software Distribution Client at service level 00000003

Step 1. Copy the DB2 Install Files to the Code Server

- 1 Insert the appropriate CD-ROM into the drive.
- 2 Copy the required files from the CD-ROM directories to the code server's hard drive. At the command prompt, use the **xcopy** command with the **/s** option.

The syntax for the two commands required is:

```
xcopy x:\db2\os2\language e:\cid\img\language /s
xcopy x:\db2\os2\common e:\cid\img\common /s
```

where

x: represents the CD-ROM drive
language is the two-character code that represents your language (for example, **EN** for English. Table 23 on page 278 lists the code for each available language.

e is the letter that designates the code server's hard disk

- 3 Leave the CD-ROM in the drive; it is required for the next step.

Step 2. Add the DB2 Product to the Software Library



Once you have added a particular DB2 product to the software library, you can omit this step for later installations of that product.

To add a DB2 product to the software library:

- 1 Start Netfinity Server.
- 2 Double-click on the **Netfinity Server Service Manager** icon.
- 3 Double-click on the **CID Software Preparation** icon.
- 4 Double-click on the **Software Library** icon.
- 5 Select **Software->New** from the menu bar on the Software Library window.
- 6 Enter the name of the product you are adding.
- 7 Select the **Find** push button.
- 8 In the Find definition file window, select:
 - The CD-ROM drive letter in the **Drive** list
 - en\install\ .adf in the **Directory** list
 - The appropriate .adf file name in the **File** list:
 - db2cae.adf for DB2 Client Application Enabler
 - db2sdk.adf for DB2 Software Developer's Kit.
- 9 Click on **OK**.
- 10 Select the **Add** push button to associate the application definition file with DB2.

This copies the remote installation files (.adf, .var, .cmd and .mrf) from the CD-ROM to the subdirectory where Netfinity Server resides.

An icon is created in the **Software Library** folder, representing the DB2 product you have just added.

Step 3. Configure the DB2 Product

In this step, you configure the DB2 product for the target workstation, just as you would during a standard interactive installation. To perform this task:

- 1 Double-click on the **DB2** product icon in the Software Library window.
- 2 Select **Configuration->New**.
The Settings notebook opens.
- 3 Enter a unique identifier to define this configuration.
- 4 On each page of the notebook, select each of the items in the **Variables** list and choose the setting. A description of each variable is displayed in the **Explanation** field of the notebook page.
- 5 When you have completed all of the pages of the notebook, click on **OK**.
- 6 When the message *Do you want to catalog the database?* appears, click on the **Yes** push button.

The values entered in the notebook are used to create a response file which will supply the information needed to run the installation.

Step 4. Install DB2 Products on the Remote Workstation

To install DB2 products on the remote workstation, you can use either the Event Scheduler or the Remote System Manager. For information on how to use these tools, refer to the Netfinity Server documentation.

After installation, you must reboot the workstation before using DB2.



To configure your client to access remote servers, go to Chapter 19, "Using the Client Configuration Assistant" on page 251.

Chapter 17. Unattended DB2 Installation on Windows 32-bit Operating Systems

During a regular installation, you interact with the install program, filling in a series of panels to set up and configure DB2.

In an unattended installation, you supply the setup and configuration data in a response file you create before running the installation.

You can use this feature to install DB2 products on Windows 32-bit operating systems operating systems:

Locally, from a CD-ROM (see “Installing DB2 Products from a Hard Disk or CD-ROM”)

Remotely, across a network connection from a hard disk or CD-ROM (see “Installing DB2 Products from a Hard Disk or CD-ROM”)

Remotely, across a network connection to target workstations, using Microsoft Systems Management Server (SMS) if it is installed on your Windows NT system (see “Installing DB2 Remotely Using SMS” on page 223).



If you need to install the same DB2 product on several machines, you can use unattended installation's "response file" feature to create a customized configuration, save the customized response file, then reuse it as a template for each installation.

Installing DB2 Products from a Hard Disk or CD-ROM

To install DB2 products for Windows 32-bit operating systems, from a hard disk or CD-ROM to local or remote workstations, perform the steps in the following sections:

“Step 1. Make DB2 Files Available for Installation” on page 216

“Step 2. Create a Response File for Unattended Installation” on page 218

“Step 3. Run Setup from the Workstation” on page 222

You can also use this procedure to perform an *interactive* remote install instead of an unattended remote one. Follow the instructions below.

Before You Begin

Read and perform the steps in this section before you begin the installation to ensure that you have the required items and information you will need.

Creating Usernames for DB2 Installation and Operation

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

Creating a Username for Installing DB2 Products:

Windows 95 and Windows 98

The username 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 C, "Naming Rules" on page 279.

Windows NT

You need to have a username that will be used to install DB2. The username may belong to the Domain or local Administrators group.



On clients, this username will be removed from the system when the installation is complete.

Step 1. Make DB2 Files Available for Installation

To make the DB2 install files accessible to the machine where DB2 will be installed:

If performing *remote* installs directly from the CD-ROM:

- 1 Insert the appropriate Client Pack CD-ROM into the drive.
- 2 Ensure that the CD-ROM remains in the drive for as long as the installation requires it. If the drive is often used for other tasks as well, consider installing from the hard disk instead.

If installing from a code server, you must copy the required files from the CD-ROM directories to the hard disk:

- 1 Insert the appropriate CD-ROM into the drive.
- 2 Create a directory on the code server, for example, `q:\db2prods`.

- 3 At the command prompt, use the **cpyssetup.bat** command to copy the files necessary for a particular language install to a code server.

The command syntax is:

- `x:\db2\winnt95\common\cpyssetup.bat q:\db2prods language`
for:
 - DB2 Client Application Enabler
 - DB2 Software Developer's Kit

where:

- *x*: represents your CD-ROM drive
- *q*: represents the code server's disk.
- *language* is the two-character country code that represents your language (for example, `EN` for English). See Table 23 on page 278

If performing a remote installation, you must enable your network clients to access the DB2 install files.

Grant *shared access* to the directory that you just created, or to the CD-ROM drive.

To set up a share:

- 1 Click on **Start** and select **Programs->Windows Explorer**.
- 2 Select the directory that you want to share, for example `q:\db2prods`.
- 3 Select **Files->Properties** from the menu bar. The properties window for the directory will open. Click on the **Sharing** tab.
- 4 Select the **Shared As** radio button.
- 5 In the **Share Name** field type a share name. For example, type `db2nt`.
- 6 To specify read access for everyone on Windows NT:
 - a Select the **Permissions** push button. The Access Through Share Permissions window opens.
 - b In the **Name** box, make sure that **Everyone** is selected.
 - c In the **Type of Access** box, select **Read**.
 - d Click on **OK**.

On Windows 95 and Windows 98, you do not need to specify type of access when you set up a share. By default, everyone is allowed read access.

In our example scenario, `q:\db2prods` uses a share name of `db2nt`. We will use this value in the examples that follow.

Step 2. Create a Response File for Unattended Installation

During an interactive installation, you interact with the install program, filling in a series of panels to set up and configure the DB2 product. During an unattended installation, you supply the setup and configuration data in a response file you create before running the installation. DB2's install package includes a number of ready-to-use sample response files, with default entries already in place.

The sample response files are located in:

```
x:\db2\winnt95\common
```

for:

- DB2 Client Application Enabler
- DB2 Software Developer's Kit

where *x* is the letter that designates the CD-ROM drive. There is a different response file for each product on the CD, so that the user has a valid list of parameters for each product.



If you intend to use the sample response file provided, without making any changes to its values, you can skip this step and go to "Step 3. Run Setup from the Workstation" on page 222.

Edit the sample response file. For example, you can use Notepad or WordPad.

- 1 Open the sample response file for the package you want to install.

`db2cae.rsp` for DB2 Client Application Enabler

`db2sdk.rsp` for DB2 Software Developer's Kit

- 2 The response file contains:

Keywords unique to installation

Registry value/environment variable settings

Database manager configuration parameter settings.



For more information on:

DB2 environment variables and registry values, see "DB2 Registry Values and Environment Variables" on page 163, or refer to the *Administration Guide*.

Database manager configuration parameters, refer to the *Administration Guide*.

- 3 To activate an item in the response file, remove the asterisk (*) to the left of the keyword/environment variable, erase the current setting to the right of the value

and type in a new setting. The possible settings are listed to the right of the current setting.

An example section of a response file appears below:

```
...
FILE                = c:\sql11b
REBOOT              = No
  COMP              = Control Center
...
```

Keywords unique to installation are parameters that are only specified in a response file during an unattended installation. The following is a list of keywords unique to installation for Windows 32-bit operating systems unattended installation:

FILE

Specifies the destination directory for DB2.

Note: Check the value for the *FILE* keyword. *FILE* specifies where to install the product. If you do not change this value, it defaults to `x:\sql11b` where `x:` is the drive on which your operating system is installed.

REBOOT

Specifies whether to restart the system following installation.

TYPE

Specifies the type of install. The options are:

- =Compact
- 1=Typical (default)
- 2=Custom

For `2` and `1` (compact and typical), any Custom Keywords (COMP) value will be ignored.

For all types, default communication values will be used where applicable, as follows:

- a** `DB2.DB2COMM` will be set for the protocols detected on your system.
- b** If a protocol is detected on the system, or if you have specified values for `DB2.DB2COMM`, values for its parameters will be generated so that the protocol will be enabled for DB2's use at the end of the install. Checks are made to ensure that the generated values are not duplicates, and that they conform to the protocols set up on your system. Default values will be generated for the following:

- For NetBIOS:
 - `DB2.DB2NBADAPTERS`
 - `DB2.NNAME`

- For TCP/IP:

DB2 . PORT_NUMBER
DB2 . SVCENAME

– For IPX/SPX:

DB2 . FILESERVER
DB2 . OBJECTNAME
DB2 . IPX_SOCKET

– For APPC:

DB2 . TPNAME

Note: You may override these values; however, if the associated protocol has not been detected or specified, the values will not be used.

PROD

Specifies the product you want installed. The options are:

- DB2_CAE for DB2 Client Application Enabler
- DB2_SDK for DB2 Software Developer's Kit

Note: You can use this parameter more than once.

COMP

Specifies the components you want installed. The installation program automatically installs components that are required for a product, and ignores requested components that are not available.

Note: Component selections have no effect unless you specify a `custom` installation.

The following table lists *COMP* keywords and the components they represent.

Keyword	Component
Product Components	
GRAPHICAL_TOOLS	Graphical Tools
CLIENT_CONFIGURATION_ASSISTANT	Client Configuration Assistant
CONTROL_CENTER	Control Center
WEB_CONTROL_CENTER	Web Control Center
PERFORMANCE_MONITOR	Performance Monitor
VISUAL_EXPLAIN	Visual Explain
ODBC_DRIVER	DB2 ODBC Driver
SAMPLE_APPLICATIONS ¹	Sample Applications
THIN_CLIENT_CODE_SERVER	Thin Client Code Server

Table 22 (Page 2 of 2). Component Keywords for Windows 32-bit Operating Systems Unattended Install

Keyword	Component
DB2 Books	
API_REFERENCE 1	API Reference
ADMINISTRATION_GUIDE	Administration Guide
ADMINISTRATION_GETTING_STARTED	Administration Getting Started
DB2_PROGRAMMING_GETTING_STARTED 1	Road Map to DB2 Programming
BUILDING_OS2_WIN_APPLICATIONS 1	Building Applications for Windows and OS/2 Environments
CLI_GUIDE_REFERENCE 1	CLI Guide and Reference
COMMAND_REFERENCE	Command Reference
SQL_PROGRAMMING_GUIDE 1	Embedded SQL Programming Guide
GLOSSARY	Glossary
MASTER_INDEX	Master Index
MESSAGES_REFERENCE	Message Reference
SQL_GETTING_STARTED	SQL Getting Started
SQL_REFERENCE	SQL Reference
SYSTEM_MONITOR_GUIDE	System Monitor Guide and Reference
TROUBLESHOOTING_GUIDE	Troubleshooting Guide
Note:	
1. Software Developer's Kit only	

AUTO_CTRL_CENTER

Specifies whether or not to automatically start the Control Center each time the system is rebooted.

Note: By default the Control Center starts automatically unless this parameter is set to **NO**.

AUTOSTART_CCA

Specifies whether or not to automatically start the Client Configuration Assistant each time the system is rebooted.

Note: By default, the Client Configuration Assistant starts automatically unless this parameter is set to **NO**.



Install DB2 products only on a drive which is local to the target workstation. Installing on a non-local drive can cause performance and availability problems.

By default, the name of the response file will be used to determine the products to be installed. You can override this setting by providing one or more **PROD** parameter values in the response file.

- 4 Exit the file. If you have made any changes, save the file under a new file name to preserve the original sample response file.

If you are installing directly from the CD-ROM, you must store the renamed response file on another drive.

Step 3. Run Setup from the Workstation

To perform the installation:

- 1 At the workstation where DB2 will be installed, logon as a username that meets the requirements for installing DB2. See “Creating Usernames for DB2 Installation and Operation” on page 216 for more information.
- 2 If performing a remote install, connect to the shared directory of the hard disk, or to the CD-ROM drive, using a command similar to the following:

```
net use e: \\machine\directory
```

where:

e is the drive letter used to connect to the shared directory on the hard disk or CD-ROM drive.

machine is the machine name of the code server.

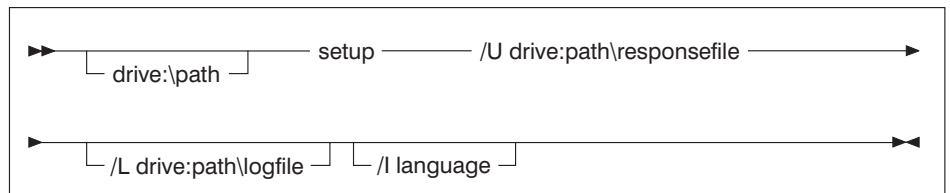
directory is the share name of the code server or CD-ROM drive.

For example, type:

```
net use e: \\codeserv\db2nt
```

Based on the example set up in previous steps, this points to `q:\db2prods` on the code server.

- 3 Run the setup program:
 - a Click on **Start** and select **Run**.
 - b In the Open Window, type the path to the setup program. The path accepts the following command line arguments:



Specify the fully qualified path to start the program. For example:

```
e:\setup /U d:\db2\common\db2setup.rsp /L db2log.log /I language
```

where:

- /U** Specifies the fully qualified response file name. If you changed and renamed the sample response file provided, make sure that this parameter matches the new name.
 - /L** Specifies the fully qualified log file name, where setup information and any errors occurring during setup are logged.
If you do not specify the log file's name, DB2 names it `db2.log`, and stores it in a directory called `db2log`, on the drive on which your operating system is installed.
 - /I** Specifies the two-character country code that represents your language (for example, `EN` for English). Table 23 on page 278 lists the code for each available language.
If you do not specify the language, setup will determine the system language, and launch the appropriate DB2 install for that language.
- c** Click on **OK** to start the setup program.
 - d** The installation proceeds without further action on your part.
- 4** Check the messages in the log file when the installation finishes.



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.

- 5** Following installation, you must reboot the workstation before using DB2.

Note: If you specified `REBOOT=yes` in your response file, the workstation will reboot automatically.



To configure your DB2 software to access remote DB2 servers, go to Chapter 9, "Configuring Client-to-Server Communications Using the Client Configuration Assistant" on page 49.

Installing DB2 Remotely Using SMS

With Microsoft Systems Management Server (SMS), you can install DB2 across a network onto remote Windows NT systems, setting up the installation from a central location.



You can also use SMS to *remove* all DB2 products from a target workstation. Simply follow the instructions below, but when creating the SMS package, select **Uninstall DB2 Products** instead of selecting the packages to be installed.

To install DB2 using SMS, perform the steps in the following sections:

- “Step 1. Make DB2 Files Available for Installation” on page 225
- “Step 2. Create a Response File for Unattended Installation” on page 226
- “Step 3. Import the DB2 Install Files into SMS” on page 230
- “Step 4. Distribute the DB2 Installation Package Using SMS” on page 230
- “Step 5. Run the Installation on the SMS Client” on page 231

Before You Begin

Read and perform the steps in this section before you begin the installation to ensure that you have the required items and information you will need.

Creating Usernames for DB2 Installation and Operation

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

Creating a Username for Installing DB2

Windows 95 and Windows 98

The username 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 C, “Naming Rules” on page 279.

Windows NT

You need to have a username that will be used to install DB2. The username may belong to the Domain or local Administrators group.



On clients, this username will be removed from the system when the installation is complete.

SMS Requirements

You must already have SMS installed and configured on your network. Refer to *Microsoft's Systems Management Server for Windows NT Administrator's Guide* for information on how to:

- Set up SMS (including setting up primary and secondary sites).
- Add clients to the SMS system.
- Set up inventory collection for clients.



To create an SMS package that copies the required files directly from the CD-ROM, insert the appropriate CD-ROM into the drive and go to "Step 2. Create a Response File for Unattended Installation" on page 226.

Step 1. Make DB2 Files Available for Installation

To make the DB2 install files accessible to the machine where DB2 will be installed:

If performing *remote* installs directly from the CD-ROM:

- 1 Insert the appropriate CD-ROM into the drive.
- 2 Ensure that the CD-ROM remains in the drive for as long as the installation requires it. If the drive is often used for other tasks as well, consider installing from the hard disk instead.

If installing from a code server, you must copy the required files from the CD-ROM directories to the hard disk:

- 1 Insert the appropriate CD-ROM into the drive.
- 2 Create a directory on the code server, for example, `q:\db2prods`.
- 3 At the command prompt, use the **cpyssetup.bat** command to copy the files necessary for a particular language install to a code server.

The command syntax is:

- `x:\db2\winnt95\common\cpyssetup.bat q:\db2prods language`
for:
 - DB2 Client Application Enabler
 - DB2 Software Developer's Kit

where:

- `x`: represents your CD-ROM drive
- `q`: represents the code server's disk.
- `language` is the two-character country code that represents your language (for example, `EN` for English). Table 23 on page 278

If performing a remote installation, you must enable your network clients to access the DB2 install files.

Grant *shared access* to the directory that you just created, or to the CD-ROM drive.

To set up a share:

- 1 Click on **Start** and select **Programs->Windows Explorer**.
- 2 Select the directory that you want to share, for example `q:\db2prods`.
- 3 Select **Files->Properties** from the menu bar. The properties window for the directory will open. Click on the **Sharing** tab.
- 4 Select the **Shared As** radio button.
- 5 In the **Share Name** field type a share name. For example, type `db2nt`.
- 6 To specify read access for everyone on Windows NT:
 - a Select the **Permissions** push button. The Access Through Share Permissions window opens.
 - b In the **Name** box, make sure that **Everyone** is selected.
 - c In the **Type of Access** box, select **Read**.
 - d Click on **OK**.

On Windows 95 and Windows 98, you do not need to specify type of access when you set up a share. By default, everyone is allowed read access.

In our example scenario, `q:\db2prods` uses a share name of `db2nt`. We will use this value in the examples that follow.

Step 2. Create a Response File for Unattended Installation

During an interactive installation, you interact with the install program, filling in a series of panels to set up and configure DB2. During an unattended installation, you supply the setup and configuration data in a response file you create before running the installation. DB2's install package includes a number of ready-to-use sample response files, with default entries already in place.

The sample response files are located in:

```
x:\db2\winnt95\common
```

for:

- DB2 Client Application Enabler
- DB2 Software Developer's Kit

where *x* is the letter that designates the CD-ROM drive. There is a different response file for each product on the CD, so that the user has a valid list of parameters for each product.



If you intend to use the sample response file provided, without making any changes to its values, you can skip this step and go to “Step 4. Distribute the DB2 Installation Package Using SMS” on page 230.

Edit the sample response file. For example, you can use Notepad or WordPad.

- 1 Open the sample response file for the package you want to install.

`db2cae.rsp` for DB2 Client Application Enabler

`db2sdk.rsp` for DB2 Software Developer's Kit

- 2 The response file contains:

Keywords unique to installation

Registry value/environment variable settings

Database manager configuration parameter settings.



For more information on:

DB2 environment variables and registry values, see “DB2 Registry Values and Environment Variables” on page 163, or refer to the *Administration Guide*.

Database manager configuration parameters, refer to the *Administration Guide*.

- 3 To activate an item in the response file, remove the asterisk (*) to the left of the keyword/environment variable, erase the current setting to the right of the value and type in a new setting. The possible settings are listed to the right of the current setting.

An example section of a response file appears below:

```
...  
FILE                = c:\sqllib  
REBOOT              = No  
  COMP               = Control Center  
...
```

Keywords unique to installation are parameters that are only specified in a response file during an unattended installation. The following is a list of keywords unique to installation for Windows 32-bit operating systems unattended installation:

FILE

Specifies the destination directory for DB2.

Note: Check the value for the *FILE* keyword. *FILE* specifies where to install the product. If you do not change this value, it defaults to `x:\sql11ib` where *x*: is the drive on which your operating system is installed.

REBOOT

Specifies whether to reboot the system following installation.

TYPE

Specifies the type of install. The options are:

=Compact

1=Typical (default)

2=Custom

For 2 and 1 (compact and typical), any Custom Keywords (COMP) value will be ignored.

For all types, default communication values will be used where applicable, as follows:

- a** `DB2.DB2COMM` will be set for the protocols detected on your system.
- b** If a protocol is detected on the system, or if you have specified values for `DB2.DB2COMM`, values for its parameters will be generated so that the protocol will be enabled for DB2's use at the end of the install. Checks are made to ensure that the generated values are not duplicates, and that they conform to the protocols set up on your system. Default values will be generated for the following:

- For NetBIOS:

`DB2.DB2NBADAPTERS`
`DB2.NNAME`

- For TCP/IP:

`DB2.PORT_NUMBER`
`DB2.SVCENAME`

- For IPX/SPX:

`DB2.FILESERVER`
`DB2.OBJECTNAME`
`DB2.IPX_SOCKET`

- For APPC:

`DB2.TPNAME`

Note: You may override these values; however, if the associated protocol has not been detected or specified, the values will not be used.

PROD

Specifies the product you want installed. The options are:

- *DB2_CAE* for DB2 Client Application Enabler
- *DB2_SDK* for DB2 Software Developer's Kit

Note: You can use this parameter more than once.

COMP

Specifies the components you want installed. The installation program automatically installs components that are required for a product, and ignores requested components that are not available.

Note: Component selections have no effect unless you specify a *custom* installation.

Table 22 on page 220 lists *COMP* keywords and the components they represent.

AUTO_CTRL_CENTER

Specifies whether or not to automatically start the Control Center each time the system is rebooted.

Note: By default, the Control Center starts automatically unless this parameter is set to *NO*.

AUTOSTART_CCA

Specifies whether or not to automatically start the Client Configuration Assistant each time the system is rebooted.

Note: By default, the Client Configuration Assistant starts automatically unless this parameter is set to *NO*.



Install DB2 only on a drive that is local to the target workstation. Installing on a non-local drive can cause performance and availability problems. By default, the name of the response file will be used to determine the products to be installed. You can override this setting by providing one or more **PROD** parameter values in the response file.

- 4 Exit the file. If you have made any changes, save the file under a new file name to preserve the original sample response file.

If you are installing directly from the CD-ROM, you must store the renamed response file on another drive.

Step 3. Import the DB2 Install Files into SMS

To import the DB2 install files into SMS:

- 1 Insert the appropriate CD-ROM into the drive.
- 2 From the **Start** button select **Programs**, then **Systems Management Server**, and then **SMS Administrator**. The SMS Administrator window opens.
- 3 Open the Package Properties window. The first time you start the SMS Administrator program you will see a dialog box prompting you to open an SMS Window. You can select the Packages option from there. Otherwise, select **File->New** from the menu bar.
- 4 Click on the **Import** push button. The **File Browser** opens.
- 5 In the **File Name** field, select the `db2.pdf` file.

The server `db2.pdf` file for DB2 Universal Database is located in the

```
x:\db2\common directory (for DB2 Server)
x:\db2\winnt95\common directory (for DB2 CAE or DB2 SDK)
```

where `x:` designates your CD-ROM drive

Step 4. Distribute the DB2 Installation Package Using SMS

In this step, you create an SMS package and send it to the SMS client where you want to install DB2.

To create the SMS package:

- 1 From the Package Properties window, click on the **Workstations** push button. The Setup Package For Workstations window opens, showing a list of predefined DB2 installation packages that you can use.
- 2 In the **Source Directory** field, type the name of the parent directory where you put the copied DB2 files (for example, `q:\db2prods`).
- 3 Select the name of the product to install from the **Workstation Command Lines** list.
- 4 If you changed and renamed the sample response file, click on the **Properties** push button and change the value of the parameter to match the new response file name.

If you do not specify the language, setup will determine the system language, and launch the appropriate DB2 install for that language. If you want to install a different language, use the `/l` parameter to specifies the two-character code that

represents your language (for example, `EN` for English). Table 23 on page 278 lists the code for each available language.

- 5** Click on the **Close** push button.
- 6** Click on **OK** to close the opened windows.
The Packages window shows the name of the new SMS package.
- 7** Specify where to run the package:
 - a** Open the Sites and Packages windows.
 - b** In the Packages window, select the appropriate package and drag it onto the target client in the Sites window. The Job Details window opens.
 - c** Select the **Run Workstation Command** check box and select the installation package that you want to use for the Run Command on Workstation job. For more information about completing the Job Details window, refer to *Microsoft's Systems Management Server for Windows NT Administrator's Guide*.
 - d** When the job specifications are complete, click on **OK** to close the opened windows.

The job is created and sent to the SMS client.

Step 5. Run the Installation on the SMS Client

Now that the remote installation package has been prepared and sent, the SMS client on the target machine can run the installation.

- 1** On the SMS client where DB2 will be installed, logon as a username that meets the requirements for installing DB2. See "Creating Usernames for DB2 Installation and Operation" on page 224 for more information.
- 2** Select **Start->Programs -> Systems Management Client** and then click on **Package Command Manager**. The Package Command Manager window opens.
- 3** When packages are sent from the SMS server, they are listed in the window. Select the DB2 package, then click on the **Execute** push button. The installation runs automatically.
- 4** Following installation, you must reboot the SMS client before using DB2.
Note: If you specified `REBOOT=yes` in your response file the SMS client will reboot automatically.



To configure your DB2 software to access remote DB2 servers, go to Chapter 9, "Configuring Client-to-Server Communications Using the Client Configuration Assistant" on page 49.

Chapter 18. Installing Thin Clients on Windows 32-bit Operating Systems

With this feature, you can install DB2 Client Application Enabler on a code server, and have thin client workstations access the code across a LAN connection.

Functionally, these client workstations act identically like any other DB2 clients, and can access DB2 servers using the code server. In setup, the main difference is that the DB2 Client Application Enabler code is installed on the code server, and not individually on each workstation. The thin clients need only a minimal configuration to set their parameters and establish the links to the code server. The benefits to this arrangement are:

- Reduced disk space required for each thin client (this feature can save from 22–40 MB per client workstation)

- Easier software changes and updates, since only one copy of DB2 Client Application Enabler is involved.

- In cases where the configuration information is identical for each thin client, one administrator on the central code server can set up and manage node and database access for all thin clients simultaneously.

The potential drawback for this arrangement is a minor loss of system performance, since the DB2 programs are loaded from the code server instead of the client machine.

You can install thin clients:

- Across a network connection, from the code server hard disk (see “Installing Thin Clients using the Code Server” on page 234).

- Across a network connection, using Microsoft Systems Management Server (SMS) if it is installed on your system (see “Installing Thin Clients using SMS” on page 243).

- From a diskette (see “Step 4. Make Thin Client Setup Program Accessible to the Target Workstation” on page 241).

Registry values, environment variables, and configuration parameters control your database environment.



To set up a code server, and use Microsoft Systems Management Server (SMS) to install one or more thin clients, see “Installing Thin Clients using SMS” on page 243.

Installing Thin Clients using the Code Server

To set up a code server and install one or more thin clients which will access it, perform the steps in the following sections:

“Step 1. Install DB2 Client Application Enabler on the Code Server” on page 236.

“Step 2. Create Windows 95 or Windows 98 Code Base on Windows NT Code Server” on page 237.

“Step 3. Create a Response File for Thin Client Installation” on page 238.

“Step 4. Make Thin Client Setup Program Accessible to the Target Workstation” on page 241.

“Step 5. Install Thin Client on the Target Workstation” on page 241.

If you plan to install multiple thin clients in a row, perform all the steps for the first thin client, then perform only “Step 4. Make Thin Client Setup Program Accessible to the Target Workstation” on page 241 and “Step 5. Install Thin Client on the Target Workstation” on page 241 for each subsequent thin client on that code server.

Before You Begin

Read and perform the steps in this section before you begin the installation to ensure that you have the required items and information you will need.

Creating a Username for Installing DB2 Products

Windows 95 and Windows 98

The username 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 C, “Naming Rules” on page 279.

Windows NT

You need to have a username that will be used to install DB2. The username may belong to the Domain or local Administrators group.

Note: The above requirements apply to both the codeserver and the target thin client workstation.



The username will be removed from the system when the installation is complete.

Thin Client Security Considerations

The Thin Client Installation program performs three types of actions that are subject to security restrictions:

Updates the Local Machine's registry.

Accesses the Code Server's registry.

Optionally, creates a permanent NET USE for the Shared LAN Drive.

Windows NT Considerations:

Updates the Local Machine's registry.

This requires that the account you use to perform the DB2 installation be a Local Administrator (that is, a member of the local machines Administrator group). In addition, DB2 requires that your userid be 8 characters or less and meet its naming restrictions. If this is not the case, DB2 will attempt to create and use an account called DB2ADMIN (This account will be deleted after install).

Accesses the Code Server's registry.

Your account must be a member of the Administrators group on the Code Server machine. If you are a domain Administrator and the "Domain Administrators" group is a member of the code server's Administrator group then you will be able to access the registry. If not, your account (both userid and password) must be defined on the Code Server and that account must be part of the Administrator's group. In addition, if DB2 creates the Default userid of DB2ADMIN, this userid would have to be defined on the Code Server (password DB2ADMIN).

If this is NOT true then the install program will not be able to access the remote registry. As a result, the contents of your response file will not be verified and the registry on the local machine will be updated with default values. For most installations, this is not a problem.

Optionally, creates a permanent NET USE for the Shared LAN Drive.

It is recommended that you set up the LAN Share on the Code Server to provide everyone read access. In this case a userid and password is not required.

If a userid and password is provided it must be defined in the code server's security database. DB2 will set up a permanent net use for the user installing the thin client. If the password changes in the future, the NET USE information must be updated.

If you do not wish universal access and also do not wish to provide a userid and password to use to access the shared LAN drive, then the userid and password of the account used to install the thin client must be defined in the Code Server's security database.

Windows 95 and Windows 98 Considerations:

Updates the Local Machine's registry.

Windows 95 and Windows 98 have no notion of an Administrator. However, you must use an account that meets DB2's naming restrictions.

Accesses the Code Server's Registry.

Windows 95 and Windows 98 do not support the remote procedure call that must be performed to access the Code Server's registry. So the DB2 thin client installation will continue with default values.

Optionally, creates a permanent NET USE for the Shared LAN Drive.

Windows 95 and Windows 98 do not support provision for supplying a userid and password for the LAN Connection. You must either allow Everyone to access the drive (for read) or the userid and password of the account used to install the thin client must be defined in the Code Server's security database.

Step 1. Install DB2 Client Application Enabler on the Code Server

See Chapter 4, "Installing DB2 Clients on Windows 32-Bit Operating Systems" on page 17 for complete instructions on installing DB2 Client Application Enabler on the code server machine. For the thin client feature, keep the following considerations in mind:

You must install DB2 Client Application Enabler in a location which can be accessed in READ mode by the thin client workstations. Use Windows Explorer to set up a share for the entire drive, or just for the directory where DB2 Client Application Enabler is installed. For example, you can install the DB2 Client Application Enabler to `c:\codesrv\sqllib`, then a share can be set up for the directory `c:\codesrv`.

Also, if you intend on having a double code base, and service both Windows 95, Windows 98, and Windows NT clients, the code server must be a Windows NT machine. For example, the sample directory structure could be:

`c:\codesrv\sqllibNT`, `c:\codesrv\sqllib95`, and `c:\codesrv\sqllib98`, where the whole directory `c:\codesrv` is a share.

Note: If you are using a Windows 95 or Windows 98 machine as your code server, you cannot service Windows NT clients (only Windows 95 or Windows 98).

To set up a share:

- 1** Click on **Start** and select **Programs->Windows Explorer**.
- 2** Select the directory that you want to share, for example `c:\codesrv`.
- 3** Select **Files->Properties** from the menu bar. The properties window for the directory will open. Click on the **Sharing** tab.
- 4** Select the **Shared As** radio button.
- 5** In the **Share Name** field, enter a share name. For example, `db2nt`.
- 6** To specify read access for everyone on Windows NT:
 - a** Select the **Permissions** push button. The Access Through Share Permissions window opens.
 - b** In the **Name** box, make sure that **Everyone** is selected.
 - c** In the **Type of Access** box, select **Read**.
 - d** Click on **OK**.

On Windows 95 and Windows 98, you do not need to specify type of access when you set up a share. By default, everyone is allowed read access.

In our example scenario, `c:\codesrv` uses a share name of `db2nt`. We will use this value in the examples that follow.

We recommend that you install all components of DB2 Client Application Enabler. You can specify which components are installed on each client by means of the response file (see “Step 3. Create a Response File for Thin Client Installation” on page 238).

Step 2. Create Windows 95 or Windows 98 Code Base on Windows NT Code Server

If you intend to have both Windows NT, and Windows 95 or Windows 98 thin clients, running off the code server, you must have performed step 1 on a Windows NT machine. Step 1 provides the code base to install thin clients on other Windows NT machines.

To run a thin client on a Windows 95 or Windows 98 machine, you must perform the following three steps:

1. On the code server, copy the DB2 Client Application Enabler directory (for example, `sqllibNT`) into another directory, that will be used to service Windows 95 or Windows 98 clients (for example, `sqllib95` or `sqllib98`).
2. Delete the Windows NT versions of the following files:
`db2sysp.dll`

db2cmd.dll

db2w.dll

db2cliw.dll

which are found in the bin subdirectory of your new Windows 95 or Windows 98 code base (for example, c:\codesrv\sqllib95\bin or c:\codesrv\sqllib98\bin).

3. Rename the following files, also found in the bin subdirectory, to the file names you have just deleted in step 2:

db2syspw.dll -> db2sysp.dll

db2cmdw.dll -> db2cmd.dll

db2ww.dll -> db2w.dll

db2cliww.dll -> db2cliw.dll

You now have two code bases on your Windows NT code server. If you are installing a thin client on a Windows NT machine, install from the Windows NT code base (sqllibNT), and if you are installing a thin client on a Windows 95 or Windows 98 machine, use the Windows 95 or Windows 98 code base (sqllib95 or sqllib98).

Step 3. Create a Response File for Thin Client Installation

When you have done "Step 1. Install DB2 Client Application Enabler on the Code Server" on page 236, the DB2 Client Application Enabler on the code server will contain the directory `c:\codesrv\sqllib\thnsetup` which will contain all files needed for the thin client setup, including the sample response file.

During an interactive installation, you provide the information needed to install DB2 and configure its environment. During an unattended installation, this information is provided in the form of keywords and values in a response file. DB2 Client Application Enabler installed on the code server includes a ready-to-use sample response file, `db2thin.rsp`, with default entries already in place. This information will be applied when you install a thin client.



You can create a different response file, and therefore a different configuration, for each thin client installation. You can also edit and resave the response file to create one or two or more "standard" configurations, using the same ones as templates for multiple thin client installations.

Edit the sample response file.

- 1 The sample response file, `db2thin.rsp`, is located in the `c:\codesrv\sqllib\thnsetup` directory on the code server in our example.
- 2 The response file contains:

Keywords unique to installation

Registry value/environment variable settings

Database manager configuration parameter settings.



For more information on:

DB2 environment variables and registry values, see “DB2 Registry Values and Environment Variables” on page 163, or refer to the *Administration Guide*.

Database manager configuration parameters, refer to the *Administration Guide*.

- 3** To activate an item in the response file, remove the asterisk (*) to the left of the keyword/environment variable, erase the current setting to the right of the value and type in a new setting. The possible settings are listed to the right of the current setting. You can edit the keywords unique to installation contained within the response file. These are as follows:

REBOOT	= Yes, No
TYPE	= ,1,2 (=compact,1=typical,2=custom)
COMP	= GRAPHICAL_TOOLS
COMP	= CLIENT_CONFIGURATION_ASSISTANT
COMP	= CONTROL_CENTER
COMP	= WEB_CONTROL_CENTER
COMP	= VISUAL_EXPLAIN
COMP	= PERFORMANCE_MONITOR
COMP	= ODBC_DRIVER
COMP	= THIN_CLIENT_CODE_SERVER
COMP	= DOCUMENTATION
USERNAME	= (userid that the security service will run under)
PASSWORD	= (password for the security service id)



You will need to reboot the target workstation after installing. To have this done automatically, activate the **REBOOT** keyword.

The **TYPE** keyword specifies the type of install that you want.

Install type specifies a compact install. This option provides you with access to the client code, but none of the other components are accessible from the client workstation.

Install type 1 specifies a typical install, and is the default setting. With this option, all components defined on the code server are accessible from the client workstation.

Install type 2 specifies a custom install. This option allows you to install only selected components. Set the **TYPE** keyword to 2 (Custom Install) and

activate the **COMP** keyword for each component you want to be accessible from the client workstation.



On Windows NT, you should also specify a value for both USERNAME and, if applicable, PASSWORD. Specify a username that belongs to the local Administrator's group, is eight characters or less, and complies with DB2's naming rules.

If the username that you specify does not meet DB2's naming rules, but has the "Act as part of the operating system" advanced user right, the setup program will create a temporary username (that belongs to the Administrators group) called *DB2ADMIN*, with a password of *DB2ADMIN*. The setup program will use this username to perform the installation.

4 Edit the environment variables contained within the response file.

The following is a sample section of the environment variables contained within a response file:

```
DB2BQTIME           = BLANK or 1 - MAX
DB2BQTRY            = BLANK or - MAX
DB2CHKPTR           = BLANK, ON or OFF
DB2CODEPAGE         = BLANK or - MAX
DB2DBDFT            = BLANK or char(8)
DB2INSTPROF         = C:\CFG
DB2IQTIME           = BLANK or 1 - MAX
DB2OPTIONS          = BLANK or char()
```

...



The only environment variable setting that must be specified within the response file is *DB2INSTPROF*. To set up a separate client configuration for each thin client, set the *DB2INSTPROF* value to a directory on the thin client workstation. To maintain one client configuration on the code server, set *DB2INSTPROF* to a directory on the code server workstation.

5 Edit the configuration parameters contained within the response file.

The following is a sample section of the configuration parameters contained within a response file:

```
DB2.DIAGLEVEL       = - 4
DB2.DIAGPATH        = BLANK or char(215)
DB2.DIR_CACHE       = YES, NO
DB2.DIR_OBJNAME     = BLANK or char(255)
DB2.DIR_PATH_NAME   = BLANK or char(255)
DB2.DISCOVER        = DISABLE, KNOWN or SEARCH
```

...

6 Exit the file. If you have made any changes, save the file under a new file name to preserve the original sample response file.

Step 4. Make Thin Client Setup Program Accessible to the Target Workstation

This step gives the target workstation access to the thin client setup program, so that the thin client install can run. From the target workstation, perform one of the following:

NET USE the codeserver sharename explicitly and run the thnsetup.exe from the thnsetup directory directly. Use a command similar to the following:

```
net use x: \\machine\directory
```

where:

- *x* is the drive letter used to connect to the shared directory on the server's hard disk or CD-ROM drive.
- *machine* is the machine name of the code server.
- *directory* is the share name of the code server.

For example:

```
NET USE x: \\db2server\db2nt
```

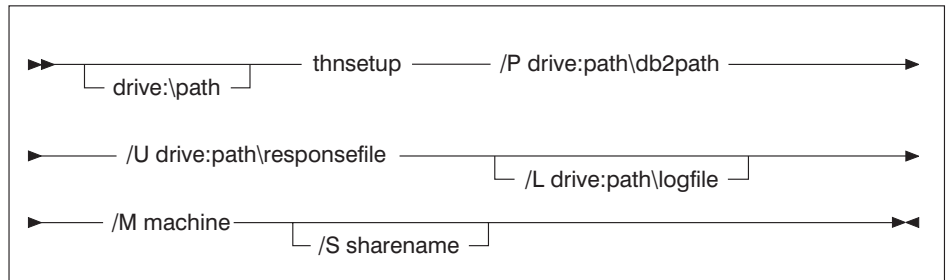
Copy all files located in `c:\codesrv\sql1lib\thnsetup` directory on the code server to a diskette. Run the install program from the diskette. It is not necessary to NET USE the code server share since the setup program can perform the NET USE for you.

Step 5. Install Thin Client on the Target Workstation

This step installs the thin client code on the target workstation, and sets up the links to the code server.

Note: Make sure that the thin client setup program is available to the client machine. You can accomplish this by performing either of the methods detailed in “Step 4. Make Thin Client Setup Program Accessible to the Target Workstation.”

- 1** At the workstation where the thin client code will be installed, logon as a username that meets the requirements for installing DB2. See “Creating a Username for Installing DB2 Products” on page 234 for more information.
- 2** Run the setup program:
 - a** Click on **Start**, and select the **Run** option.
 - b** In the **Open** field, type the path to the setup program. The path accepts the following command line arguments:



Specify the fully qualified path to invoke the program. For example:

```
x:\sqllib\thnsetup\thnsetup /P x:\sqllib
/U x:\sqllib\thnsetup\db2thin.rsp /L c:\logdir\db2log.txt
/M \\db2server /S db2path
```

where:

- /P** Specifies the path to the location where the client is installed on the code server. If you specify the sharename, then the install program will implicitly perform a NET USE for you. It will take the drive letter specified on this argument, the machine specified, and the sharename, and issue the NET USE.
- /U** Specifies the fully qualified response file name.
- /L** Specifies the fully qualified log file name, where setup information and any errors occurring during setup are logged.
If you do not specify the log file's name, DB2 names it `db2.log`, and stores it in a directory called `db2log`, on the drive on which your operating system is installed.
- /M** Specifies the machine name of the code server.
- /S** Specifies the share name of the code server drive/directory where DB2 Client Application Enabler is installed.
If this share requires a username and/or password, specify the share name as: `sharename,userid,password`.
If there is no password, type: `sharename,userid`.
If a username is not required, type: `sharename only`.

C Click on **OK** to start the setup program.

- 3** Check the messages in the log file when the installation finishes.
- 4** After the installation, the workstation must be rebooted before you use the thin client.



To configure the thin clients to access servers, go to Chapter 9, "Configuring Client-to-Server Communications Using the Client Configuration Assistant" on page 49.

Installing Thin Clients using SMS

To set up a code server, and use Microsoft Systems Management Server (SMS) to install one or more thin clients which will access it, perform the steps in the following sections:

"Step 1. Install DB2 Client Application Enabler on the Code Server" on page 244

"Step 2. Create a Response File for Thin Client Installation" on page 245

"Step 3. Import the Thin Client PDF File into SMS" on page 247

"Step 4. Distribute the Thin Client Installation Package Using SMS" on page 248

"Step 5. Run the Installation on the SMS Clients" on page 249

If you plan to install multiple thin clients in a row, perform all the steps for the first thin client, then perform only "Step 4. Distribute the Thin Client Installation Package Using SMS" on page 248 and "Step 5. Run the Installation on the SMS Clients" on page 249 for each subsequent thin client on that code server.

Before You Begin

Read and perform the steps in this section before you begin the installation to ensure that you have the required items and information you will need.

Creating a Username for Installing DB2 Products

Windows 95 and Windows 98

The username 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 C, "Naming Rules" on page 279.

Windows NT

You need to have a username that will be used to install DB2. The username may belong to the Domain or local Administrators group.

Note: The above requirements apply to both the codeserver and the target thin client workstation.

SMS Requirements

SMS must already be installed and configured on your network. Refer to *Microsoft's Systems Management Server for Windows NT Administrator's Guide* for information on how to:

- Set up SMS (including setting up primary and secondary sites)
- Add clients to the SMS system
- Set up inventory collection for clients.

Step 1. Install DB2 Client Application Enabler on the Code Server

See Chapter 4, "Installing DB2 Clients on Windows 32-Bit Operating Systems" on page 17 for complete instructions on installing DB2 Client Application Enabler on the code server machine. For the thin client feature, keep the following considerations in mind:

You must install DB2 Client Application Enabler in a location which can be accessed in READ mode by the thin client workstations. Use Windows Explorer to set up a share for the entire drive, or just for the directory where DB2 Client Application Enabler is installed. For example, you can install the DB2 Client Application Enabler to `c:\codesrv\sqllib`, then a share can be set up for the directory `c:\codesrv`.

To set up a share:

- 1** Click on **Start** and select **Programs->Windows Explorer**.
- 2** Select the directory that you want to share, for example `c:\codesrv`.
- 3** Select **Files->Properties** from the menu bar. The properties window for the directory will open. Click on the **Sharing** tab.
- 4** Select the **Shared As** radio button.
- 5** In the **Share Name** field, enter a share name. For example, `db2nt`.
- 6** To specify read access for everyone on Windows NT:
 - a** Select the **Permissions** push button. The Access Through Share Permissions window opens.
 - b** In the **Name** box, make sure that **Everyone** is selected.
 - c** In the **Type of Access** box, select **Read**.
 - d** Click on **OK**.

On Windows 95 and Windows 98, you do not need to specify type of access when you set up a share. By default, everyone is allowed read access.

In our example scenario, `c:\codesrv` uses a share name of `db2nt`. We will use this value in the examples that follow.

We recommend that you install all components of DB2 Client Application Enabler. You can specify which components are installed on each client by means of the response file (see “Step 3. Create a Response File for Thin Client Installation” on page 238).

In this example, we will assume that you are installing DB2 Client Application Enabler on the code server where SMS is installed. If you are installing to a different location, you must NET USE to this share in order to make the **thinsetup** install accessible to SMS.

Step 2. Create a Response File for Thin Client Installation

When you are done installing DB2 Client Application Enabler, the code server will contain a directory `c:\codesrv\sqllib\thinsetup` which will contain all files needed for the thin client setup, including the sample response file, `db2thin.rsp`.

During an interactive installation, you provide the information needed to install DB2 and configure its environment. During an unattended installation, this information is provided in the form of keywords and values in a response file. DB2 Client Application Enabler installed on the code server includes a ready-to-use sample response file for the thin client install, with default entries already in place. This information will be applied when you install a thin client.



You can create a different response file, and therefore a different configuration, for each thin client installation. You can also edit and resave the response file to create one or two or more "standard" configurations, using the same ones as templates for multiple thin client installations.

Edit the sample response file using your favorite editor. For example, you can use Notepad or WordPad.

- 1** The sample response file, `db2thin.rsp`, is located in `c:\codesrv\sqllib\thinsetup` directory on the code server in our example.
- 2** The response file contains:
 - Keywords unique to installation
 - Registry value/environment variable settings
 - Database manager configuration parameter settings.



For more information on:

DB2 environment variables and registry values, see “DB2 Registry Values and Environment Variables” on page 163, or refer to the *Administration Guide*.

Database manager configuration parameters, refer to the *Administration Guide*.

- 3** To activate an item in the response file, remove the asterisk (*) to the left of the keyword/environment variable, erase the current setting to the right of the value and type in a new setting. The possible settings are listed to the right of the current setting. You can edit the keywords unique to installation contained within the response file. These are as follows:

```
REBOOT          = Yes, No
TYPE            = ,1,2 ( =compact,1=typical,2=custom)
COMP           = GRAPHICAL_TOOLS
COMP           = CLIENT_CONFIGURATION_ASSISTANT
COMP           = CONTROL_CENTER
COMP           = WEB_CONTROL_CENTER
COMP           = VISUAL_EXPLAIN
COMP           = PERFORMANCE_MONITOR
COMP           = ODBC_DRIVER
COMP           = THIN_CLIENT_CODE_SERVER
COMP           = DOCUMENTATION
USERNAME        = (userid that the security service will run under)
PASSWORD       = (password for the security service id)
```



You will need to reboot the target workstation after installing. To have this done automatically, activate the **REBOOT** keyword.

The **TYPE** keyword specifies the type of install that you want.

Install type specifies a compact install. This option provides you with access to the client code, but none of the other components are accessible from the client workstation.

Install type 1 specifies a typical install, and is the default setting. With this option, all components defined on the code server are accessed from the client workstation.

Install type 2 specifies a custom install. This option allows you to install only selected components. Set the **TYPE** keyword to 2 (Custom Install) and activate the **COMP** keyword for each component you want to access.



The only environment variable setting that must be specified within the response file is *DB2INSTPROF*. To set up a separate client configuration for each thin client, set the *DB2INSTPROF* value to a directory on the thin client workstation. To maintain one client configuration on the code server, set *DB2INSTPROF* to a directory on the code server workstation.

4 Edit the environment variables contained within the response file.

The following is a sample section of the environment variables contained within a response file:

```
DB2BQTIME           = 1 -
DB2BQTIME           = BLANK or 1 - MAX
DB2BQTRY            = BLANK or - MAX
DB2CHKPTR           = BLANK, ON or OFF
DB2CODEPAGE         = BLANK or - MAX
DB2DBDFT            = BLANK or char(8)
DB2INSTPROF         = C:\CFG
DB2IQTIME           = BLANK or 1 - MAX
DB2OPTIONS          = BLANK or char()
...
```

5 Edit the configuration parameters contained within the response file.

The following is a sample section of the configuration parameters contained within a response file:

```
DB2.DIAGLEVEL       = - 4
DB2.DIAGLEVEL       = - 4
DB2.DIAGPATH        = BLANK or char(215)
DB2.DIR_CACHE       = YES, NO
DB2.DIR_OBJNAME     = BLANK or char(255)
DB2.DIR_PATH_NAME   = BLANK or char(255)
DB2.DISCOVER        = DISABLE, KNOWN or SEARCH
...
```

6 Exit the file. If you have made any changes, save the file under a new file name to preserve the original sample response file.

Step 3. Import the Thin Client PDF File into SMS

To import the thin client PDF files into SMS:

- 1 From the **Start** button select **Programs**, then **Systems Management Server**, and then **SMS Administrator**. The SMS Administrator window opens.
- 2 Open the Package Properties window. The first time you start the SMS Administrator program you will see a dialog box prompting you to open an SMS Window. You can select the Packages option from there. Otherwise, select **File->New** from the menu bar.
- 3 Click on the **Import** push button. The **File Browser** opens.

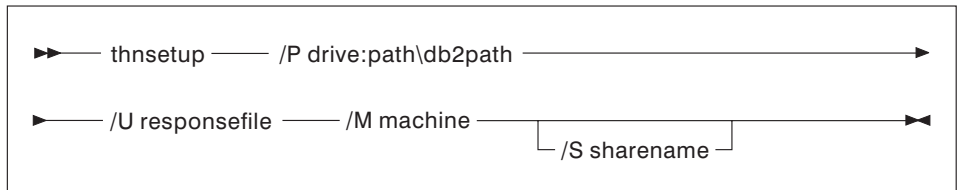
- 4 In the **File Name** field, select the `db2thin.pdf` file from the `c:\codesrv\sqllib\thnsetup` directory on the code server.
- 5 Click on **OK** to return to the Package Properties window.

Step 4. Distribute the Thin Client Installation Package Using SMS

In this step, you create an SMS package and send it to the SMS client where you want to install the thin client.

To create the SMS package:

- 1 Click on the **Workstations** push button. The Setup Package For Workstations window opens, showing a list of predefined DB2 installation packages that you can use.
- 2 In the **Source Directory** field, type the location of the thin client install files, `c:\codesrv\sqllib\thnsetup`.
- 3 Select the name of the product to install from the **Workstation Command Lines** list.
- 4 Click on the **Properties** push button and update the command line for the **thnsetup** program. **Thnsetup** accepts the following command line arguments:



Specify the relative filepath to invoke the program. For example:

```
thnsetup.exe /U db2thin.rsp /P x:\sqllib /M \\db2server /S db2nt
```

where:

- /P** Specifies the path to the location where the client is installed on the code server. If you specify the sharename, then the install program will implicitly perform a NET USE for you. It will take the drive letter specified on this argument, the machine specified, and the sharename, and issue the NET USE.
- /U** Specifies the relative path to the response file.
- /M** Specifies the machine name of the code server.

- /S** Specifies the share name of the code server drive/directory where DB2 Client Application Enabler is installed.
- If this share requires a username and/or password, specify the share name as: `sharename,userid,password`.
- If there is no password, type: `sharename,userid`.
- If a username is not required, type: `sharename` only.
- 5** Click on **OK** to return to the Setup Package For Workstations window.
 - 6** Click on the **Close** push button to return to the Package Properties window.
 - 7** The Packages Properties window shows the name of the new SMS package.
 - 8** To specify where to run the package:
 - a** Open the Sites and Packages windows.
 - b** In the Packages window, select the appropriate package and drag it onto the target client in the Sites window. The Job Details window opens.
 - c** Select the **Run Workstation Command** check box and select the installation package that you want to use for the Run Command on Workstation job. For more information about completing the Job Details window, refer to *Microsoft's Systems Management Server for Windows NT Administrator's Guide*.
 - d** When the job specifications are complete, click on **OK** to close the opened windows.

The job is created and sent to the SMS client.

Step 5. Run the Installation on the SMS Clients

Now that the remote installation package has been prepared and sent, the SMS client on the target machine can run the installation.

- 1** On the SMS client where DB2 will be installed, logon as a username that meets the requirements for installing DB2. See "Creating a Username for Installing DB2 Products" on page 243 for more information.
- 2** Double-click on the **SMS Client** group.
- 3** Double-click on the **Package Command Manager** icon. The Package Command Manager window opens.
- 4** When packages are sent from the SMS server, they are listed in the window. Select the thin client package, then click on the **Execute** push button. The installation runs automatically.
- 5** After the installation, reboot the SMS client before using the thin client.



To configure the thin clients to access servers, go to Chapter 9, “Configuring Client-to-Server Communications Using the Client Configuration Assistant” on page 49.

Chapter 19. Using the Client Configuration Assistant

Use the information in this section to use the Client Configuration Assistant (CCA) to perform administration tasks on OS/2 and Windows 32-bit operating systems.

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 and Windows 32-bit operating systems. 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 databases to be configured as CLI or 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.
- Change your password.

Configuring Database Connections



To complete the steps in this section, you must be logged on to the local system as a user with System Administrative (SYSADM) authority on the instance.



You can configure database connections through the Add Database SmartGuide, by:

Using a profile as a source of information to add database connections, see “Using an Access Profile” on page 252.

Searching the network for databases, see “Searching the Network for Databases” on page 255.

Adding database connections manually, see “Manually Configure a Connection to a DB2 Database” on page 260.

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 in the Administration Server's configuration file 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 259.

Generating a Server Profile: To generate an access profile, perform the following steps:

- 1 Click on **Start** and select **Programs->DB2 for Windows NT->Administration Tools->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 this 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 Click on **Start** and select **Programs->Client Configuration Assistant**.
- 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 and select the profile for the server that you want to access.
- 5 Select a database to be added and click on the **Next** push button.
- 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 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.)

Import function, which allows you to selectively process database connection data, client configuration settings, and CLI or 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 or ODBC settings).
- Client settings (including database manager configuration parameters).
- CLI or ODBC common parameters.
- Configuration data for the local APPC communications subsystem.

The Export function can be used to generate a customized profile that can be imported on another client to set it up, or 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 belong to the local Administrator group 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. To add the *admin* parameter to the CCA startup:

- 1 Select the **CCA** icon and click on the right mouse button.

- 2 Select the **Properties** option

- 3 Add the *admin* parameter to the:

OS/2 **Parameters** field

&win32; **Target** field (located on the **Security** tab)

right click on the CCA icon and select the **Properties** option by entering *admin* in the **Parameters** field on OS/2 or appending *admin* to the **Target** field on Windows 32-bit operating systems.

- 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.

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:: Perform these steps at the client that you want to set up.

- 1 Click on **Start** and select **Programs->DB2 for Windows NT->Administration Tools->Control Center**.
- 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. To run the following scenario, you must ensure that the *discover* parameter in the database manager configuration file on the client, and the *discover* parameter in the Administration Server configuration file on the server are set to *SEARCH*. Also, the *discover_comm* parameter must be set to the search protocols used on the client and Administration Server.

- 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 a 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 a 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 retrieve 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.

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.



The TCP/IP host name returned to a client by Search discovery is the same host name that is returned by your DB2 server system when you enter the *hostname* command. On the client, the IP address that this host name maps to is determined by either the TCP/IP domain name server (DNS) configured on your client machine or, if a DNS is not configured, a mapping entry in the client's *hosts* file. If you have multiple adapter cards configured on your DB2 server system, you must ensure that TCP/IP is configured on the server to return the correct hostname on a , and that the DNS or local client's *hosts* file, maps the hostname to the IP address desired.

Search may appear to be a simpler discovery method. However, in larger networks, network routers and bridges can filter the messages that the 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 configuration file to *KNOWN*. To have it support Search discovery, set this parameter to *SEARCH* (which also supports all the facilities of Known discovery). To prevent discovery of a 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** push button. When the *discover* parameter is set to *KNOWN*, the CCA will not be able to search the network.
- Search** Enables all the facilities of Known discovery, and enables network searching.
- DISABLE** Disables discovery. In this case, the **Search the network** option is not available in the Add Database SmartGuide.



The *discover* parameter defaults to *SEARCH* on all client and server instances. The *discover* parameter defaults to *SEARCH* on all Administration Servers except Administration Servers installed in a UNIX Extended Enterprise Edition environment, where *discover* defaults to *KNOWN*.

Additional Settings for Search Discovery

Search discovery requires that the *discover_comm* parameter 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 to 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's configuration file. On the client (or a server acting as a client), *discover_comm* is set in the database manager configuration file.

Note: When using Search discovery, at least one protocol specified by the *discover_comm* parameter on the client must match those specified by the *discover_comm* parameter 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 entering the **db2set DB2COMM** command. For more information, see Chapter 15, "Controlling Your DB2 Environment" on page 157.

In addition, there are two DB2 profile registry values that can be used to tune Search discovery via NetBIOS 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 configuration file on the server system, and in the database manager configuration file on the client. Set these parameters as follows:

On the Administration Server:

Update the Administration Server's configuration file, in the command processor, as follows:

```
update admin cfg using discover [ DISABLE | KNOWN | SEARCH ]
update admin cfg using discover_comm [ TCPIP ]
```

Stop and restart the Administration Server by entering the following commands:

```
db2admin stop
db2admin start
```

Note: Search Discovery will only operate on TCP/IP.

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 the parameters that you want to modify from the **Parameters** window.
- 5 Select a value for the parameter that you want to modify from the **Value** box.
- 6 Click on the **OK** push button to close the **Client Settings** window. A DB2 message window opens.
- 7 Click on the **OK** push button and restart your applications so that your changes can take affect.

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 Click on the **[+]** sign beside the **Systems** icon to get a list of systems.
- 3 Click on the **[+]** sign beside a system icon to get a list of instances on it.
- 4 Select the instance that you want to configure and click on the right mouse button.
- 5 Select the **Configure** option from the pop-up menu. The Configuration Instance window opens.

- 6 Select the **Environment** tab and select the *discover_inst* parameter from the **Parameters** box.
- 7 Select the desired value from the **Value** box and click on **OK**.
- 8 Click on the **[+]** beside an instance icon to get a list of databases in it.
- 9 Select the database that you want to configure and click on the right mouse button.
- 10 Select the **Configure** option from the pop-up menu. The Configuration Database window opens.
- 11 Select the **Environment** tab and select the *discover_db* parameter.
- 12 Select the desired value from the **Value** box and click on **OK**.

db2discovertime and db2nbdiscoverrcvbufs

The *db2discovertime* 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 *db2discovertime* registry value to 60 seconds, enter the following command:

```
db2set db2discovertime=60
```

This specifies that Search discovery should wait 60 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 and click on the **Next** push button.
- 5 Enter the required protocol parameters and click on the **Next** push button.
- 6 In the **Target database** panel, enter the name of the remote database alias in the **Target Database** field.



When a database is created on the remote server, if a database alias is not specified during database creation, the database is created with a database alias=*database_name*; otherwise, the database alias is the name specified.

- 7 If desired, you can select the **Alias** and **ODBC** tabs at the top of the panel to specify a local database alias name or select **CLI/ODBC** options for the remote 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.



When you add a database using this configuration method, the Client Configuration Assistant will generate a default node name for the database.

For example, if you used TCP/IP to add the database to your system, the instance name could be *TCP0000*.

Part 5. Appendixes

Appendix A. How the DB2 Library Is Structured

The DB2 Universal Database library consists of SmartGuides, online help, and books. This section describes the information that is provided, and how to access it.

To access product information online, you can use the Information Center. You can view task information, DB2 books, troubleshooting information, sample programs, and DB2 information on the Web. See "Information Center" on page 275 for details.

SmartGuides

SmartGuides help you complete some administration tasks by taking you through each task one step at a time. SmartGuides are available through the Control Center. The following table lists the SmartGuides.

Note: Not all SmartGuides are available for the partitioned database environment.

SmartGuide	Helps you to...	How to Access...
<i>Add Database</i>	Catalog a database on a client workstation.	From the Client Configuration Assistant, click on Add .
<i>Create Database</i>	Create a database, and perform some basic configuration tasks.	From the Control Center, click with the right mouse button on the Databases icon and select Create->New .
<i>Performance Configuration</i>	Tune the performance of a database by updating configuration parameters to match your business requirements.	From the Control Center, click with the right mouse button on the database you want to tune and select Configure performance .
<i>Backup Database</i>	Determine, create, and schedule a backup plan.	From the Control Center, click with the right mouse button on the database you want to backup and select Backup->Database using SmartGuide .
<i>Restore Database</i>	Recover a database after a failure. It helps you understand which backup to use, and which logs to replay.	From the Control Center, click with the right mouse button on the database you want to restore and select Restore->Database using SmartGuide .

SmartGuide	Helps you to...	How to Access...
<i>Create Table</i>	Select basic data types, and create a primary key for the table.	From the Control Center, click with the right mouse button on the Tables icon and select Create->Table using SmartGuide .
<i>Create Table Space</i>	Create a new table space.	From the Control Center, click with the right mouse button on the Table spaces icon and select Create->Table space using SmartGuide .

Online Help

Online help is available with all DB2 components. The following table describes the various types of help. You can also access DB2 information through the Information Center. For information see "Information Center" on page 275.

Type of Help	Contents	How to Access...
<i>Command Help</i>	Explains the syntax of commands in the command line processor.	From the command line processor in interactive mode, enter: <i>? command</i> where <i>command</i> is a keyword or the entire command. For example, ? catalog displays help for all the CATALOG commands, while ? catalog database displays help for the CATALOG DATABASE command.
<i>Control Center Help</i>	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.

Type of Help	Contents	How to Access...
<i>Message Help</i>	Describes the cause of a message, and any action you should take.	<p>From the command line processor in interactive mode, enter:</p> <p>? XXXnnnnn</p> <p>where <i>XXXnnnnn</i> is a valid message identifier.</p> <p>For example, ? SQL30081 displays help about the SQL30081 message.</p> <p>To view message help one screen at a time, enter:</p> <p>? XXXnnnnn more</p> <p>To save message help in a file, enter:</p> <p>? XXXnnnnn > filename.ext</p> <p>where <i>filename.ext</i> is the file where you want to save the message help.</p>
<i>SQL Help</i>	Explains the syntax of SQL statements.	<p>From the command line processor in interactive mode, enter:</p> <p>help statement</p> <p>where <i>statement</i> is an SQL statement.</p> <p>For example, help SELECT displays help about the SELECT statement.</p>
<i>SQLSTATE Help</i>	Explains SQL states and class codes.	<p>From the command line processor in interactive mode, enter:</p> <p>? sqlstate or ? class-code</p> <p>where <i>sqlstate</i> is a valid five-digit SQL state and the <i>class-code</i> is first two digits of the SQL state.</p> <p>For example, ? 08003 displays help for the 08003 SQL state, while ? 08 displays help for the 08 class code.</p>

DB2 Books

The table in this section lists the DB2 books. They are divided into two groups:

Cross-platform books These books contain the common DB2 information for UNIX-based and Intel-based platforms.

Platform-specific books These books are for DB2 on a specific platform. For example, for DB2 on OS/2, on Windows NT, and on the UNIX-based platforms, there are separate *Quick Beginnings* books.

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.

If you want to read the English version of the books, they are always provided in the directory that contains the English documentation.

You can obtain DB2 books and access information in a variety of different ways:

- View** See "Viewing Online Books" on page 272.
- Search** See "Searching Online Books" on page 273.
- Print** See "Printing the PostScript Books" on page 273.
- Order** See "Ordering the Printed DB2 Books" on page 274.

Book Name	Book Description	Form Number File Name
Cross-Platform Books		
<i>Administration Getting Started</i>	Introduces basic DB2 database administration concepts and tasks, and walks you through the primary administrative tasks.	S10J-8154 db2k0x50
<i>Administration Guide</i>	Contains information required to design, implement, and maintain a database to be accessed either locally or in a client/server environment.	S10J-8157 db2d0x51
<i>API Reference</i>	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.	S10J-8167 db2b0x51
<i>CLI Guide and Reference</i>	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.	S10J-8159 db2l0x50
<i>Command Reference</i>	Explains how to use the command line processor, and describes the DB2 commands you can use to manage your database.	S10J-8166 db2n0x51

Book Name	Book Description	Form Number File Name
<i>DB2 Connect Enterprise Edition Quick Beginnings</i>	Provides planning, migrating, installing, configuring, and using information for DB2 Connect Enterprise Edition. Also contains installation and setup information for all supported clients.	S10J-7888 db2cyx51
<i>DB2 Connect Personal Edition Quick Beginnings</i>	Provides planning, installing, configuring, and using information for DB2 Connect Personal Edition.	S10J-8162 db2c1x51
<i>DB2 Connect User's Guide</i>	Provides concepts, programming and general using information about the DB2 Connect products.	S10J-8163 db2c0x51
<i>DB2 Connectivity Supplement</i>	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.	No form number db2h1x51
<i>Embedded SQL Programming Guide</i>	Explains how to develop applications that access DB2 databases using embedded SQL, and includes discussions about programming techniques and performance considerations.	S10J-8158 db2a0x50
<i>Glossary</i>	Provides a comprehensive list of all DB2 terms and definitions. Note: Available in HTML format only.	No form number db2t0x50
<i>Installing and Configuring DB2 Clients</i>	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.	No form number db2iyx51
<i>Master Index</i>	Contains a cross reference to the major topics covered in the DB2 library. Note: Available in PostScript format and hardcopy only.	S10J-8170 db2w0x50
<i>Message Reference</i>	Lists messages and codes issued by DB2, and describes the actions you should take.	S10J-8168 db2m0x51
<i>DB2 Replication Guide and Reference</i>	Provides planning, configuring, administering, and using information for the IBM Replication tools supplied with DB2.	S95H-0999 db2e0x52
<i>Road Map to DB2 Programming</i>	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.	S10J-8155 db2u0x50

Book Name	Book Description	Form Number File Name
<i>SQL Getting Started</i>	Introduces SQL concepts, and provides examples for many constructs and tasks.	S10J-8156 db2y0x50
<i>SQL Reference</i>	Describes SQL syntax, semantics, and the rules of the language. Also includes information about release-to-release incompatibilities, product limits, and catalog views.	S10J-8165 db2s0x51
<i>System Monitor Guide and Reference</i>	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.	S10J-8164 db2f0x50
<i>Troubleshooting Guide</i>	Helps you determine the source of errors, recover from problems, and use diagnostic tools in consultation with DB2 Customer Service.	S10J-8169 db2p0x50
<i>What's New</i>	Describes the new features, functions, and enhancements in DB2 Universal Database, Version 5.2, including information about Java-based tools.	S04L-6230 db2q0x51
Platform-Specific Books		
<i>Building Applications for UNIX Environments</i>	Provides environment setup information and step-by-step instructions to compile, link, and run DB2 applications on a UNIX system.	S10J-8161 db2axx51
<i>Building Applications for Windows and OS/2 Environments</i>	Provides environment setup information and step-by-step instructions to compile, link, and run DB2 applications on a Windows or OS/2 system.	S10J-8160 db2a1x50
<i>DB2 Personal Edition Quick Beginnings</i>	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.	S10J-8150 db2i1x50
<i>DB2 SDK for Macintosh Building Your Applications</i>	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.	S50H-0528 sqla7x02
<i>DB2 SDK for SCO OpenServer Building Your Applications</i>	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.	S89H-3242 sqla9x02
<i>DB2 SDK for SINIX Building Your Applications</i>	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.	S50H-0530 sqla8x00

Book Name	Book Description	Form Number File Name
<i>Quick Beginnings for OS/2</i>	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.	S10J-8147 db2i2x50
<i>Quick Beginnings for UNIX</i>	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.	S10J-8148 db2ixx51
<i>Quick Beginnings for Windows NT</i>	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.	S10J-8149 db2i6x50
<i>DB2 Extended Enterprise Edition for UNIX Quick Beginnings</i>	Provides planning, installing, configuring, and using information for DB2 Universal Database Extended Enterprise Edition for UNIX. This book supercedes the <i>DB2 Extended Enterprise Edition Quick Beginnings for AIX</i> book, and is suitable for use with all versions of DB2 Extended Enterprise Edition that run on UNIX-based platforms.	S99H-8314 db2v3x51
<i>DB2 Extended Enterprise Edition for Windows NT Quick Beginnings</i>	Provides planning, installing, configuring, and using information for DB2 Universal Database Extended Enterprise Edition for Windows NT.	S09L-6713 db2v6x51

Notes:

1. The character in the sixth position of the file name indicates the language of a book. For example, the file name `db2d e5` 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	Turkish	M

2. For late breaking information that could not be included in the DB2 books:

On UNIX-based platforms, see the `Release.Notes` file. This file is located in the `DB2DIR/Readme/%L` directory, where `%L` is the locale name and `DB2DIR` is:

- /usr/lpp/db2_5_ on AIX
- /opt/IBMDB2/V5. on HP-UX, Solaris, SCO UnixWare 7, and SGI.

On other platforms, see the `RELEASE.TXT` file. This file is located in the directory where the product is installed.

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, and provides hypertext links to related information. It also makes it easier to share the library across your site.

You can use any HTML Version 3.2-compliant browser to view the online books.

To view online books:

If you are running DB2 administration tools, use the Information Center. See “Information Center” on page 275 for details.

Use the open file function of your Web browser. The page you open contains descriptions of and links to DB2 books:

- On UNIX-based platforms, open the following page:

```
file:/INSTHOME/sqlllib/doc/%L/html/index.htm
```

where %L is the locale name.

- On other platforms, open the following page:

```
sqlllib\doc\html\index.htm
```

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.

Note: The **DB2 Online Books** icon is only available if you do not install the Information Center.

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 `\sqlllib\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, see *Setting up DB2 Online Documentation on a Web Server* at:

<http://www.software.ibm.com/data/pubs/papers/db2html.html>

3. If you are using the Java version of the Information Center, you can specify a base URL for all HTML files. You should use the URL for the list of books.
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 *ALTER TABLE* topic
- Search form.

For information about setting up a search, see the *What's New* book.

Searching Online Books

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. See "Information Center" on page 275 for details.

Printing the PostScript Books

If you prefer to have printed copies of the manuals, you can decompress and print PostScript versions. For the file name of each book in the library, see the table in "DB2 Books" on page 268.

Note: Specify the full path name for the file you intend to print.

On OS/2 and Windows platforms:

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\books\ps` directory, where `x`: is the letter representing the CD-ROM drive and `language` is the two-character country code that represents your language (for example, EN for English).

2. Decompress the file that corresponds to the book that you want. The result from this step is a printable PostScript file with a file extension of `.psz`.
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.psz
```

On UNIX-based platforms:

1. Mount the CD-ROM. Refer to your *Quick Beginnings* manual for the procedures to mount the CD-ROM.
2. Change to `/cdrom/doc/%L/ps` directory on the CD-ROM, where `/cdrom` is the mount point of the CD-ROM and `%L` is the name of the desired locale. The manuals will be installed in the previously-mentioned directory with file names ending with `.ps.Z`.
3. Decompress and print the manual you require using the following command:

For AIX:

```
zcat filename | qprt -P PSprinter_queue
```

For HP-UX, Solaris, or SCO UnixWare 7:

```
zcat filename | lp -d PSprinter_queue
```

For Silicon Graphics IRIX and SINIX:

```
zcat < filename | lp -d PSprinter_queue
```

where *filename* is the name of the full path name and extension of the compressed PostScript file and *PSprinter_queue* is the name of the PostScript printer queue.

For example, to print the English version of *Quick Beginnings for UNIX* on AIX, you can use the following command:

```
zcat /cdrom/doc/en/ps/db2ixe5 .ps.Z | qprt -P ps1
```

Ordering the Printed DB2 Books

You can order the printed DB2 manuals either as a set, or individually. There are two sets of books available. The form number for the entire set of DB2 books is SBOF-8915- . The form number for the books listed under the heading "Cross-Platform Books" is SBOF-8914- .

Note: These form numbers only apply if you are ordering books that are printed in the English language.

You can also order books individually by the form number listed in "DB2 Books" on page 268. To order printed versions, contact your IBM authorized dealer or marketing representative, or phone 1-800-421-879-2755 in the United States or 1-800-421-4YOU in Canada.

Information Center

The Information Center provides quick access to DB2 product information. You must install the DB2 administration tools to obtain 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
- Help menu of the Control Center
- db2ic** command.

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 some search capabilities so you can look for specific topics, and filter capabilities to limit the scope of your searches.

For a full text search, follow the *Search DB2 Books* link in each HTML file, or use the search feature of the help viewer.

The HTML search server is usually started automatically. If a search in the HTML information does not work, you may have to start the search server via its icon on the Windows or OS/2 desktop.

Refer to the release notes if you experience any other problems when searching the HTML information.

Appendix B. National Language Support (NLS)

This section contains information about the National Language Support (NLS) provided by DB2, including information about supported languages and code pages on OS/2 and Windows operating environments, and supported locales and code sets on UNIX operating systems. For more detailed information on developing applications, refer to the *Embedded SQL Programming Guide*.

Language and Codeset Support for UNIX Operating Systems

DB2 supports many code sets and locales without translating the messages for the corresponding languages. Supporting a locale means that you can create and use a database in that locale, but you may have to view all panels and messages in a different language, if translated messages are not available in DB2. For a complete list of locales supported, refer to the *Administration Guide*.

If you want to operate in a different language environment, do the following:

- 1 Ensure that the appropriate message option for the desired language has been installed.
- 2 Set the *LANG* environment variable to the desired locale.

For example, to use *fr_FR* messages on DB2 for AIX, you must have the *fr_FR* message option installed and must set *LANG* to *fr_FR*.

The selected message catalog filesets are placed in the following directories on the target workstation:

DB2 for AIX	<code>/usr/lpp/db2_5_ /msg/%L</code>
DB2 for HP-UX	<code>/opt/IBMdb2/V5. /msg/%L</code>
DB2 for SCO UnixWare 7	<code>/opt/IBMdb2/V5. /msg/%L</code>
DB2 for Solaris	<code>/opt/IBMdb2/V5. /msg/%L</code>

where *%L* is equal to the locale name of the message catalog.

Code Page and Language Support for OS/2 and Windows Operating Environments

During installation of DB2, the country, codepage and regional settings are established. You can, however, change these settings after installing DB2. This includes changing regional settings such as code page, country language (for monetary, date, and numeric formatting) and time zone. When a new connection to a database is made, the database manager uses these new values.



You must ensure that your regional settings are set correctly. DB2 may not produce the expected results if the country, code page, or regional settings are incorrect for the intended language.

Table 23 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\win32\install` would be `x:\en\win32\install` for English. For more detailed information on the languages and code pages support, refer to the *Administration Guide*.

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	TR	Turkish
		TW	Traditional Chinese (Taiwan)

Appendix C. 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 280
 - "Object Names" on page 280
 - "Usernames, Group Names, and Instance Names" on page 281
 - "DB2SYSTEM Naming Rules" on page 282
 - "Workstation Names (nname)" on page 282
 - "Password Rules" on page 282
-

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 279.

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 279.

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 these rules.

Username, Group Name, and Instance Names

Username are the identifiers assigned to individual users. When naming users, groups, or instances, see “General Naming Rules” on page 279.

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 32-bit operating systems, use any case.
- On UNIX, use lowercase names.

Workstation Names (nname)

A workstation name is 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 279.

In addition, the name you specify:

- Can contain 1 to 8 characters;
- Cannot include &, #, and @;
- Must be unique within the network.

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 32-bit operating systems, 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 279.

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 D. 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	MVS/ESA
ADSTAR	MVS/XA
AISPO	NetView
AIX	OS/400
AIXwindows	OS/390
AnyNet	OS/2
APPN	PowerPC
AS/400	QMF
CICS	RACF
C Set++	RISC System/6000
C/370	SAA
DATABASE 2	SP
DatagLANce	SQL/DS
DataHub	SQL/400
DataJoiner	S/370
DataPropagator	System/370
DataRefresher	System/390
DB2	SystemView
Distributed Relational Database Architecture	VisualAge
DRDA	VM/ESA
Extended Services	VSE/ESA
FFST	VTAM
First Failure Support Technology	WIN-OS/2
IBM	
IMS	
Lan Distance	

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, Visual Basic, and the Windows 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 Operation.

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.

Appendix E. 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 accessing the following page:

<http://www.ibm.com/support/>

then performing a search using the keyword "handbook."

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
--

Index

Special Characters

- .INI file
 - db2cli.ini 145
 - ODBC 145

A

- Access Profile
 - creating 87, 90, 96, 100, 102, 103, 107, 252
 - using 252
- accessing multiple servers 55
- accessing servers
 - IPX/SPX 77
 - Named Pipes 55
 - NetBIOS 55, 69
 - overview 55
 - TCP/IP 60
- APPC
 - software requirements 7, 8, 9
 - support on AIX 87
 - support on HP-UX 91
 - support on OS/2 96
 - Communications Manager for OS/2 7
 - Communications Server for OS/2 7
 - support on Solaris
 - SunLink SNA 8, 102
 - supported platforms 85
 - verifying the connection 114
- APPENDAPINAME, keyword 173
- application development
 - using ODBC 133
- ASYNCEENABLE, keyword 173
- Asynchronous ODBC, enabling 173
- authentication
 - client 11, 42
 - DCS 42
 - server 42
- authority
 - levels 119, 123

B

- binding
 - utilities and applications 132
- BITDATA, keyword 174

C

- cataloging
 - APPC node 111, 112
 - databases 57, 65, 73, 81, 112
 - IPX/SPX node 79, 81
 - Named Pipe node 56, 57
 - NetBIOS node 72
 - TCP/IP node 64, 65
- CD-ROM
 - installing DB2 Client Applications Enablers 11
 - installing DB2 Software Developer's Kit 11, 13
- Client Application Enabler
 - installing 11
 - installing on OS/2 13
 - installing on Windows 3.x 21
 - software requirements 5
- Client Configuration Assistant
 - administering clients communications 49
 - client profiles 49
 - exporting 253
 - using 253
 - discover 257
 - discovering databases 50
 - searching the network 49, 255
 - setting configuration parameters 169, 259
 - testing the database connection 53
 - using access profiles 252
- client profile
 - exporting 253
- clients
 - configuring 55
 - installing 11
 - operating systems supported 11
 - overview 11
 - verifying the installation 53
- CLP
 - See using the command line processor (CLP)

Command Center
 entering DB2 commands 151
 entering SQL statements 151

commands

db2icrt 124, 125
db2idrop 129
db2ilist 126
db2imigr 42, 43
db2iupdt 128
db2licd 42
db2ln 39
db2set 158
db2stop 127, 129
echo 159, 160
groupadd 37
mount 31
rlogin 32
SAM, to update kernel 27
set 159
SYSADM 37
useradd 37

communication protocols

APPC 5, 6, 7, 8, 9, 85
IPX/SPX 6, 7, 8, 9, 77
Named Pipes 7, 8, 9, 55
NetBIOS 7, 8, 9, 69
TCP/IP 5, 6, 7, 8, 9, 60

 configuring 60

communications

 configuring the client 49, 55, 60, 69, 77, 85
 managing 55

 Client Configuration Assistant 49

 supported protocols

 APPC 85
 Named Pipes 55
 NetBIOS 69
 TCP/IP 60, 77

configuration parameters

 setting DB2 11, 157, 169, 233

configuring

 APPC 85
 IPX/SPX 77
 Named Pipes 55
 NetBIOS 69
 ODBC driver 136, 138
 steps to configure SNAPplus 92
 steps to configure SunLink SNA 102
 TCP/IP 60

configuring client communications

 setting configuration parameters 55, 169

 configuring client communications (*continued*)
 using the Client Configuration Assistant 49

 using the command line processor 55

 configuring communications

 overview 55

 using Client Configuration Assistant 49

 configuring server communications

 setting DB2COMM 252

 connecting to a database 120

 CONNECTTYPE, keyword 174

 creating

 instances 124, 125

 CURRENTFUNCTIONPATH, keyword 175

 CURRENTPACKAGESET, keyword 175

 CURRENTSQLID, keyword 176

 CURSORHOLD, keyword 177

D

database

 naming rules 280

 database alias 57, 66, 73, 81, 112

 naming rules 280

database manager

 starting 126

 stopping 126

database objects

 naming rules 280

DB2 Client Application Enabler

installing

 Windows 3.x 21

 Windows 95 17

 Windows NT 17

 overview 11

DB2 Client Application Enablers CD-ROM

 installing clients from 11

DB2 Extended Enterprise Edition

 disk requirements 3

DB2 Installer

 See installation

DB2 library

 books 268

 Information Center 275

 language identifier for books 271

 late breaking information 271

 online help 266

 ordering printed books 274

 printing PostScript books 273

 searching online books 273

 setting up document server 272

- DB2 library (*continued*)
 - SmartGuides 265
 - structure of 265
 - viewing online books 272
- DB2 server, definition ix
- db2 terminate command 41
- DB2 Universal Database
 - getting started 119
 - software requirements 5
- DB2 Workgroup Edition
 - disk requirements 3
- db2.ini file
 - environment keywords 161
- db2cli.ini 145
- DB2DEGREE, keyword 177
- DB2DMNBCKCTLR
 - registry value 164
- DB2ESTIMATE, keyword 178
- DB2EXPLAIN, keyword 179
- db2icrt command 38, 124, 125
- db2idrop command 129
- db2ilist command 126
- db2imigr command 43
- DB2INSTANCE 126
- db2iupdt command 128
- db2ln command 39
- DB2OPTIMIZATION, keyword 180
- db2profile script 39
- db2set command
 - using 158
- db2stop command 127, 129
- DBALIAS, keyword 180
- DBNAME, keyword 181
- DEFAULTPROCLIBRARY, keyword 181
- DEFERREDPREPARE, keyword 182
- DISABLEMULTITHREAD, keyword 183
- Discovery
 - choosing a discovery method 257
 - searching the network 255
 - setting Discovery parameters 259
- disk requirements
 - client 3
 - server 3
- Distributed Computing Environment
 - software requirements 5, 6, 7, 8

E

- EARLYCLOSE, keyword 183

- environment variables
 - DB2INSTANCE 123, 126, 167
 - db2instprof 167
 - db2path 168
 - db2profile script 26
 - setting 39

F

- files
 - db2system 129
- fixed disks
 - hardware requirements 3

G

- GATEWAYVERSION, keyword 184
- GRANTEELIST, keyword 184
- GRANTORLIST, keyword 185
- GRAPHIC, keyword 185

H

- hardware requirements
 - fixed disk 3

I

- IGNOREWARNINGS, keyword 186
- initialization file, ODBC 145
- installation
 - CID using SystemView LAN 223
 - clients
 - AIX 25
 - HP-UX 25
 - Macintosh 45
 - OS/2 13, 21
 - SCO OpenServer 36
 - SINIX 35
 - Solaris 25
 - Windows 3.x 21
 - Windows 95 17
 - Windows NT 17
 - DB2 Installer 125
 - DB2 server on UNIX systems
 - assigning groups and users 37
 - creating an instance 38
 - creating groups and users 37
 - creating links 39
 - setting the DB2 operating environment 38, 39
 - updating kernel components 26, 27, 28, 35

installation (*continued*)

- disk requirements
 - client 3
 - server 3
- errors 223
- log 223
- mounting the CD-ROM 28
 - AIX 29
 - HP-UX 30
 - mount point 25
 - SINIX 35
 - Solaris 31
- remote 32
- unattended
 - OS/2 205, 212
 - Windows 3.x 205
 - Windows 95 215
 - Windows NT 215
- instances
 - creating 34, 124, 125
 - creating multiple 123
 - creating the default 123
 - definition 123
 - listing 126
 - naming restrictions 25, 125, 281
 - removing 128
 - selecting 126
 - starting 126
 - stopping 126, 127
 - updating 128
- IPX/SPX
 - client 77
 - configuring 77
 - software requirements 7, 8
 - supported platforms 77
 - verifying the connection 53, 59, 67, 75, 83, 114

J

- Java, running programs 147
- JDBC, running programs 147

K

- KEEPCONNECT, keyword 187
- KEEPSTATEMENT, keyword 187
- kernel configuration parameters
 - updating 26

L

- LANG environment variable 277
- LOBMAXCOLUMNSIZE, keyword 188
- logging off the system 119
- logging on to the system 119
- LONGDATACOMPAT, keyword 188

M

- managing connections
 - client
 - overview 55
 - using the Client Configuration Assistant 49
 - using the command line processor 55
 - server
 - using the Client Configuration Assistant 49
 - using the Command Line Processor 55
- MAXCONN, keyword 189
- Microsoft ODBC Driver Manager 135
- Microsoft SNA Server
 - version required 9
- migration
 - data 42
 - DB2 Version 2.x 41
 - instance 42, 43
 - post-installation 41
- MODE, keyword 189
- MULTICONNECT, keyword 190

N

- Named Pipes
 - client 55
 - configuring 55
 - software requirements 7, 8
- naming rules
 - database 280
 - database alias 280
 - database objects 280
 - general 279
 - groups 281
 - instance names 281
 - password 282
 - userids 281
 - username 17, 281
 - workstation name (nname) 282
- National Language Support (NLS)
 - code set 277
 - codepage support 277, 278

National Language Support (NLS) (*continued*)
 determining codepage 278
Net.Data 7
NetBIOS
 on the client 7, 8, 69
 supports SEARCH discovery 258
Netfinity Server 7
NLS
 See National Language Support (NLS)

O

ODBC
 odbc.ini file 145
 odbcinst.ini file 145
 registering the driver manager 135
 running ODBC-enabled applications 133
 running programs 133
odbcad32.exe 135
operating systems
 server 11
OPTIMIZEFORNROWS, keyword 190

P

Parallelism, Setting degree of 177
parameters
 setting kernel configuration 26
password
 naming rules 282
PATCH1, keyword 191
PATCH2, keyword 191
POPUPMESSAGE, keyword 192
profile registry
 using 157
protocols
 APPC 85
 IPX/SPX 77
 Named Pipes 55
 NetBIOS 69
 TCP/IP 60
PWD, keyword 192

R

registering
 ODBC driver manager 135
registry value
 DB2DMNBCKCTL 164

registry values
 db2account 163
 db2clientadpt 168
 db2clientcomm 168
 db2codepage 163
 db2country 163
 db2dbdft 163
 db2dbmsaddr 164
 db2dirpathname 168
 db2discoverytime 164
 db2include 165
 db2instdef 123, 127, 165
 db2libpath 165
 db2netreq 165
 db2options 165
 db2route 168
 db2slogon 166
 db2timeout 166
 db2tracename 166
 db2traceon 166
 db2trcflush 167
 db2trcsyserr 167
 db2yield 167
 dsmi_config 169
 dsmi_dir 169
 dsmi_log 169
 general 163
 setting 157, 233
remote installation
 OS/2 212
 Netfinity Server 212
response files
 keywords 238
 OS/2 207
 Thin clients 238
 Windows 3.x 207
 Windows 32-bit operating systems 218
restrictions
 instance name 125, 281
running applications
 database client 131
 ODBC considerations 133

S

SCHEMALIST, keyword 193
script
 db2cshrc 39, 161
 db2profile 39, 161

- selecting an instance 126
- setting
 - environment variables 157, 233
- setting configuration parameters 55
- setting up client communications
 - using the Client Configuration Assistant 49
 - using the command line processor 55
- setting up document server 272
- SNA
 - starting the SNAPPlus subsystem 95
 - steps to configure SNAPPlus 92
 - steps to configure SunLink SNA 102
- snapconfig program
 - description 102
- SNAPPlus
 - steps to configure 92
- snapmanage 95
- software requirements
 - communication protocols 5
 - DB2 Client Application Enabler 5, 6, 7, 8, 9
 - DB2 Connect 5
 - DB2 SDK 5, 6, 7, 8, 9
 - DB2 Universal Database 5
 - Net.Data 5, 7, 8, 9
- SQLSTATEFILTER, keyword 193
- SYNCPOINT, keyword 194
- SYSCTRL 123
- SYSMAINT 123
- SYSSHEMA, keyword 194
- system environment variables
 - DB2INSTANCE 126
- system management 126

T

- TABLETYPE, keyword 195
- TCP/IP
 - client 60
 - configuring 60
 - preventing socket collisions 61
 - setting up client-to-server 60
 - software requirements 5, 6, 7, 8
 - supports SEARCH discovery 258
 - troubleshooting 61
- TEMPDIR, keyword 196
- TRACE, keyword 197
- TRACEFILENAME, keyword 197
- TRACEFLUSH, keyword 198
- TRACEPATHNAME, keyword 198

- TRANSLATEDLL, keyword 199
- TRANSLATEOPTION, keyword 200
- TXNISOLATION, keyword 200

U

- UID, keyword 201
- unattended installation
 - OS/2 212
 - Netfinity Server 212
 - Windows 32-bit operating systems
 - Server Management System (SMS) 223, 243
- UNDERSCORE, keyword 201
- username
 - naming rules 281
- using the command line processor (CLP)
 - cataloging a database 57, 65, 73, 81, 112
 - cataloging a node 56, 64, 72, 79, 111
 - connecting to a database 52, 59, 67, 75, 83, 114, 120
 - updating the database manager configuration 72
 - verifying the connection 53
- utilities
 - binding 131

V

- verifying the connection
 - using the Client Configuration Assistant 53
 - using the command line processor
 - APPC 85
 - IPX/SPX 77
 - Named Pipes 55
 - NetBIOS 69
 - TCP/IP 60
- verifying the installation
 - client 53

W

- WARNINGLIST, keyword 202

IBM

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.