

IBM[®] DB2[®] Universal Database



Troubleshooting Guide

Version 7

IBM[®] DB2[®] Universal Database



Troubleshooting Guide

Version 7

Before using this information and the product it supports, be sure to read the general information under "Appendix D. Notices" on page 295.

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 the United States 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, 2000. All rights reserved.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Welcome to DB2	ix
Who Should Read This Book?	ix
Conventions	ix
How to Read the Syntax Diagrams.	x

Part 1. Basic DB2 Troubleshooting **1**

Chapter 1. Good Troubleshooting Practices **3**

Maintaining a Problem History	3
Determining Origins and Symptoms of Problems	4
Determining the Location of a Problem	4
Responding to Unexpected Messages or SQLCODEs	5
Responding to Abends	6
Responding to Suspensions and Loops	7
Using All Available Information.	9
Updating DB2 Products	10
Updating DB2 Products Using Fix Packs	10
Using Logged Data and Taking Traces	11
Contacting IBM for Support.	11

Chapter 2. Troubleshooting the DB2

Universal Database Server **13**

DB2 Administrative Tools	14
Graphical Tools	14
Miscellaneous Troubleshooting Tools	15
The Database System Monitor	18
Troubleshooting Tips	19
Problems Migrating to Version 7	19
Problems When Applying Fix Packs	20
Server Failures	20
Starting DB2 on Windows 95 and Windows 98 When the User Is Not Logged On	22
Using a Windows 2000 Terminal Server environment	26
SQL1403N Received When Authenticating Usernames or Groups.	27
Administration Difficulties	27
Backing Up and Restoring Data	28
Data Not Loading	33
Difficulties Importing Data	34

DB2 Support for Raw Devices when Using Windows 2000	34
Disk Alignment when Using Windows 2000.	34
Problems Using Commands, Utilities, or the Command Line Processor	35
Locking and Deadlocks	36
Improving Database Performance.	38
Still Have Problems?	40

Chapter 3. Troubleshooting on the Client **43**

Overview of Client Connectivity	43
Client Connectivity Tools	44
Troubleshooting Tips	45
Determining the Scope of a Client Problem	46
Initial Connection after Installation Fails	47
Clients Suddenly Experiencing Problems	48
User Name Not Valid on Windows 95 and Windows 98	49
TCP/IP Problems	49
IPX/SPX Problems.	53
NetBIOS Problems	55
Problems When Using Named Pipes	58
APPC Problems.	59
Still Have Problems?	59

Chapter 4. Troubleshooting Host Communications **61**

Troubleshooting Host Connections	62
Direct Connection	62
Gateway Connection	64
Troubleshooting Tips for DB2 Connect	64
Problems Encountered after Initial Connection	65
Changing the Number of Connections	65
Authentication Problems.	66
Windows Clients Do Not Connect.	67
Troubleshooting Tips for Host Connections to DB2 Universal Database	67
Troubleshooting Installation Problems on Windows 2000	68
Still Have Problems?	68

Chapter 5. Troubleshooting Applications **71**

Troubleshooting Tips	72
--------------------------------	----

Developing Applications	72	Backup Data Links Manager Environment	109
Compiling and Linking Applications	73	Export Operation	110
Running Applications	74	Import Operation	110
Using CLI/ODBC/JDBC Applications	75	Load Considerations	111
Using SQLJ Applications	79	DB2 Data Links Manager for Windows	
Using the db2ocat Tool	80	NT	112
Stored Procedures	82	Still Have Problems?.	112
Still Have Problems?	83		
Chapter 6. Troubleshooting DB2 Query		Chapter 9. Troubleshooting Replication	113
Patroller	85	DB2 DataPropagator	114
Still Have Problems?	85	Using the Log Header Control File with	
		db2flsn	114
		Example of Using the db2flsn Command	
		and the db2diag.log File	115
		DB2 DataPropagator Enhancements	115
		Still Have Problems?.	116
Chapter 7. Troubleshooting DB2 Universal		Chapter 10. Troubleshooting the Data	
Database Enterprise - Extended Edition	87	Warehouse Center	117
Troubleshooting Tips	87	Prerequisite or related product issues	117
DB2 Universal Database Enterprise -		A DataJoiner Version 2 CREATE SERVER	
Extended Edition Will Not Install	88	MAPPING statement for an Oracle data	
Determining a Failing Node.	88	source fails	117
Database Manager Will Not Start	88	Unable to connect to a DB2 database from	
Cannot Create Database	91	a Windows NT or Windows 2000 agent	
Commands Cannot Be Issued	92	site.	117
Problems Backing Up On Database		Cannot connect to an ODBC data source	
Partitions	92	on UNIX.	119
Lost Database Partitions on Windows 2000	92	Unable to use TCP/IP to connect to	
Database Partition Size Rounding on		AS/400 from Windows NT or Windows	
Windows 2000	93	2000	119
Problems with LOAD and with Splitting		Connecting to a Sybase source fails	120
and Loading Data	93	FTP log displays "NAMEFMT not a valid	
Problems with Indexing Data	96	command"	121
Problems during Recovery	96	The warehouse server or warehouse agent	
Developing Applications in DB2 Enterprise		is unresponsive when using ODBC	
- Extended Edition	98	drivers that were purchased separately	122
Severe Error Occurs on System.	99	Installation issues	122
Additional Enterprise - Extended Edition		Windows NT or Windows 2000 cannot	
Considerations	101	install services for Data Warehouse Center	
First Failure Data Capture	101	components.	122
DB2 Trace Facility.	101	Initialization fails at "Add install path to	
Other Methods.	102	NT service path" message	123
General Instructions When in a Bad State	103	Initialization fails with a "3043 Disk or	
Hierarchy of Shutdowns	103	Network Error" message	123
Still Have Problems?.	103	Data Warehouse Center files disappear	
		after you install the product again	123
		The Data Warehouse Center will not start	124
		Initialization fails with an "Unable to	
		connect" message	124
Chapter 8. Troubleshooting DB2 Data			
Links Manager	105		
DB2 Data Links Manager Considerations	105		
DB2 Data Links Manager	106		
Data Access Token	106		
Recovery Data Links Manager			
Environment	107		

Initialization fails with a 9159: Win32 function RegOpenKeyEx failed message	124	An file access error occurs when using the Data Warehouse Center ODBC text file driver or accessing files using warehouse programs	135
Product function issues	125	Column is no longer visible in the column grid	136
A Data Warehouse Center component is unable to communicate with another component	125	Supplied warehouse program issues	137
The Data Warehouse Center and the AS/400 agent are unable to communicate	126	Step running VWPRCPY fails with RC2 = 12	137
The warehouse logger stops while the Data Warehouse Center is running	126	On AS/400, the VWPFTP trace says that the job completed successfully, but data was not transferred and the FTP message log is missing	137
A warehouse agent stops processing when promoting a step	127	One of the OLAP Server warehouse programs cannot find the rules file or load file	137
Creation of an AIX or Solaris Operating Environment warehouse agent process fails	127	User-defined programs are not found, or they run with errors	138
The Data Warehouse Center stops processing if you run several steps at the same time	128	Still Have Problems?	139
Cannot import tables from AS/400	128	Chapter 11. Troubleshooting DB2 Satellite Edition	141
Columns defined as BINARY WITH PRECISION are interpreted as INTEGER	129	Still Have Problems?	142
Prepare of SQL statement fails	129	Chapter 12. Troubleshooting NetQuestion	143
The Data Warehouse Center displays return code 7356	129	Still Have Problems?	143
Promoting a step that uses DB2 for MVS Version 4 Release 1 as the source and an AS/400 agent hangs	131	<hr/>	
Promoting a step that uses DB2 for OS/390 as the warehouse target and the AS/400 agent fails	132	Part 2. Advanced DB2 Troubleshooting	145
The Data Warehouse Center displays the "Could not retrieve the step RC- 3142" message when creating a step with a long name	132	Chapter 13. Logs and Traces Roadmap	147
The warehouse server cannot create a warehouse agent or it did not receive a valid reply back from the warehouse agent	133	Chapter 14. The DB2 Process Model	151
The Start/End Date/Time is incorrect in the Work In Progress window.	134	Description of the DB2 Process Model	151
DB2 reports an SQL0973N error when you open and close steps	134	Differences between Intel and UNIX	154
A step that uses Microsoft Access as a source fails	134	Viewing System Threads and Processes	155
The Data Warehouse Center supplies incorrect table names or an ODBC OLE error occurs when using the Microsoft Excel 95/97 ODBC driver to access Excel spreadsheets	135	Chapter 15. Logged Information	157
		First Failure Data Capture	157
		Setting Configuration Parameters Affecting FFDC	158
		Interpreting the db2diag.log	160
		Interpreting db2diag.log Entry Headers	160
		Interpreting an SQLCA Structure.	162
		Interpreting Hexadecimal Codes	163
		Some Examples for Interpreting the db2diag.log	163
		Dump Files	169
		Trap Files	170

Gathering Stack Traceback Information on UNIX-Based Systems	170
Using the Administrative Tools Log	171
Administrative Tools Log File Sample	171
Using Message Files	172
Alerts	173
SNMP Alerts	173
Chapter 16. Traces	177
Using the DB2 Trace Facility (db2trc)	177
Starting a DB2 Trace	179
Verifying a DB2 Trace	180
Examples of Using db2trc	180
Taking Other Traces	183
Administration Tools Trace	183
Taking a DB2 Connect Trace (ddcstrc)	184
Taking a DRDA Trace (db2drdat)	185
Taking an SNA Trace	186
Using the DB2 CLI/ODBC/JDBC Trace Facility	186
Chapter 17. Diagnostic Tools for UNIX-Based Systems	207
Using the System Error Log (syslog)	207
Setting Up the System Error Log	208
Understanding the System Error Log	209
Using the Core File	210
Accessing Core File Information	210
Viewing Process Status Using the ps Command	212
Other Commands	213
AIX Troubleshooting Commands	213
UNIX-Based Troubleshooting Commands Performance Monitoring Tools	213
Commands for DB2 Enterprise - Extended Edition	214
Chapter 18. Diagnostic Tools for OS/2	217
Using the OS/2 Logging Facility	218
Preliminary Considerations	218
Using OS/2 Logging Facility Commands	218
Using the FFST for OS/2 Utility	219
Adding the FFST for OS/2 Program Group to the Desktop Manager Window	220
Configuring and Enabling the FFST for OS/2 Utility	220
Accessing FFST for OS/2 Error Records	223
Routing Generic Alerts	224
Using the pstat Command	225

Chapter 19. Diagnostic Tools for Windows 32-bit Operating Systems	229
Diagnostic Tools for Windows 95 and Windows 98	229
Diagnostic Tools for Windows NT	229
Using Administrative Tools Provided by Windows NT	230
Taking Screen Captures	231
Using DB2-Supplied Tools	231
Chapter 20. GUI and Monitoring Tools	233
Database System Monitor	233
Using the Snapshot Monitor	233
Using the Event Monitor	234
Control Center	235
Using Search Utility for DB2 Online Information	236
Troubleshooting Tips	237
Troubleshooting Page	238
Stored Procedure Builder	238
DB2 Index Advisor	239

Chapter 21. Table and Table Space Recovery	241
Recovery Concepts	241
Recoverable and Non-Recoverable Databases	241
Database Logs	242
Point of Recovery	243
Dropped Table Point-in-Time Recovery	243
Table Space OFFLINE	245
Archive Logging	245
Table Space Usability	246
RESTART ... DROP PENDING TABLESPACE	246
RESTART ... DROP PENDING Table Space State	247
Table Space Usability	247
RESTART ... DROP PENDING	248
Database Directories	248
Database Files	249

Part 3. Appendixes 251

Appendix A. DB2 Internal Return Codes	253
Appendix B. SQL Communications (SQLCA)	271
Viewing the SQLCA Interactively	271

SQLCA Field Descriptions	271	Using DB2 Wizards	292
Order of Error Reporting	275	Setting Up a Document Server	293
DB2 Enterprise - Extended Edition Usage of the SQLCA	276	Searching Information Online	294
Appendix C. Using the DB2 Library	277	Appendix D. Notices	295
DB2 PDF Files and Printed Books	277	Trademarks	298
DB2 Information	277	Index	301
Printing the PDF Books	286	Contacting IBM	309
Ordering the Printed Books	287	Product Information	309
DB2 Online Documentation	288		
Accessing Online Help	288		
Viewing Information Online	290		

Welcome to DB2

Welcome to the world of DB2 troubleshooting.

This guide provides information to get you started solving problems with DB2 servers and clients. It helps you to:

- Identify problems or errors in a concise manner
- Solve problems based on their symptoms
- Learn about available diagnostic tools
- Develop a troubleshooting strategy for day-to-day operations

Who Should Read This Book?

This guide is intended for customers, users, database administrators (DBAs), application developers, and technical support representatives for DB2 clients and servers. To use it, you should be familiar with:

- Communications, relational database, and local area network (LAN) concepts
- Hardware and software requirements and options
- The overall configuration of your network
- Application programs and other facilities that run on your network
- Basic DB2 administrative tasks (see the *Administration Guide: Implementation*)
- The information on installation and early tasks described in the *Quick Beginnings* guides for the products you installed

For more information on the documentation available in the DB2 library, see “Appendix C. Using the DB2 Library” on page 277.

Conventions

You will find this book easier to use if you look for these conventions:

- Information is provided for all platforms currently supported by DB2. Unless information is identified as being particular to one or more platforms, it is applicable to all.
- The panels, screens, and menus shown are only examples and may not exactly match what you see on your workstation.
- The format for date and time, and the characters used as separators for date and time, are intended for a system that is configured to use the United States national language format. These may be different on your display, depending on the national language format used by your system.
- This book uses the following highlighting conventions:

- **Boldface type** indicates the name of an item on your screen; for example, a field, folder, icon, or menu. It is also used to indicate a command name, a utility name, and to emphasize important words or phrases.
- Monospace type indicates an example or a command as you should type it.
- *Italic type* is used for book titles and configuration variables. When used in command strings, it indicates a variable value that you should replace with your value.

How to Read the Syntax Diagrams

Throughout this book, syntax is described using syntax diagrams.

Read the syntax diagrams from left to right and top to bottom, following the path of the line.

The **▶▶**— symbol indicates the beginning of a statement.

The —**▶** symbol indicates that the statement syntax is continued on the next line.

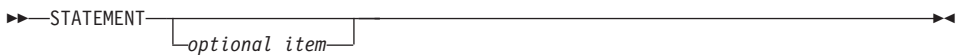
The **▶**— symbol indicates that a statement is continued from the previous line.

The —**▶▶** symbol indicates the end of a statement.

Required items appear on the horizontal line (the main path).



Optional items that are part of the syntax appear below the main path.

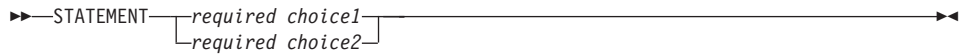


If an optional item appears above the main path, that item has no effect on the execution of the statement and is used only for readability.

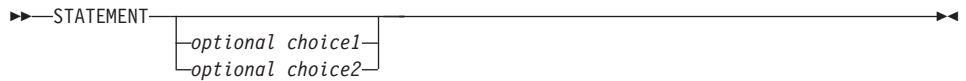


If you can choose from two or more items, they appear in a stack.

If you *must* choose one of the items, one item of the stack appears on the main path.



If the statement is complete without any of the items, the entire stack appears below the main path.



If one of the items is the default, it appears above the main path and the remaining choices are shown below.



An arrow returning to the left, above the main line, indicates an item that can be repeated. In this case, repeated items must be separated by one or more blanks.



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



A repeat arrow above a stack indicates that you can make more than one choice from the stacked items or repeat a single choice.

Keywords appear in uppercase (for example, FROM). They must be spelled exactly as shown. Variables appear in lowercase (for example, column-name). They represent user-supplied names or values in the syntax.

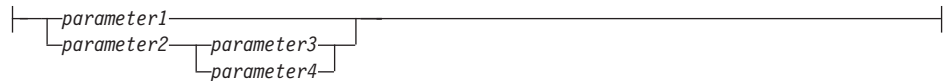
If punctuation marks, parentheses, arithmetic operators, or other such symbols are shown, you must enter them as part of the syntax.

Placeholder variables, such as *path*, appear in italics. Replace them with the appropriate value when you key the syntax.

Sometimes a single variable represents a set of several parameters. For example, in the following diagram, the variable parameter-block can be replaced by any of the interpretations of the diagram that is headed **parameter-block:**



parameter-block:



Adjacent segments occurring between “large bullets” (●) may be specified in any sequence.



The above diagram shows that item2 and item3 may be specified in either order. Both of the following are valid:

STATEMENT item1 item2 item3 item4
STATEMENT item1 item3 item2 item4

Part 1. Basic DB2 Troubleshooting

This part addresses the following topics:

- “Chapter 1. Good Troubleshooting Practices” on page 3
- “Chapter 2. Troubleshooting the DB2 Universal Database Server” on page 13
- “Chapter 3. Troubleshooting on the Client” on page 43
- “Chapter 4. Troubleshooting Host Communications” on page 61
- “Chapter 5. Troubleshooting Applications” on page 71
- “Chapter 6. Troubleshooting DB2 Query Patroller” on page 85
- “Chapter 7. Troubleshooting DB2 Universal Database Enterprise - Extended Edition” on page 87
- “Chapter 8. Troubleshooting DB2 Data Links Manager” on page 105
- “Chapter 9. Troubleshooting Replication” on page 113
- “Chapter 10. Troubleshooting the Data Warehouse Center” on page 117
- “Chapter 11. Troubleshooting DB2 Satellite Edition” on page 141
- “Chapter 12. Troubleshooting NetQuestion” on page 143.

Chapter 1. Good Troubleshooting Practices

This chapter emphasizes the following troubleshooting practices:

- “Maintaining a Problem History”
- “Determining Origins and Symptoms of Problems” on page 4
- “Using All Available Information” on page 9
- “Updating DB2 Products” on page 10
- “Using Logged Data and Taking Traces” on page 11
- “Contacting IBM for Support” on page 11

Maintaining a Problem History

Maintain a history of reported problems and the actions you took to identify or resolve them. This information may help you isolate later problems and anticipate or avoid others. In particular, you should record the following information related to a problem:

- The problem description, including:
 - The problem symptoms.
 - A list of the DB2 utilities that were running, including **RUNSTATS**, **REORG**, **LOAD**, **IMPORT**, and others.
 - A complete description of any messages that were received, including **SQLCODEs**, associated reason codes, or system error codes.
 - Any **SQLSTATE** received. **SQLSTATEs** are useful for diagnosing problems, because they are consistent across all platforms. For a list of **SQLSTATEs**, see the *Message Reference*.
- What was happening at the time. By recording information such as the following, you can spot patterns should a problem occur again:
 - The actions that led up to the problem
 - Any applications that were running at the time
 - Any SQL statements that were running at the time, including information on whether they were dynamic or static, and Data Definition Language (DDL) or Data Manipulation Language (DML)
- The type of hardware and software running on your system, and their configuration
- The service level installed on each system or partition server (on Enterprise - Extended Edition systems), and any fix packs applied
- The results of any corrective actions
- The problem resolution

- The number assigned to the problem, if you contacted DB2 Customer Support.

Determining Origins and Symptoms of Problems

If you are supporting an organization using DB2, you will receive calls from users to resolve a variety of problems. Your response depends on:

- The severity of the problem
- The specific nature of the problem
- Any related information that you can gather
- Your experience in resolving similar problems

To solve a problem, start by obtaining a comprehensive description of the problem. This way, you can begin to determine its origin. For example, a problem may exist in any of the following:

- Hardware
- Operating system
- Networking system or other subsystem
- DB2 server
- DB2 client
- DB2 Connect gateway to host systems

Determining the Location of a Problem

Most applications run in a client/server environment. You must determine if a problem is on the client, the server, or somewhere in between (that is, in the LAN or communication protocol stack).

Investigating where the problem is detected or reported is the best way to start. For example, if you receive an unexpected SQLCODE on a client, then investigate the SQLCODE on that client. (See “Responding to Unexpected Messages or SQLCODEs” on page 5 for information.)

Often the SQLCODE alone provides enough information to determine the source and cause of the problem. If the SQLCODE does not give enough information to determine the source of a problem, examine the `db2diag.log` file at the partition server where the problem was reported. For example, if the problem was reported on a client, first look at the `db2diag.log` file on that particular client.

The `db2diag.log` file is an ASCII file written by DB2 that contains diagnostic information for DB2. The `db2diag.log` file reports exceptions encountered in DB2 code. If you know the date and time when the problem occurred, you can go directly to the corresponding `db2diag.log` file entries.

Note: The error messages relating to a user application problem most often do not cause exceptions in DB2. This type of problem is handled as a normal part of DB2 processing. As a result, they are not reported in the `db2diag.log` file.

For information on this important file, see “First Failure Data Capture” on page 157. When viewing the file, keep in mind that the most recent conditions are always at the end.

Responding to Unexpected Messages or SQLCODEs

When you receive an unexpected message or SQLCODE, follow these steps until you can determine the problem:

1. When you receive a message, take note of all available information, including the following:
 - The code, an eight-digit alphanumeric message identification number. This code may begin with the prefix SQL, DBA or CCA. Also note all reason codes, return codes, and other information associated with the message returned.
 - Any SQLSTATE received. SQLSTATEs are useful for diagnosing problems, because they are consistent across all platforms. For a list of SQLSTATEs, see the *Message Reference*.
 - The text of the message (especially if the message does not include an identification number or a code).
 - The SQLCA if available.
 - Any action suggested in the message.
 - Diagnostic files, such as the `db2diag.log` file. In addition, note any operating system diagnostic files such as traceback files, core files (for UNIX-based systems), event logs (for Windows NT), `syslog` files (for OS/2), and any dumps. See “Part 2. Advanced DB2 Troubleshooting” on page 145.
 - The environment in which the message occurred. For example, what the user was doing at the time, the steps that led up to the problem, the type of operating system, applications that were running, and the communication protocol. Also note if any utilities such as **RUNSTATS**, **REORG**, **LOAD**, **IMPORT**, or others were being used.
 - The SQL statement that encountered the error, and any preceding statements in the unit of work.
2. Check the online message help by typing `db2 "? message"` from the command prompt, where *message* is the complete SQLCODE, SQLSTATE, or message number. Read and follow the suggested actions.
3. Use the SQLCODE or message number to search available DB2 documentation for additional information.

4. If the problem persists, ensure that you have as much of the following information as possible before contacting DB2 Customer Support:
 - The DB2 diagnostic log (db2diag.log file), and any trap or dump files it mentions. See “First Failure Data Capture” on page 157.
 - The SQLCA structure from the db2diag.log file or syslog files, or as captured by an application. See “Interpreting an SQLCA Structure” on page 162.
5. If you determine that the problem is not with DB2 but with a vendor-supplied application, contact the vendor.

Responding to Abends

In this book the term *abend* includes:

- Segmentation violations and general protection faults (GPFs) on Windows systems
- Traps on OS/2
- Exceptions on UNIX-based systems

When an abend occurs, work through the following steps until you can determine the problem:

1. Confirm that all DB2 components (clients, servers, DB2 Connect, each partition server in an Enterprise - Extended Edition system) are at the same service level, especially if a fix pack has recently been installed. See “Updating DB2 Products” on page 10.
2. Note the executable module that reported the abend.
3. If the problem persists, try to collect the following additional information before contacting DB2 Customer Support:
 - Any logged information, in particular:
 - All the files in the *diagpath* as specified in the database manager configuration file.
 - The db2diag.log file and any trap or dump files it mentions. See “First Failure Data Capture” on page 157.
 - For UNIX, the syslog and core files. See “Chapter 17. Diagnostic Tools for UNIX-Based Systems” on page 207.
 - For OS/2, the syslog file. See “Using the syslog Command” on page 218.
 - For Windows operating systems, Dr. Watson logs, and, for Windows NT, the event log. See “Chapter 19. Diagnostic Tools for Windows 32-bit Operating Systems” on page 229.
 - If the problem can be reproduced and you are willing to abend the system again, a trace on the client and server may be helpful. Follow the steps in “Example of Tracing to a File” on page 181.

See “Part 2. Advanced DB2 Troubleshooting” on page 145 for information.

4. If you determine that the problem is not with DB2 but with a vendor-supplied application, contact the vendor.

Responding to Suspensions and Loops

When the system appears to be suspended or looping, try to identify the problem by working through the following steps:

1. Using messages, the `db2diag.log` file, and other information, attempt to determine why the suspension or loop occurred.

Some common problems that cause suspensions or loops can include the following:

- The operating system has run out of swap space or paging space.
- Applications are waiting on a lock, waiting for a database restart, or waiting for a response from a remote peer. To see if applications are still running, check for disk activity:
 - For UNIX-based systems, use `iostat` (see “Other Commands” on page 213)
 - For the Windows NT operating system, use its performance monitor (see “Diagnostic Tools for Windows NT” on page 229)

Another way to see if applications are still active is to use database monitor snapshots. For additional information on database monitor snapshots, see the *System Monitor Guide and Reference* manual.

2. Recover the system:
 - If the operating system is suspended (with no sign of disk activity), reboot the machine and check the `db2diag.log` file for problems. Also, check the hardware operating system reports; for example, on AIX, use `errpt -a`.
 - If you can access the operating system but not the application:
 - Check the status of applications with the Control Center or the `LIST APPLICATIONS FOR DATABASE database-alias` command. The status information shows if applications are running (UOW Executing), waiting for a lock (Lock Wait), or for user input (UOW Waiting), rather than being suspended inside of the database manager.
 - Use a CPU monitor to check for applications that are using large amounts of CPU time, and then use your judgment to determine whether or not the applications are suspended or behaving as expected.
 - Check for disk activity using an operating system command such as `iostat` on AIX.
 - Check the `db2diag.log` file for DB2 problems.
 - On UNIX-based environments, work through the following steps until you can stop your DB2 instance:

- a. Stop the DB2 instance normally with **db2stop**
- b. Stop the DB2 instance and force any remaining applications with **db2stop force**

There may be a time when you may not be able to stop the DB2 instance using the methods suggested above. Before performing any of the following steps collect as much information as possible. This may include:

- Snapshots from each of the partition servers.
- Use `ps -ef` on each partition server and save the results.
- Use `ipcs` on each partition server and save the results.
- Use `db2_call_stack` and save the results.

After carrying out the above steps, work through the following steps:

- a. Abruptly kill the DB2 instance with **db2stop -kill** In an DB2 UDB Enterprise - Extended Edition (EEE) system, you can use **db2_kill**. Use **db2kill** in a single partition server environment.
- b. Use the **kill** command to terminate any DB2 agents that cannot be stopped
- c. Use the **kill** command to terminate DB2 by killing the `db2syscs` process.
- d. As a very last resort, reboot your entire system

If you must use the **kill** command, ensure that all DB2 interprocess communications (IPC) resources are removed. Either:

- For AIX, use the **ipclean** command (in the `sql/lib/misc` directory)

Note: The **ipclean** command might not remove all DB2 interprocess communications. Use the **ipcs** command to check to see if any remain.

- Use the following command to identify those resources that belong to the DB2 instance:

```
ipcs | grep [instance_id]
```

Then use **ipcrm** to remove them.

3. If the problem persists, try to collect the following additional information before contacting DB2 Customer Support:
 - Any information logged by DB2. See “First Failure Data Capture” on page 157.
 - If the system is suspended:
 - Set up a DB2 trace to dump output to a file. Follow the instructions in “Example of Tracing to a File” on page 181.

- For UNIX-based systems, get a stack traceback for the application. Stack tracebacks provide information on the system calls for each process ID up to the point of the suspension. For more information, see “Gathering Stack Traceback Information on UNIX-Based Systems” on page 170.
 - For AIX, issue `kill -36` against the `db2sysc` process and all DB2 processes owned by the instance and the partition server.
 - For HP-UX, issue `kill -29` against the `db2sysc` and all DB2 processes owned by the instance and the partition server.
 - For Solaris Operating Environment, issue `kill -21` against the `db2sysc` and all DB2 processes owned by the instance and the partition server.
4. If you determine that the problem is not with DB2 but with a vendor-supplied application, contact the vendor.

Using All Available Information

In addition to this manual, the following resources are available to help you:

- For a complete list of DB2 information, see “Appendix C. Using the DB2 Library” on page 277.
- Online help is available:
 - For help with the DB2 graphical tools, use the **Help** menu.
 - For a description of SQLCODEs and other messages, type `db2 "? message"` from the command line, where *message* is the SQLCODE or other message number.
 - For a description of DB2 commands, type `db2 "? command"`, where *command* is the name of the command.
- For up-to-date bulletins and technical documentation see the DB2 Product and Service Technical Library at <http://www-4.ibm.com/software/data/db2/library/> (Note that this information is in English only.)
- For updated information on developing applications for DB2, refer to the Application Development with DB2 Universal Database web site at <http://www-4.ibm.com/software/data/db2/udb/ad/>
- Newsgroups and forums are provided on the Internet or from online service providers. See the “Contacting IBM” section at the end of this manual.
- There are IBM redbooks on DB2. Redbooks are “how to”, reference, and research books named for their red covers (to acquire redbooks, go to <http://www.redbooks.ibm.com/redbooks/>).

Updating DB2 Products

DB2 product updates are through *fix packs* (also known as FixPaks, patches, updates, or PTFs). Some fix packs can be applied to any level of a DB2 installation, while others require that previous fix packs are applied as a prerequisite. Each supported DB2 operating system has its own fix pack numbering scheme, and its own schedule for creating fix packs.

There is an Internet site where you can download DB2 fix packs and find out more about the fix packs that are available. Fix packs are stored in separate directories for each locale (national language).

The Internet site also provides lists of fixes that you can browse to see if there is a fix for a problem you encountered.

To access this site, either:

- Go to the DB2 Services Web page at <http://www-4.ibm.com/software/data/db2/db2tech/indexsvc.html>
- Use FTP to go to the anonymous server [ftp.software.ibm.com](ftp://ftp.software.ibm.com). Then go to the directory `/ps/products/db2/fixes`.
- On non-UNIX operating systems, use the Troubleshooting tab from within the Information Center.

Updating DB2 Products Using Fix Packs

When you have downloaded a fix pack, please ensure that you fully review the associated README file before applying the fix pack. The README file contains any special instructions regarding the fix pack. Release Notes also accompany each fix pack, and may contain corrected documentation as well as supplemental information about DB2 Universal Database and other vendor products.

Note that the README file may instruct you to rebind the applications and utilities on your DB2 server and clients. For information on binding, see the *Quick Beginnings* for the appropriate operating system.

Before you use the fix pack on all of your production systems, use the fix pack on your test environment to ensure that it installs correctly. This should be done . On the rare occurrence when a fix pack may not install correctly, it is easier to recover from a problem with your test environment than with your production systems.

Using Logged Data and Taking Traces

DB2 provides several First Failure Data Capture (FFDC) facilities that log information as errors occur, the most important being the `db2diag.log` file. The DB2 FFDC facilities are used in conjunction with CLI and DB2 traces to diagnose problems.

For information on the logging, tracing, and alert facilities available to you, see “Part 2. Advanced DB2 Troubleshooting” on page 145.

Contacting IBM for Support

Before contacting DB2 Customer Support, you should do the following:

- Attempt to solve the problem using all available information, including the resources and methods described in this guide and on the DB2 Web site at <http://www-4.ibm.com/software/data/db2/library/> (Note that this information is in English only.)
- Output from **db2level**. This allows DB2 Customer Support to quickly know which version and fix pack level of DB2 you are using.
- Collect diagnostic information (as a minimum, the `db2diag.log` file and any dump or trap files). For information on what to collect, see the relevant chapters in Part 1. For information on how to collect it, see “Part 2. Advanced DB2 Troubleshooting” on page 145.

When you can proceed no further on your own, contact DB2 Customer Support.

For more information on IBM contacts, see “Contacting IBM” on page 309.

Chapter 2. Troubleshooting the DB2 Universal Database Server

This chapter presents troubleshooting tips for the database administrator of these DB2 Universal Database products:

- IBM DB2 Universal Database Workgroup Edition
- IBM DB2 Universal Database Enterprise Edition
- IBM DB2 Universal Database Enterprise - Extended Edition.

It is beyond the scope of this book to cover all aspects of problem determination on a DB2 server. See the information in the DB2 library to find out more about operations required of DB2 administrators. For more information on administering databases, see:

- The *Quick Beginnings* guides for information on setting up and migrating databases
- The *Administration Guide* for detailed information on database administration
- The *System Monitor Guide and Reference* for information on monitoring databases
- The *Command Reference* for information on DB2 commands
- The *Administrative API Reference* for the API syntax to call utilities from a program.

The *Administration Guide* can help you with the following tasks:

- Designing the physical and logical characteristics of a database
- Day-to-day operations, such as backing up data
- Authorizing users to use the database
- Importing, exporting, and loading data
- Tuning the database manager for optimal application and operational performance.

The following list provides some basic troubleshooting rules that you should follow:

- [] Spend time designing a logical database that meets your requirements and a physical database that will be adequate for future demands. Use SMS or DMS table spaces as appropriate.
- [] Always ensure that you have a well-tested backup and recovery plan.

If you encounter serious problems, you may need to restore a backup image of your database and recover transactions through the **ROLL FORWARD** command.

- [] User-defined functions and stored procedures should be fenced, unless performance is critical and they have been thoroughly tested and debugged.
- [] To ensure optimal performance of your database, regularly run utilities such as **REORGCHK**, **REORG**, and **RUNSTATS**. (See “Improving Database Performance” on page 38.)
- [] Consult the `db2diag.log` file if you suspect problems, and remember to prune this file occasionally when it gets too big. (See “Interpreting the `db2diag.log`” on page 160.)

DB2 Administrative Tools

The following sections outline some of the tools available to help you perform database administration tasks and DB2 troubleshooting. See “Improving Database Performance” on page 38 for additional performance-related tools.

Graphical Tools

The following graphical user interface (GUI) tools are available from the DB2 folder on OS/2, and from the **Start** → **Programs** menu choice on the Windows 95, Windows 98, and Windows NT operating systems. UNIX platforms also have GUI tools. For information on these tools, see the online help for each platform and the *Administration Guide*.

Information Center

Provides a central source to access and search help, books, and up-to-date World Wide Web information.

Control Center

Displays local and remote database objects (such as databases, tables, packages, and table spaces), and lets you perform operations on them.

Use the Control Center to perform administrative tasks, such as configuring databases, managing directories, backing up and recovering databases, scheduling jobs, collecting statistics for data, and managing media.

Use graphical interfaces to perform the following tasks:

- Analyze the access plan chosen by the DB2 optimizer for SQL statements and packages using Visual Explain. To invoke Visual Explain, left-click a database object in the Control Center, and select **Explain SQL** from the pop-up menu. See also “Improving Database Performance” on page 38.

Note: With IBM DB2 Universal Database Version 7, Visual Explain is no longer invoked from the command line. It can still, however, be invoked from various database objects in the Control Center. For this version, the documentation continues to use the name Visual Explain.

- Monitor the performance of the database manager. You can capture point-in-time information at specified intervals, and send alerts if performance falls outside of a desired range. To invoke the monitoring option, left-click a database object, and select **Snapshot monitoring** from the pop-up menu.
- Record information over the duration of an event, such as a connection. To invoke this ability to record information, left-click a database object, and select **Monitor events** from the pop-up menu.

DB2 Wizards

DB2 wizards are invoked from the pop-up menus in the Control Center.

Use DB2 wizards to help you simplify some monotonous jobs:

- Create databases, table spaces, and tables
- Back up and restore databases
- Tune the performance of databases by optimizing their configuration
- Create indexes

DB2 wizards prompt you step-by-step on how to fill in the information necessary for the task you are doing. Calculations and recommendations based on the information you supply are done by the wizards.

All of the tools mentioned above apply to the DB2 Universal Database Enterprise – Extended Edition (EEE) environment. In addition, you can identify database nodes that are down, run user exits, or display alerts.

Miscellaneous Troubleshooting Tools

Standalone tools are provided for troubleshooting. You can find these tools in the `bin`, `misc`, and `adm` subdirectories of the `sql1ib` directory. In UNIX-based systems, the `sql1ib` directory is a subdirectory of the instance owner's home directory. On other operating systems, you can specify the directory where you would like the `sql1ib` directory to be placed. Typically, this is under the drive where you install DB2.

These tools provide syntax help, which you can access by typing the command followed by a question mark. (Example: `db2look ?`)

The following list highlights some of the tools available to you:

- db2bfd** Provides a bind file description. For more information on binding, see the *Application Development Guide*, the *Command Reference*, or the *Administrative API Reference*.
- db2cat** Dumps the contents of packed descriptors for tables and formats them in a readable form.
- Note:** A packed descriptor is a column within the system catalog tables that DB2 uses to identify the details of a database object.
- db2dart** Verifies that the architectural integrity of a database is correct. For example, this tool confirms that:
- The control information is correct
 - There are no discrepancies in the format of the data
 - The data pages are the correct size and contain the correct column types
 - Indexes are valid
- Note:** If **db2dart** reports a problem with an index, use the tool's `/MI` option to mark the index as invalid. The index is rebuilt based on the value of the *indexrec* database and database manager configuration parameters (see the *Administration Guide: Performance* for details).
- You must run this tool on the DB2 server where the database resides. You must also ensure that there are no active connections to the database. (Use the **LIST APPLICATIONS FOR DATABASE** *database-alias* command and disconnect any applications that are listed.)
- In a DB2 UDB Enterprise - Extended Edition (EEE) environment, **db2dart** must be run on each database partition server.
- For information on **db2dart** options, type `db2dart` without any options.
- db2flsn** Returns the name of the file that contains the log record identified by a specified log sequence number (LSN).
- The log header control file `sqlogctl.lfh` must reside in the current directory before using this tool. The tool also uses the *logfilsiz* database configuration parameter. This tool can only be used with recoverable databases.

db2ipxad	Returns the DB2 server's IPX/SPX internetwork address. This command <i>must</i> be issued locally from the DB2 server machine. Issuing the command from a remote client is not supported.
db2level	Displays detailed output about the level of DB2 Universal Database code, including the fix pack level, that is currently installed.
db2look	<p>Extracts the DDL necessary to re-create a database or database objects. This tool allows you to, for example, mimic a production database on your test database. Use the <code>-m</code> option against a production database to generate the update statements that will match the catalog statistics of a test database with those of the production database. Use the <code>-e</code> option to generate the DDL for one or more tables from the database catalogs.</p> <p>For introductory information on db2look, type <code>db2look</code>. For detailed information, type <code>db2look -h</code>.</p> <p>For more information on the options available on <code>db2look</code>, see the <i>Command Reference</i>.</p>
db2recr	Re-creates indexes that were marked as inconsistent during a database restart or use of the db2dart tool.
db2sql92	Reads SQL statements from either a flat file or standard input, dynamically describes and prepares the statements, and returns an answer set. Supports concurrent connections to multiple databases. Several parameters are associated with this command. See the <i>Command Reference</i> manual for additional information.
db2tbst	Provides a text description for a table space state.
db2untag	<p>Removes the DB2 tag from a table space container. The tag is used to prevent DB2 from using a container for more than one table space. If a table space or database is destroyed, this tag may be left behind, preventing future DB2 use of the resource.</p> <p>Use this tool if a <code>DROP TABLESPACE</code> command does not work. Typically, the <code>SQL0294N</code> message is received.</p> <p>Attention: Use this tool in consultation with DB2 Customer Support, and only if you are an experienced database administrator. You must be completely sure that the container is not used by any other database, because this command is equivalent to dropping the container. Running this command on a container that has data results in the loss of all the data from that container.</p>

The Database System Monitor

The DB2 database manager maintains data about its operation and performance as it runs. This data can provide important troubleshooting information. For example, you can find out:

- The number of applications connected to a database, their status, and which SQL statements each application is executing, if any.
- Counts that show how well the database manager is configured.
- When deadlocks occurred for a specified database, which applications were involved, and which locks were in contention.
- The list of locks held by an application or a database. If the application cannot proceed because it is waiting for a lock, there is additional information on the lock, including which application is holding it.

Because collecting some of this data introduces overhead on the operation of DB2, *monitor switches* are available to control which information is collected. To set monitor switches explicitly, use the **UPDATE MONITOR SWITCHES** command or the **sqlmonon()** API. (You must have SYSADM, SYSCTRL, or SYSMANT authority.)

There are two ways to access the data maintained by the database manager:

1. *Taking a snapshot.* Use the **GET SNAPSHOT** command from the command line, or the Control Center on the OS/2, Windows 95, Windows 98, and Windows NT operating systems. You can also write your own application, using the **sqlmonss()** API call.
2. *Using an event monitor.* The event monitor captures system monitor information after particular events have occurred, such as the end of a transaction, the end of a statement, or the detection of a deadlock. This information can be written to files or to a named pipe.

To use an event monitor:

1. Define it with the Control Center or the SQL statement **CREATE EVENT MONITOR**. This statement stores the definition in the database system catalog.
2. Activate the event monitor with the Control Center or the SQL statement:
`SET EVENT MONITOR evname STATE 1`

If you are writing to a named pipe, start the application reading from the named pipe before activating the event monitor. You can either write your own application to do this, or use **db2evmon**. When the event monitor is active and starts writing events to the pipe, **db2evmon** will read them as they are being generated and write them to standard output.

3. If using a file event monitor, you can view the binary trace that it created in either of the following ways:
 - Use the **db2evmon** tool to format the trace to standard output.

- Click the **Event Analyzer** icon in the Control Center to view the trace, search for keywords, and filter out unwanted data.

For information on the system database monitor and the event monitor, see the *System Monitor Guide and Reference*. For a scenario of how to use them from the Control Center, see the *Administration Guide: Implementation*.

Troubleshooting Tips

This section describes how to deal with some problems frequently encountered by users working with a DB2 server. The following topics are presented in this section:

- “Problems Migrating to Version 7”
- “Server Failures” on page 20
- “SQL1403N Received When Authenticating Usernames or Groups” on page 27
- “Administration Difficulties” on page 27
- “Backing Up and Restoring Data” on page 28
- “Difficulties Importing Data” on page 34
- “Problems Using Commands, Utilities, or the Command Line Processor” on page 35
- “Locking and Deadlocks” on page 36
- “Improving Database Performance” on page 38

For additional troubleshooting information on this topic, see:

- “Troubleshooting Tips for Host Connections to DB2 Universal Database” on page 67 for information on connecting to DB2 servers from host systems
- “Chapter 7. Troubleshooting DB2 Universal Database Enterprise - Extended Edition” on page 87 for information on administering DB2 Enterprise - Extended Edition systems

Important: This section represents a small sampling of the information available from DB2 Customer Support. For a complete and up-to-date source of DB2 information, use the DB2 Product and Service Technical Library on the Web, at <http://www-4.ibm.com/software/data/db2/library/> (Note that this information is in English only.)

Problems Migrating to Version 7

If you experience problems migrating to Version 7, ensure that:

- [] You followed the instructions in the *Quick Beginnings* guide.
- [] You checked the list of incompatibilities from previous versions. (See the *Administration Guide: Planning*.)
- [] The database is consistent.

To verify that databases are consistent so that they can be migrated, use the **db2ckmig** command. (See the *Quick Beginnings* book for more details.)

You must correct all errors that are reported for databases that are in any of the following states:

- Backup pending: Perform a backup of the database.
- Rollforward pending: Recover the database as required by performing or resuming the **ROLLFORWARD DATABASE** command.
- Database inconsistent: Restart the database to make it consistent.

Additionally, this tool identifies databases that contain objects that use SYSCAT, SYSSTAT, or SYSFUN as schema names. These objects must be dropped and re-created using a different schema name.

- [] The disk space on the machine is adequate. (The amount of disk space required varies with each database.)
- [] You use appropriate values for the number of primary and secondary log files (*logprimary* and *logsecond* in database configuration), and for the size of the log files (*logfilsiz* in database configuration). See “Server Failures” for details.

Problems When Applying Fix Packs

Symptom

The error message SQL1042C is received following the installation of a fix pack or a change in the operating system level.

Possible Cause

A fix pack or a change in the operating system level needs to be identified to the DB2 instance (database manager).

Action

Run **db2iupt** following the installation of a fix pack for DB2 or when there is a change in the operating system level.

Server Failures

Symptom

The DB2 server does not respond to requests or provides unexpected responses.

Action

Ensure that:

- [] You check the `db2diag.log` file on the server to diagnose a cause. For information, see “Interpreting the `db2diag.log`” on page 160.

- [] The DB2COMM registry variable is set to the protocols required by connecting clients. (If not, the message SQL5043N is usually received.)

Consult the `db2diag.log` file to see which communication protocols were not started successfully, and why. For information, see “Using the `db2diag.log` File to Diagnose Server Communication Problems” on page 47.

- [] There is enough log space. (If not, the SQL0964C message is typically received.)

To increase log space:

- Increase the number of primary or secondary log files (*logprimary* and *logsecond* in database configuration)
- Increase the size of the log files (*logfilsiz* in database configuration)

To initiate your changes, you must disconnect all applications from the database and deactivate the database, if it was activated.

You must ensure that the file system or disk space that contains the logs is big enough to hold all of the logs. The amount of space (in bytes) required for log files can range from:

$$(\text{logprimary} * (\text{logfilsiz} + 2) * 4096) + 8192$$

to:

$$((\text{logprimary} + \text{logsecond}) * (\text{logfilsiz} + 2) * 4096) + 8192$$

- [] The indexes are valid.

The `db2diag.log` file indicates if indexes need to be re-created. Use **db2dart /MI** to mark the indexes as inconsistent. See “Miscellaneous Troubleshooting Tools” on page 15 for details on **db2dart /MI**.

To delay the rebuilding of indexes until after the database is started, use the INDEXREC configuration parameter as documented in the *Administration Guide: Performance* manual.

- [] The data is consistent.

Look for the following symptoms for inconsistent data:

- There is a DIA3700C “A bad page was encountered” error in the `db2diag.log` file.
- The server goes down when particular data is accessed. In this case, this data resource may be damaged.

- The server fails repeatedly.

Run **db2dart** to verify that the data is consistent. If data is not consistent, contact DB2 Customer Support.

- [] You are using the latest version of DB2, with appropriate fix packs installed. Check the Web site for known problems and the fix packs that fix them. For information, see “Updating DB2 Products” on page 10.

For DB2 Universal Database Enterprise - Extended Edition for supported operating systems, see also “Database Manager Will Not Start” on page 88.

Starting DB2 on Windows 95 and Windows 98 When the User Is Not Logged On

For a **db2start** command to be successful in a Windows 95 or a Windows 98 environment, you must either:

- Log on using the Windows logon window or the Microsoft Networking logon window
- Issue the **db2logon** command (see note 1 on page 24 for information about the **db2logon** command).

In addition, the user ID that is specified either during the logon or for the **db2logon** command must meet DB2’s requirements (see note 2 on page 24).

When the **db2start** command starts, it first checks to see if a user is logged on. If a user is logged on, the **db2start** command uses that user’s ID. If a user is not logged on, the **db2start** command checks whether a **db2logon** command has been run, and, if so, the **db2start** command uses the user ID that was specified for the **db2logon** command. If the **db2start** command cannot find a valid user ID, the command terminates.

During the installation of DB2 Universal Database Version 6 on Windows 95 and Windows 98, the installation software, by default, adds a shortcut to the Startup folder that runs the **db2start** command when the system is booted (see note 1 on page 24 for more information). If the user of the system has neither logged on nor issued the **db2logon** command, the **db2start** command will terminate.

If you or your users do not normally log on to Windows or to a network, you can hide the requirement to issue the **db2logon** command before a **db2start** command by running commands from a batch file as follows:

1. Create a batch file that issues the **db2logon** command followed by the **db2start.exe** command. For example:

```
@echo off
db2logon db2local /p:password
db2start
cls
exit
```

2. Name the batch file `db2start.bat`, and store it in the `/bin` directory that is under the drive and path where you installed DB2. You store the batch file in this location to ensure that the operating system can find the path to the batch file.

The drive and path where DB2 is installed is stored in the DB2 registry variable `DB2PATH`. To find the drive and path where you installed DB2, issue the following command:

```
db2set -g db2path
```

Assume that the `db2set` command returns the value `c:\sql1ib`. In this situation, you would store the batch file as follows:

```
c:\sql1ib\bin\db2start.bat
```

3. To start DB2 when the system is booted, you should run the batch file from a shortcut in the Startup folder. You have two options:
 - Modify the shortcut that is created by the DB2 installation program to run the batch file instead of `db2start.exe`. In the preceding example, the shortcut would now run the `db2start.bat` batch file. The shortcut that is created by DB2 installation program is called **DB2 - DB2.lnk**, and is located in `c:\WINDOWS\Start Menu\Programs\Start\DB2 - DB2.lnk` on most systems.
 - Add your own shortcut to run the batch file, and delete the shortcut that is added by the DB2 installation program. Use the following command to delete the DB2 shortcut:

```
del "C:\WINDOWS\Start Menu\Programs\Startup\DB2 - DB2.lnk"
```

If you decide to use your own shortcut, you should set the *close on exit* attribute for the shortcut. If you do not set this attribute, the DOS command prompt is left in the task bar even after the `db2start` command has successfully completed. To prevent the DOS window from being opened during the `db2start` process, you can create this shortcut (and the DOS window it runs in) set to run minimized.

Note: As an alternative to starting DB2 during the boot of the system, DB2 can be started prior to the running of any application that uses DB2. See note 5 on page 25 for details.

If you use a batch file to issue the `db2logon` command before the `db2start` command is run, and your users occasionally log on, the `db2start` command will continue to work, the only difference being that DB2 will use the user ID of the logged on user. See note 1 on page 24 for additional details.

Notes:

1. The **db2logon** command simulates a user logon. The format of the **db2logon** command is:

```
db2logon userid /p:password
```

The user ID that is specified for the command must meet the DB2 naming requirements (see note 2 for more information). If the command is issued without a user ID and password, a window opens to prompt the user for the user ID and password. If the only parameter provided is a user ID, the user is not prompted for a password; under certain conditions a password is required, as described below.

The user ID and password values that are set by the **db2logon** command are only used if the user did not log on using either the Windows logon window or the Microsoft Networking logon window. If the user has logged on, and a **db2logon** command has been issued, the user ID from the **db2logon** command is used for all DB2 actions, but the password specified on the **db2logon** command is ignored

When the user has not logged on using the Windows logon window or the Microsoft Networking logon window, the user ID and password that are provided through the **db2logon** command are used as follows:

- The **db2start** command uses the user ID when it starts, and does not require a password.
- In the absence of a high-level qualifier for actions like creating a table, the user ID is used as the high-level qualifier. For example:
 - a. If you issue the following: `db2logon db2local`
 - b. Then issue the following: `create table tab1`
The table is created with a high-level qualifier as `db2local.tab1`.

You should use a user ID that is equal to the schema name of your tables and other objects.

- When the system acts as client to a server, and the user issues a **CONNECT** statement without a user ID and password (for example, **CONNECT TO TEST**) and authentication is set to server, the user ID and password from the **db2logon** command are used to validate the user at the remote server. If the user connects with an explicit user ID and password (for example, **CONNECT TO TEST USER *userID* USING *password***), the values that are specified for the **CONNECT** statement are used.
2. In Version 6, the user ID that is either used to log on or specified for the **db2logon** command must conform to the following DB2 requirements:
 - It can be a maximum of 8 characters (bytes) in length.

- It cannot be any of the following: USERS, ADMINS, GUESTS, PUBLIC, LOCAL, or any SQL reserved word that is listed in the *SQL Reference*.
 - It cannot begin with: SQL, SYS or IBM
 - Characters can include:
 - A through Z (Windows 95 and Windows 98 support case-sensitive user IDs)
 - 0 through 9
 - @, #, or \$
3. You can prevent the creation of the **db2start** shortcut in the Startup folder during a customized interactive installation, or if you are performing a response file installation and specify the DB2.AUTOSTART=NO option. If you use these options, there is no db2start shortcut in the Startup folder, and you must add your own shortcut to run the db2start.bat file.
 4. On **Windows 98**, an option is available that you can use to specify a user ID that is always logged on when Windows 98 is started. In this situation, the Windows logon window will not appear. If you use this option, a user is logged on and the **db2start** command will succeed if the user ID meets DB2 requirements (see note 2 on page 24 for details). If you do not use this option, the user will always be presented with a logon window. If the user cancels out of this window without logging on, the **db2start** command will fail unless the **db2logon** command was previously issued, or invoked from the batch file, as described above.
 5. If you do not start DB2 during a system boot, DB2 can be started by an application. You can run the db2start.bat file as part of the initialization of applications that use DB2. Using this method, DB2 will only be started when the application that will use it is started. When the user exits the application, a **db2stop** command can be issued to stop DB2. Your business applications can start DB2 in this way, if DB2 is not started during the system boot.

To use the DB2 Synchronizer application or call the synchronization APIs from your application, DB2 must be started if the scripts that are download for execution contain commands that operate either against a local instance or a local database. These commands can be in database scripts, instance scripts, or embedded in operating system (OS) scripts. If an OS script does not contain Command Line Processor commands or DB2 APIs that use an instance or a database, it can be run without DB2 being started. Because it may be difficult to tell in advance what commands will be run from your scripts during the synchronization process, DB2 should normally be started before synchronization begins.

If you are calling either the **db2sync** command or the synchronization APIs from your application, you would start DB2 during the initialization of your application. If your users will be using the DB2 Synchronizer shortcut in the DB2 for Windows folder to start synchronization, the DB2

Synchronization shortcut must be modified to run a db2sync.bat file. The batch file should contain the following commands to ensure that DB2 is running before synchronization begins:

```
@echo off
db2start.bat
db2sync.exe
db2stop.exe
cls
exit
```

In this example, it is assumed that the db2start.bat file invokes the **db2logon** and **db2start** commands as described above.

If you decide to start DB2 when the application starts, ensure that the installation of DB2 does not add a shortcut to the Startup folder to start DB2. See note 3 on page 25 for details.

Using a Windows 2000 Terminal Server environment

Symptom

You receive the error message SQL1032N when attempting use applications such as REXX or ODBC Driver Managers which dynamically load DB2 DLLs.

Possible Cause

The execution mode that was set by the **REGISTER** command has no effect. The error message is returned because the application could not attach to the shared memory needed to run the application.

Action

There are actually two problems::

- A problem with Windows 2000 Terminal Server when accessing DB2 GLOBAL Kernel objects. Contact IBM DB2 Service and Support to discuss what should be done. There is a fix from Microsoft.
- A problem when local applications fail using the Remote Administration mode of Windows 2000 Terminal Server. All required binary files must be marked with GLOBAL execution mode by the **REGISTER** command when working in Windows NT. Here are is how to register the REXX binary files to run as SYSTEM GLOBAL:

1. Ensure that you do not have any REXX applications running.
2. Start the task manager window and remove any REXX processes. For example, ensure that rxapi.exe is not running.
3. Go to the \winnt\system32 directory.
4. Register the 3 REXX files: rxapi.exe, rexx.dll, and rexxapi.dll as SYSTEM GLOBAL. For example:

```
register /SYSTEM rxapi.exe
```


SQL1403N Received When Authenticating Usernames or Groups

Symptom

You receive the SQL1403N error message when attempting to get authorization for usernames or groups on a Windows NT Primary Domain running DB2.

Possible Cause

The authorization restrictions on the Windows NT Primary Domain.

Action

Ensure the service for the instance is being started under a local administrator who is not a user from the master domain; and who has operating system rights on the local machine.

Administration Difficulties

Symptom

You have problems installing or administering a database server.

Action

Ensure that:

- [] You are using a valid user name and password that have administrator authority.
 - For OS/2 Warp Connect and later versions of OS/2: The default DB2 administrator user name and password are the same as those specified when the operating system was installed.
 - For versions of OS/2 before OS/2 Warp Connect: The default administrator user name and password depend on the national language installed on a machine, and on whether UPM was installed with a particular user name. In most cases, the defaults are USERID and PASSWORD in English countries. For more information, see the *Quick Beginnings* for your platform.
 - For Windows NT: Your membership in a group controls which actions you are permitted to perform. For DB2 for Windows NT, you must belong to the "Administrators" or "Domain Admins." groups to perform installation and some administration tasks.

You can find out what your group is by using the User Manager tool to display user names and group memberships. To invoke this tool, select **Start** → **Programs** → **Administrative Tools** → **User Manager**.

To install DB2, you must have administrator authority on the local machine. To be added to the administrator group, you must be added by an existing member of that group.

- For UNIX-based environments: You must belong to a SYSADM group. The name of the group must be eight characters or less in length.
- For DB2 for Solaris: If a DB2 instance is started by a user who does not belong to the same group as the DB2 instance owner, the SQL1042C message is received for many commands. In this situation, the executing db2sysc process inherits the user's group name and does not have the correct read permission to open a file in the /proc directory. You must start the instance as the instance owner, and the id or group of the db2sysc process attempting to read a file must be the same as the id or group of the DB2 instance.

[] For UNIX-based environments other than AIX, you must update and rebuild the kernel configuration parameters. (If not, the SQL1016N or SQL1018N messages typically occur when you try to create an instance or use the command line processor.) See the *Quick Beginnings* guide for your platform for details.

Backing Up and Restoring Data

It is important to have a good backup and recovery strategy, and a plan to implement this strategy. This plan should address questions such as the following:

- How crucial is the data?
- How long can the data be unavailable without impacting users?
- How much time and resource can be spent on restoring the database?
- Is the data read-only, or is it updated?
- Can the data be rebuilt easily from another source?
- How much resource is allocated to a backup and recovery system?

Backup plans are heavily influenced by the importance of having the data in the database available for use. If your business depends heavily on the data, then you must develop a backup and recovery plan that will minimize downtime and ensure that the data is available when needed at a cost that you are willing to accept.

For help on backup and recovery see the *Data Movement Utilities Guide and Reference* and use the DB2 Backup Database Wizard from the Control Center.

SQL0902C Received When Backing up a Database or Accessing DB2 Data

Symptom

The SQL0902C message is received when a database is backed up.

Possible Cause

The database is not architecturally consistent.

Action

Use the **db2dart** command to verify the architectural consistency of the database. See “Miscellaneous Troubleshooting Tools” on page 15 for information on this command.

Usually you can perform a restore with a roll forward to the end of the logs to correct the problem. If the problem persists, contact DB2 Customer Support.

ROLLFORWARD TABLESPACE Fails with SQL4908N Message

Symptom

The error message SQL4908N is received when running a ROLLFORWARD TABLESPACE.

Possible Cause

If starting the first attempt to roll forward a table space, the message may be received when one or more of the table spaces specified to be rolled forward is not in a ROLLFORWARD PENDING state on the specified nodes. If you are continuing to roll forward a table space that is already in progress, the message may be received when one or more of the table spaces specified to be rolled forward is not in a “roll forward in progress” state on the specified nodes.

Action

1. Use the **LIST TABLESPACES SHOW DETAIL** command on the nodes specified to find out which table spaces are not ready to be rolled forward.
2. Use the **QUERY STATUS** option of the **ROLLFORWARD** command to determine the status of the table space roll forward.
3. Depending on your task:
 - If you are starting a new table space roll forward, put the table spaces into ROLLFORWARD PENDING state by restoring them.
 - If you are continuing a table space rollforward and one or more of the table spaces involved has been restored and put into ROLLFORWARD PENDING state, the table space roll forward in progress must be cancelled.
 - Put the table spaces into the RESTORE PENDING state.
 - Restore the table spaces.
4. Submit the original **ROLLFORWARD** command again.

Backup to Disk Fails with SQL2419N

Symptom

The SQL2419N error message is received when taking an online backup to disk.

Possible Cause

Although the disk was not full, the backup file was greater than the operating system file size limit.

Action

You can do a backup and specify multiple target directories. The backup directories divide the file into sizes smaller than the 2 GB operating system file size limit.

NetBIOS Connections from DB2 Clients are Lost When Backing up Using TSM

Symptom

NetBIOS connections from DB2 clients are lost when backing up using Tivoli Storage Manager (TSM).

Possible Cause

As part of TSM's initialization, a **NCB.RESET** is issued, which resets the NetBIOS resources of the calling application. In the case of DB2 for OS/2 and Windows NT, the calling application is DB2 itself. This results in the dropping of all connections to the database made through NetBIOS.

Action

There is no way to prevent this from happening when using NetBIOS as the communication protocol. Use another protocol like TCP/IP when using TSM to backup the database.

Problems Restoring a Database or Table Space Backup

Symptom

Data cannot be restored.

Action

Ensure that:

[] You have enough disk space to restore the data.

Use redirected restore to specify a modified list of containers for the table spaces being restored. See the *Administration Guide: Implementation*.

[] You specify the correct path to the backup image and logs. (It is possible that they were moved.)

- [] If an online backup was taken, all logs from the beginning to the end of the backup are required to restore and roll forward the database.

(The roll forward must proceed to the minimum point in time as specified by the output from the **ROLLFORWARD QUERY STATUS** command. Otherwise, the database is not accessible.)

RESTORE Using an API Fails with SQL0902C Return Code 22

Symptom

On AIX, the SQL0902C error message with return code 22 is received when restoring a database using an API.

Possible Cause

Abnormal termination of applications has left the message queue connected to an existing database file.

Action

Stop the database by using **db2stop**. Use **db2terminate** to eliminate all back-end processes. Clean up DB2 interprocess communications (IPC) resources. Start the database by using **db2start**. Retry the **RESTORE** again. Contact DB2 Support if the problem persists.

After a RESTORE, Some Table Spaces Are in a ROLLFORWARD PENDING State

Symptom

After issuing a **RESTORE** command specifying that no roll forward is required on a Windows NT machine, the table spaces are left in a ROLLFORWARD PENDING state.

Possible Cause

Windows NT is having a resource conflict with DB2 where the hard file cache limit of 192 MB is reached. This hard file cache is used whenever DB2 opens or closes many files, or opens or closes large files.

Action

Ensure the DB2 product is updated to the latest fix pack, and that the DB2NTNOCACHE registry variable is set to one. The DB2NTNOCACHE setting can be checked using the **db2set -all** command.

SQL1277N is Received When Attempting a Redirected Restore

Symptom

The SQL1277N error message is received during a redirected restore to a new database.

Possible Cause

One or more containers used in the redirected restore may already be in use. Check the db2diag.log file for additional information.

Action

Remove or untag the containers in use and then perform the redirected restore using the Control Center. Remove SMS containers. Untag DMS containers. (SMS may be untagged.) If you prefer to use the command line processor, you must use the **SET TABLESPACE CONTAINERS** API and then reissue the **RESTORE** command with the *CONTINUE* parameter.

To find out more about the **SET TABLESPACE CONTAINERS** API, type `db2 ? set tablespace` at the command line.

SQL1024N is Received when Attempting a Redirected Restore**Symptom**

The SQL1024N error message is received during a redirected restore.

Possible Cause

A redirected restore has three steps:

1. Restore redirect
2. Set table space containers
3. Restore continue.

After the first step, an implicit connection to the database is established. The connection associated with a session identifier. If a shell script is used for the remaining two steps, a new session is started. There is no implicit connection to the database and as a result the second step fails.

Action

When using shell scripts, ensure that all three steps are in one script.

SQL0298N is Received when Setting Table Space Containers During a Redirected Restore**Symptom**

The SQL0298N error message is received when setting table space containers during a redirected restore.

Possible Cause

The SQL0298N error message is received when the customer was attempting to restore a DMS table space into a SMS table space using a redirected restore.

Action

This is not a valid action. The table space type cannot be changed.

Problem with Recovery of Large Databases on Windows NT and Windows 2000

Symptom

BACKUP or **RESTORE** fails on Windows NT or Windows 2000.

Possible Cause

NTFS file caching runs out of space when the backup image file is larger than 128GB.

Action

Reduce the number of logical drives being used during the **BACKUP** or **RESTORE**.

Data Not Loading

SQL3508N Returned during a LOAD Operation

Symptom

The SQL3508N error message is received during a **LOAD** operation.

Possible Causes

1. Issuing the **LOAD** command without explicitly specifying the *REMOTE FILE* parameter before attempting a **LOAD RESTART** operation causes the utility to overwrite the default remote file.
2. The specified file names used are similar even though the specified paths are different. For example: *MESSAGES C:\table.MSG* and *REMOTE FILE D:\table*.

Action

1. Restore the database or restore the table space.
2. Specify different file names for each of the *MESSAGES* and *REMOTE FILE* parameters. For example: *MESSAGES C:\table.MSG* and *REMOTE FILE D:\table.RMT*. The RMT extension solves the problem.

General Problems with LOAD

Symptom

Data does not load on a database server.

Action

Ensure that:

- [] You used the *RESTART* or *REPLACE* parameters with the **LOAD** command. (If not, the SQL3805N message is typically received.)

For more information on the **LOAD** command, see the *Data Movement Utilities Guide and Reference*.

- [] The size of a database object in an SMS table space has not reached an operating system limit.

For example, OS/2 and Windows NT have a 2 GB file size limit.

(See “Problems with LOAD and with Splitting and Loading Data” on page 93 for more information on loading data in DB2 Universal Database Enterprise – Extended Edition.)

Difficulties Importing Data

Symptom

You are having problems importing data.

Action

- [] Ensure that there is enough log space available. (See “Server Failures” on page 20.)
- [] Use the COMMITCOUNT *n* option to commit data after every *n* records are imported. This option protects the committed data from loss in case of a failure. It also reduces the log requirements for an import operation, allowing a number of smaller transactions rather than one long transaction.

DB2 Support for Raw Devices when Using Windows 2000

Symptom

Referencing raw devices when using Windows 2000 does not appear to work.

Possible Cause

Windows 2000 has a new Dynamic Disk mode that fails when there are too many partitions and you are using raw devices. The Windows 2000 Logical Disk Manager handles the definition of raw devices differently from the method used on Windows 2000.

Action

Define raw devices using the method associated with the Basic Disk mode and using the type of long partition names used on a single partition.

Disk Alignment when Using Windows 2000

Symptom

Performance problems with different DB2 tasks and utilities that involve input and output to disk.

Possible Cause

Windows 2000 has changed the default values for *SectorsPerTrack* and *TracksPerCylinder*. For example, *SectorsPerTrack* is increased from 32 to

63. The problems occur when there are many disks and RAID controllers provide optimizations for track-aligned transfers.

Action

There are several ways to correct this problem. The easiest to do is:

- Use a private `classnp.sys` driver to revert to the Windows NT behavior.
- Create disks and partitions under Windows 2000 using this private driver.
- Replace the private driver with the one that was shipped with Windows 2000, and reboot your machine.

Note: There is no disk alignment problem if you create your partitions using Windows NT and then upgrade to Windows 2000 (or use a dual-boot with Windows 2000).

Problems Using Commands, Utilities, or the Command Line Processor

Symptom

You cannot use commands, utilities, or the Command Line Processor.

Action

Ensure that:

- [] You bound the utilities and your applications to the database after you installed a fix or newer executable. (The SQL0818N or SQL0805N messages indicate that you need to rebind.)

When you precompile an SQL application, a compilable file is produced, and optionally a bind file. Both of these files have a new timestamp. The default behavior of the precompile operation is to automatically create a package so that you do not need to bind anything. However, if you create a bind file but not a package, then you must bind the new bind file to the database.

For information on binding, see your *Quick Beginnings* guide.

- [] You used the correct syntax (especially for UNIX-based systems).

It is recommended that you either:

- Use double quotation marks (" ") around a command line processor request.
- Precede any special characters such as asterisks (*), brackets, or question marks (?) with a back slash (\) character to ensure that the command line processor interprets them correctly.

Example

For:

```
db2 SELECT * FROM SYSCAT.TABLES
```

Use either:

```
db2 "SELECT * FROM SYSCAT.TABLES"
```

or

```
db2 SELECT \* FROM SYSCAT.TABLES
```

Locking and Deadlocks

The database manager provides concurrency control and prevents uncontrolled access by means of locks. Because of the basic principles of DB2 locking, you do not need to take action to control locks in most cases. For information on how locking works, see the *Administration Guide*.

Diagnosing locking problems

Use the following methods to investigate whether or not applications are creating deadlocks or holding locks:

- *Deadlocks* occur when two or more applications connected to the same database wait indefinitely for a resource. The waiting is never resolved because each application is holding a resource that the other needs to continue.

Use an event monitor for deadlocks on a database to keep a log of each deadlock as it occurs.

Sometimes deadlocks are unavoidable. Your application should plan for such an occurrence by handling SQLCODE -911. In this way you can detect a deadlock and plan to do something about it, but you may not be able to avoid the occurrence of a deadlock.

- To determine if an application is **waiting for a lock**, use the snapshot monitor and see if *lock_wait_time* is high. A high value could indicate that an application is waiting for a lock that another application is holding. It indicates that applications may not be committing their transactions frequently enough.

To see which locks are causing waits, and which applications are holding them, set the LOCK monitor switch ON and then get an application snapshot.

For information, see “The Database System Monitor” on page 18 and the *System Monitor Guide and Reference*.

SQL0911N Received

Symptom

The SQL0911N message is received.

Possible Cause

There are deadlocks or time-outs.

Action

Check the reason code in the SQLCA to determine if deadlocks or time-outs are causing the problem (see “Interpreting an SQLCA Structure” on page 162).

In the case of deadlocks, ensure your applications are coded in such a way that they can handle -911 return codes. Possibly increase the size of the lock list to avoid lock escalation which can often result in a deadlock scenario.

Note that time-outs are not captured using the deadlock event monitor.

Applications Slow or Appear to Hang

Symptom

Applications accessing a DB2 database are slow or appear to hang.

Possible Cause

There is lock contention or lock escalation.

Action

Use the Control Center or the **LIST APPLICATIONS FOR DATABASE *database-alias* SHOW DETAIL** command to find out if applications are waiting on a lock and who is holding the lock that the applications are waiting for. Then use the database monitor to determine if lock escalation is occurring.

If escalation is occurring, ensure that:

- [] Applications use an appropriate frequency of commits.
Take a monitor snapshot for locks to determine which applications are causing other applications to wait for locks. Also, monitor the *db2.lock_waits* performance variable.
- [] The values of the database configuration parameters *maxlocks* and *locklist*, which determine the number of locks allowed, are appropriate.
- [] The *locktimeout* database configuration parameter is set appropriately. When it is turned on, this parameter can help avoid hangs due to lock contention, and can tell you if lock contention is your problem.
- [] The *dlchktme* database configuration parameter is set appropriately. This parameter defines the frequency at which the database manager checks for deadlocks among all the

applications connected to a database. If it is set high, CPU time is saved but deadlocks may not be detected soon enough.

- [] Other users are not preparing and binding applications.

The procedure of preparing and binding applications requires that locks be obtained on system catalog tables. This procedure should be performed in off-peak hours.

- [] Lock waits or deadlocks are not caused by next key locking.

Next key locking guarantees a cursor stability (CS) isolation level by automatically locking the next key for all INSERT and DELETE statements and the next higher key value above the result set for SELECT statements. It is required to guarantee ANSI and SQL92 standard CS, and is the DB2 default.

Examine snapshot information for the application. If the problem appears to be with next key locking, you can change the isolation level to Read Stability (RS) by setting the DB2_RR_TO_RS option on.

The DB2_RR_TO_RS option stops *all* next key locking on user tables (catalog tables are not affected). Any packages bound with CS are automatically downgraded to RS, because DB2 can no longer guarantee CS. Do not use this option if you require ANSI and SQL92 standard CS.

For information on locking, configuration parameters, and explain information, see the *Administration Guide: Performance*.

Improving Database Performance

Database performance is a complex subject and its details extend beyond the scope of this guide. As a starting point, be aware of the following:

- The Performance Configuration DB2 wizard provides a first attempt at optimizing a database. It asks you qualitative questions about the database, its data, and its purpose, and then suggests optimal configuration parameters. To invoke this DB2 wizard, right-click on the **Databases** folder in the object tree in the Control Center, and then select **Configure performance** from the pop-up menu.
- The **REORGCHK** command returns information about the physical characteristics of a table, and whether or not it would be beneficial to reorganize that table.
- If you feel your applications are running slowly, you should ensure that the database statistics are up-to-date. The **RUNSTATS** command updates catalog statistics, thereby improving the potential for good query

performance. It is accessible from the command line or from the Control Center. Rebind all static applications after running it.

- The Index DB2 wizard (or SQL Advise facility) can be used with a specific query to create an index that may improve performance. See the *Administration Guide: Performance* manual for details on how to use and how to access the SQL Advise Facility.
- To understand the access plan chosen for a particular SQL statement, use Visual Explain from the Control Center. This tool accesses the integrated Explain facility, a component of the SQL compiler that captures information about the environment in which static and dynamic SQL statements are compiled.

This information can help you:

- Understand the way in which a query executes
- Design application programs and databases
- Determine when an application should be rebound

Visual Explain provides an accurate, comprehensive, and easy-to-understand profile of access plans.

- Other tools that provide information on the access plan include:
 - **db2expln** describes the access plan, selected for static SQL statements in the packages that are stored in the system catalog tables. It can render a quick explanation of an access plan for packages in which Explain data was not captured at bind time.
 - **dynexpln** converts a dynamic SQL statement into a static SQL statement and interprets it through **db2expln**. In other words, it creates a static package for the statements and in turn employs the **db2expln** tool to provide a description of those statements. Although this method is quick to interpret basic access plans for dynamic statements, the results may be misleading.
 - **db2exfmt** provides a text description of the explain tables and the operators chosen for access plans of static SQL.

You can use these Explain tools to understand the access plan chosen for a particular SQL statement. Or, you can use the integrated Explain facility in conjunction with Visual Explain to understand the access plan chosen for a particular SQL statement. Both dynamic and static SQL statements can be explained using the Explain facility. Visual Explain results are provided in a graphical format which is different from the other Explain tool results. Besides this, however, the level of detail provided in either tool is identical.

To fully use the output of **db2expln** and **dynexpln** you must understand the different SQL statements supported and the terminology related to those statements (such as predicates contained in a SELECT statement).

For information on the Explain facility and on **db2expln**, see the *Administration Guide: Performance*.

- Configuration parameters affect performance. For example, *sheapthres*, the database manager configuration parameter for sort heap threshold, can affect performance if it is set too low (see “Improving Database Performance” on page 38). It should also be at least twice as big as *sortheap*, the database configuration parameter for sort heap. For information, refer to the *Administration Guide: Performance*.
- Locking and how it is handled may affect performance of applications. See “Applications Slow or Appear to Hang” on page 37.
- The various monitoring tools available for your operating system will help you identify performance problems with your DB2 server and with your network connections:
 - For OS/2, use SPM/2.
 - For Windows NT, use the Windows NT performance monitor.
 - For UNIX-based environments, use the various monitoring commands outlined in “Performance Monitoring Tools” on page 214.

For more information on performance:

- Search for the keyword “performance” in the DB2 Product and Service Technical Library on the Web, at <http://www-4.ibm.com/software/data/db2/library/> (Note that this information is in English only.)
- See the *Administration Guide: Performance*, and the *System Monitor Guide and Reference*.

Still Have Problems?

If you still have problems with your database server, contact DB2 Customer Support with the following information:

- A description of the events that lead up to the problem
- A full description of the problem, including any SQL messages and associated reason codes
- The `db2diag.log` file and any DB2 trap or dump files (see “First Failure Data Capture” on page 157)
- For UNIX systems, any system core files (see “Using the Core File” on page 210)
- Output from **db2dart** if you suspect inconsistency in the data (see “Miscellaneous Troubleshooting Tools” on page 15).
- Output from **db2level**.
- When working on DB2 Universal Database Enterprise — Extended Edition, the contents of the `db2nodes.cfg` file.

- Output from **db2set -all**.

Chapter 3. Troubleshooting on the Client

This chapter discusses how to correct problems that may occur with a database client.

For more information, see the following documents:

- The *Quick Beginnings* guide for your platform, which provides setup information
- The *Installation and Configuration Supplement* book, which provides detailed information on connecting clients
- “Using All Available Information” on page 9, which discusses how to use the online help available for clients
- “Chapter 4. Troubleshooting Host Communications” on page 61, which describes how to connect to and from host systems

Overview of Client Connectivity

This section provides an overview of how DB2 clients connect to DB2 servers. This information can help you understand the areas where connectivity problems typically occur.

Client workstations are connected to a DB2 server through a communications protocol (for example, TCP/IP, APPC, Named Pipes (NPIPE), NetBIOS, or IPX/SPX). This connection enables applications to run on a client and interact with DB2. Examples of applications are Lotus Approach, Microsoft Access, or user-written applications.

When an application needs to access a database, it uses a `CONNECT` statement. The database alias identified in the `CONNECT` statement references an entry in the client’s *database directory*. This entry contains these important elements:

- The alias name of the database on the server.
- If the database is local, the path where the local database directory resides.
- If the database is remote, the name of a corresponding entry in the node directory. The node directory provides communications information to form the connection.
- The local alias name for the database. The local database alias name and the server database name are usually the same. However, to access databases with the same name that reside on different systems, you must give each database a different local alias name so that your applications can tell them apart.

- When working on DB2 Universal Database Enterprise — Extended Edition, the catalog node number.

The *node directory* indicates where the database is physically located. Each node directory entry specifies which network communication protocol to use to connect to the database, and the protocol parameter values required to make the connection.

The database directory and node directory ensure that access to a database is transparent to users and applications, regardless of where the database physically resides. If a database has been cataloged correctly on the client, you can connect to the database if you know the local database alias name.

The catalog node number represents a node as defined within the `db2nodes.cfg` file, where the **CREATE DATABASE** statement was issued. All system catalog tables for that database reside on the specified node.

For more information about maintaining the information in the database and node directories, see the *Quick Beginnings* guide for your platform.

In addition to the directories, the following are required for client communications to a DB2 server:

- Database manager configuration parameters on the server. For example, to determine whether or not DCE directory services is used, the Directory Services Type (*dir_type*) configuration parameter must be configured. Another example of a configuration parameter is Default Client Communication Protocol (*dft_client_comm*). This parameter indicates the communication protocols that the client applications on the instance can use for remote connections.
- DB2 registry values. Environment variables are stored in a DB2 registry. Update them with the **db2set** command. For information on the DB2 registry values available on your operating system, type `db2set -lr` and press Enter.
For example, to temporarily override the *dft_client_comm* configuration parameter, the DB2CLIENTCOMM registry variable can be modified.
- The `db2cli.ini` and `odbc.ini` files are ASCII files that store values for DB2 CLI configuration. These files are only needed for CLI or ODBC or JDBC connectivity and not general client connectivity. For more information, see the *CLI Guide and Reference*.

For details on client connectivity, see the *Quick Beginnings* guide for your platform.

Client Connectivity Tools

After installing DB2, you can use the following tools to connect a client to a database on a server:

Setup Communications

Sets up communications for server instances, including those that are remote UNIX-based environments. To update communications for a server instance, select the instance in the Control Center and select **Setup Communications** from the pop-up menu.

Note: The communications protocol stack must already be set up.

Client Configuration Assistant

Lets you define connections to databases so that the databases can be used by applications.

Alternatively, use the command line processor to set up clients in UNIX-based environments. Refer to the *Command Reference* for information on the **CATALOG DATABASE** and **CATALOG protocol NODE** commands.

For more information, see the *Quick Beginnings* guide for your platform.

Troubleshooting Tips

This section describes how to deal with some problems users frequently face when they try to connect clients to DB2 servers. It addresses the following topics:

- “Determining the Scope of a Client Problem” on page 46
- “Initial Connection after Installation Fails” on page 47
- “Clients Suddenly Experiencing Problems” on page 48
- “User Name Not Valid on Windows 95 and Windows 98” on page 49
- “TCP/IP Problems” on page 49
- “IPX/SPX Problems” on page 53
- “NetBIOS Problems” on page 55
- “Problems When Using Named Pipes” on page 58
- “APPC Problems” on page 59.

For related information, see:

- “Chapter 2. Troubleshooting the DB2 Universal Database Server” on page 13 if you suspect the problem is with the DB2 server
- “Chapter 4. Troubleshooting Host Communications” on page 61 for troubleshooting host communications
- “Chapter 5. Troubleshooting Applications” on page 71 for troubleshooting applications running on clients, including ODBC applications.

You may also see the *Quick Beginnings* guide for your platform.

Important: This section represents a small sampling of the information available from DB2 Customer Support. For a complete and up-to-date source

of DB2 information, use the DB2 Product and Service Technical Library on the Web, at <http://www-4.ibm.com/software/data/db2/library/> (Note that this information is in English only.)

Determining the Scope of a Client Problem

To determine the possible sources of a problem experienced on the client, ensure that:

- [] The client and server were installed correctly.
- [] The communication products are installed and operational on the client and server.
- [] The client functioned correctly in the past.
- [] The database manager started on the server with the appropriate communications listeners.
- [] You can establish a connection from the client to another server independent of DB2 Universal Database. You can use another command or utility like *ping*, *telnet*, or *ftp* to establish the connection.
- [] You can establish a connection from another client to the server.

This section explains how to:

- Test connections to the server
- Use the `db2diag.log` file to verify that communications listeners are enabled

Testing Connections on the Server

If you are having difficulty establishing a client connection, test a connection from the server's machine:

1. Try connecting to the database on the server using the local database directory entry (to form an IPC connection). If this connection fails, the problem is likely with the server.
2. Test a loopback connection. First, catalog a node pointing to the local machine. Second, catalog a database on this new node. Finally, attempt a connection from the server to itself.

Note: For information on cataloging databases, refer to the *Administration Guide: Implementation* manual.

3. If the connection is successful, then the problem exists on the client. If the connection fails, then the problem may be one of the following:
 - The protocol stack on the server is not working
 - The listeners for the required communications protocols are not started on the server
 - The LAN network is not working

It is recommended that you keep the directory entries set up on the server. This will enable you to diagnose connectivity problems if they reoccur.

Using the db2diag.log File to Diagnose Server Communication Problems

If you determine that a client problem is caused by the server, the db2diag.log file on the server may give more information on what is causing this problem. See “First Failure Data Capture” on page 157 for details on using this file.

For example, you may receive the SQL5043N message after issuing the **db2start** command on the server. This indicates that one or more protocols failed to start successfully. The db2diag.log file may provide additional information to assist you with diagnosing the problem.

When looking for the cause of server problems that may be affecting your clients, perform the following steps:

1. Set the DIAGLEVEL to 4 on the server:

```
db2 UPDATE DATABASE MANAGER CONFIGURATION USING DIAGLEVEL 4
db2 terminate
```

2. Disconnect all applications connected to the server:

```
db2 force application all
```

3. Restart the server:

```
db2stop
db2start
```

4. Examine the db2diag.log file on the server.

For each protocol specified in the DB2COMM registry value, there should be either a message indicating that its listener started successfully, or a message indicating why the protocol listener failed. (For an explanation of listeners, see “Chapter 14. The DB2 Process Model” on page 151.)

If you do not see a message for the protocol, DB2 did not detect the protocol in the DB2COMM registry value and did not attempt to start it.

There are many reasons for a listener not starting, including incorrect installation of a communication protocol or incorrect server configuration.

See the following sections for more information:

- “SQL5043N Received on Server for TCP/IP” on page 51
- “SQL5043N Received on Server for IPX/SPX” on page 55
- “SQL5043N Received on Server for NetBIOS” on page 57

5. If all of the expected communication protocols were started successfully at the server, try the connection again. Examine the db2diag.log file on the client for clues to diagnosing the problem.

Initial Connection after Installation Fails

Initial connection means the first attempt to connect to a remote server after installation of a client.

If the initial connection fails, try to connect to the database locally from the server itself. If this works, the problem is with the connection from the client. If you cannot make the connection, then the problem may be with the database manager on the server.

If there is a problem, perform the following steps:

- [] Ensure that the client and server were installed correctly.
During installation, messages appear on the screen if there are problems. You can also keep installation logs for unattended installations on OS/2, Windows 95, Windows 98, and Windows NT operating systems. Specify error log and history files with the `/I1` and `/I2` options respectively. (If you do not include these options with the installation request, the files are not kept.)
- [] Ensure that all the prerequisite software products were installed.
- [] Ensure that the communication products are installed and operational.
- [] Confirm a connection from another client to the server.
If there is a connection, then the server is functioning properly; if not, the problem may exist on the network or the server.
- [] Ensure that you can get a connection from the client to another server.
If so, then the client is working.

Clients Suddenly Experiencing Problems

If you suddenly encounter a problem with one or more clients which previously connected to the server, review the following questions:

Is a single client experiencing difficulties?

- [] Can other network-enabled applications run on this client? If not, the problem is likely with the communications software, rather than with DB2.
- [] What is unique about the operating environment for this client? Compare it with other clients that can connect to the server.
- [] Were recent changes made that could affect the client? (For example, is another product or fix pack installed?)
- [] On the client, are resource limits (for example, memory) exceeded?

Are multiple clients experiencing difficulties?

- [] Is the LAN available? For example, can you use the **ping** command when using TCP/IP, or issue the **net use** command when using NetBIOS?
- [] Is the server operational? Test a connection at the server machine.

- [] Are the required communication listeners present on the server? See “Using the db2diag.log File to Diagnose Server Communication Problems” on page 47 for details.
- [] On the server, are resource limits (for example, memory) exceeded?

User Name Not Valid on Windows 95 and Windows 98

Symptom

A message is received that the user name is invalid when trying to access DB2 from a Windows 95 or Windows 98 client. (Typically, the SQL1403N message is received.)

Possible Cause

If you or an application you are using does not specify a user name and password as part of a CONNECT statement, an implicit user name and password is used instead. The user name and password you used when you logged on to the operating system are passed with the CONNECT statement. However, the implicit user name and password may be incorrect, causing the CONNECT statement to fail.

Note: Windows 95 and Windows 98 allow you to log on without a user name and password. In this case, the implicit user name and password are null, resulting in a failed CONNECT statement.

Action

For a connect request, you or any applications you create should provide a user name and password as part of the CONNECT statement:

```
CONNECT TO database USER userid USING password
```

- If *authentication* is set to SERVER in the server’s database manager configuration, you must provide a user name and password that are valid on the server. This is the default.
- If *authentication* is set to CLIENT in the server’s database manager configuration, you must provide a user name and a password that are valid on the client.

For more information, see the *DB2 for Windows Quick Beginnings* guide.

TCP/IP Problems

This section outlines some common troubleshooting tips that are related to TCP/IP.

SQL30081N received

Symptom

When trying to connect to a database from a client using TCP/IP and

the connection fails, the message SQL30081N is often received with a protocol-specific error code ECONNREFUSED (often "10061" on Intel-based machines or "79" on UNIX-based environments).

Possible Cause

This error code indicates that the client connection was refused. (See the *Message Reference* for information on the other error codes of SQL30081N.)

Action

Ensure that:

- [] **db2start** was issued and the TCP/IP listener was started on the server. See "Using the db2diag.log File to Diagnose Server Communication Problems" on page 47, for more information.
- [] The TCP/IP stack is functional on both the client and the server.

The TCP/IP protocol tester, `pctt`, can be used to verify that the protocol works across the network. The tester can be found in the `bin` subdirectory of the `sql1lib` directory.

Or, from the client, try using the **ping** command with the server's host name.

- [] The directories are cataloged correctly. In particular, ensure that:
 - The database directory entry points to the correct node directory entry.
 - The service or port name in the `svcname` field in the node directory maps to the same port number as the `svcname` in the server's database manager configuration.
Try cataloging the node by specifying the `svcname` as an available port number rather than as a service name.
 - The IP address or host name that is specified in the `hostname` field of the node directory is correct. (To verify the entry, use the **ping** command to test the host name or IP address.)

To verify and change directory entries, use:

- The Client Configuration Assistant
 - The **CATALOG DATABASE** and **CATALOG TCPIP NODE** commands. See the *Command Reference* for more information.
- [] The TCP/IP services file is not corrupted, especially if you used a text editor to update it.

If you added the port settings line to the end the file, it must be followed by a blank line.

- [] The port number used is the same in the TCP/IP services files of the client and the server instance. The port number is uniquely defined within the TCP/IP services file.

SQL5043N Received on Server for TCP/IP

Symptom

The SQL5043N message is received on the server.

Possible Cause

- TCP/IP is not started on the server machine.
- Database manager configuration is not correct. (For example, the *svccname* configuration parameter defined is not correct.)
- The TCP/IP services file is not correct. (For example, the *svccname* configuration parameter in the database manager configuration is not defined in the file.)

Action

View the `db2diag.log` on the server. Look for messages that may provide more information.

Client Application or Query Appears Suspended

Symptom

A client application accessing a remote DB2 server appears to be suspended.

Possible Cause

The client was not notified that the server is down.

Due to the characteristics of the TCP/IP protocol, the TCP/IP subsystem on one host may not be notified of the failure of its partner on another host.

DB2 uses TCP/IP's connection KEEPALIVE option to detect if there is a connection failure. This option transmits a message periodically to determine if the partner is still alive. If the partner fails to respond to this message, the connection is considered to be broken, and an error is returned.

A client may appear to be suspended if the KEEPALIVE settings on the client have been set to check TCP/IP connections infrequently and the server has gone down.

Action

To remove an agent process that is suspended at the server, use the **FORCE APPLICATION** command.

If the problem persists, change the value of the `KEEPALIVE` settings to change the time interval at which messages are transmitted to detect a connection failure.

Note: `KEEPALIVE` settings affect all TCP/IP applications running on the machine.

- For Windows 95, Windows 98, and Windows NT:
Use the `KeepAliveTime` TCP/IP configuration parameter in the registry. The `KEEPALIVE` parameter may be created if it does not exist under the Parameters registry subkey. Add this parameter to:
`HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters`

The default value is two hours.

- For OS/2:
Use the `inetcfg` command. (For OS/2 TCP/IP Version 2.0, you must apply the fix CSD UN64092 to use this command.)
- For AIX:
Change the values of the network options `tcp_keepidle` and `tcp_keepintvl` with the `no` command (for details, type `man no`). The default value is two hours.
- For HP-UX systems:
Change the values of the network options `tcp_keepstart` and `tcp_keepfreq` with the `net tune` command (for details, type `man net tune`).
- For Solaris systems:
Change the value of the network option `tcp_keepalive_interval` with the following command:
`ndd -set /dev/tcp tcp_keepalive_interval value`

(For details, type `man ndd`.)
- For SINIX systems:
Change the values of the network options `TCPTV_KEEP_IDLE_SECS` and `TCPTV_KEEPINTVL_SECS` with the following commands:
`/etc/conf/bin/idtune TCPTV_KEEP_IDLE_SECS value`
`/etc/conf/bin/idtune TCPTV_KEEPINTVL_SECS value`

(For details, type `man idtune`.) The default value is 2 hours and 10 minutes.
- For other platforms:

See your TCP/IP documentation for details on configuring the `KEEPALIVE` setting. If it is not supported by the TCP/IP stack, then it is not used by DB2.

IPX/SPX Problems

This section provides troubleshooting tips for the IPX/SPX communication protocol.

SQL30081N Received

Symptom

The SQL30081N message is received when trying to connect to a DB2 server.

Action

Ensure that:

- [] Your Novell Netware TLI*.DLL files are at the correct level. (Use the DB2 or Novell Web sites to find out about the latest Novell Netware fixes.)
- [] If file server addressing mode is being used, the file server and object names in the client's node directory entry match the values of *fileserver* and *objectname* in the database manager configuration file on the server. These names must be the same, and both should be in uppercase.
- [] If file server addressing mode is being used, the database server was registered on the file server sometime after DB2 was installed and configured. For more information on the **REGISTER** command, see the *Command Reference*.
- [] If the server machine was moved on the network and its IPX/SPX internetwork address was changed, the DB2 server's internetwork address was deregistered before any change, and reregistered after the change.
- [] If file server addressing mode is being used to connect to the database, the object name that represents the DB2 server instance (that is, stores the IP address of the server instance) is in the bindery of the file server.
- [] If direct addressing mode is being used, the following values are in the client's node directory:
 - The file server entry is specified as an asterisk (*).
 - The value of *objectname* is the server's IPX/SPX internetwork address.Issue **db2ipxad** on the server to get the server's IPX/SPX internetwork address. This command is in the `bin` subdirectory of the `sql11ib` directory.

At that same location, the IPX/SPX protocol tester, `pct i`, can be used to verify that the protocol works across the network.

SQL30081N on OS/2

Symptom

For OS/2, the SQL30081N message is received with the `t_open` function and a reason code of 8.

Possible Cause

- The NetWare product is not functioning correctly because it was not installed properly, not configured properly, or corrupted.
- There are not enough system resources to handle the request.

Action

Ensure that:

- [] For DOS and Windows, the first two lines of the `net.cfg` file on the client are:

```
ECB COUNT=50  
DATA ECB COUNT=89
```

The `net.cfg` file is in the root directory.

- [] The FILES parameter in the OS/2 `CONFIG.SYS` file is set to an appropriate level. This parameter determines the maximum number of files that can be used by all programs running in DOS and Win-OS/2 sessions.
- [] For OS/2, the `AUTOEXEC.BAT` file does not contain a path to the OS/2 NETWARE directory. This path is for the DOS, Windows, and OS/2 environment, and the DLL files in the OS/2 NETWARE directory are OS/2 DLL files. In some cases, Windows and OS/2 NetWare DLL files have the same name, but DOS and Windows cannot load or run OS/2 DLL files.

Connection to OS/2 Server Hangs Unexpectedly

Symptom

An IPX/SPX connection to a DB2 for OS/2 server hangs, even though the connection was previously successful.

Possible Cause

There may be NetWare resource problems.

Action

To ensure that you have provided yourself an adequate number of connection resources, verify that the `net.cfg` file provides:

- 128 sockets in the protocol stack `ipx` subsection

- 50 sessions in the protocol stack spx subsection

See your IPX/SPX documentation for details about these configuration parameters.

The `net.cfg` file is usually in the root directory. Alternatively it may be in the `NETWARE` directory for OS/2. Check the system boot-up screen to determine which `net.cfg` file is being used.

SQL1109N When Connecting from a Windows or OS/2 Client

Symptom

The SQL1109N error message is received when trying to connect from a Windows or OS/2 client.

Possible Cause

There are two versions of the `NWCALLS.DLL` and the `TLI_SPX.DLL` files: one for OS/2 and one for Windows. These files may not be in the correct location.

Action

The `NWCALLS.DLL` file from Novell's `NWDLL2.exe` package should be in the `WINDOWS\SYSTEM` directory. Ensure that Windows is not trying to load the OS/2 version of the `NWCALLS.DLL` file.

SQL5043N Received on Server for IPX/SPX

Symptom

The SQL5043N message is received on the server.

Possible Cause

- IPX/SPX is not started on the server machine.
- The database manager configuration is not correct. (For example, the *filesrver*, *objectname*, or *ipx_socket* parameters are not correct.)

Action

View the `db2diag.log` file on the server. Look for messages that provide more information.

NetBIOS Problems

This section provides troubleshooting tips related to the NetBIOS communication protocol. NetBIOS is not used in UNIX-based environments.

SQL30081N Received

Symptom

If you cannot connect to the server from a client, you typically receive the SQL30081N message with return code of 0x14.

Action

Use the following checklist to diagnose the problem. If you are using directory caching and changing database or node directories, you must use the **TERMINATE** command at the client for your changes to come into effect.

- [] Is the NetBIOS listener started on the server? Check the NetBIOS resources in the `db2diag.log` file to see if there are problems. See “SQL5043N Received on Server for NetBIOS” on page 57, for more information.
- [] Are your client and server set to start NetBIOS support?
NetBIOS must be included in both the server’s and the client’s configurations:
 - On the server, use the `db2set DB2COMM` command to verify that NetBIOS is a supported protocol.
 - The node name in the client’s database directory must match the alias name of a NetBIOS node entry in the client’s node directory. The *nname* that corresponds to this alias name in the client’s node directory must match the value of the configuration parameter *nname* in the server’s database manager.
If these three names are not the same, you must recatalog the node entry at the client. See the *Command Reference* manual to find out more about the **CATALOG** command.
- [] Was the correct adapter number specified when the node was cataloged on the client?
Check the adapter specified in the client’s node directory. This adapter number must match the adapter configured for NetBIOS communications at the client.
Usually, the adapter number is 0. However, if more than one adapter is configured, you must ensure that the adapter being used by the client is for a LAN that can reach the server.
With native NetBIOS on Windows NT, the adapter number is called the logical LAN adapter number (Lana number). To check its value perform the following steps:
 1. Select the **Network** icon from the Control Panel.
 2. From the **Services** tab, select the NetBIOS interface.
 3. Select **Properties**.
 4. The Lana number associated with the network route Nbf must match how you have cataloged your nodes.

[] Is there a physical LAN layer problem with the gateway, bridge, router, or LAN cables?

[] Is name filtering occurring on the bridge or router that connects the server's LAN to the client's LAN?

A LAN bridge or router may be preventing client requests from reaching a server on a different LAN because it is ignoring names with the DB2 name structure. Discuss this possibility with your LAN administrator.

[] Are the server and client using compatible NetBIOS stacks?

Ensure that the server and its clients are using native NetBIOS or identical NetBIOS emulation.

The NetBIOS protocol tester, `pctn`, can be used to verify that the protocol works across the network. The tester can be found in the `bin` subdirectory of the `sql1ib` directory.

Connection Ends Suddenly

Symptom

You can successfully connect a client to a server, but the connection ends suddenly, usually with the SQL30081N message with a return code of 0x08 or 0x18 logged in the `db2diag.log` file.

Possible Cause

The NetBIOS protocol is reporting a time-out to the DB2 server, possibly because of a physical LAN problem. This happens occasionally on OS/2 systems.

Action

Report the situation to your NetBIOS service organization, and use the DB2 Product and Service Technical Library on the Web, at <http://www-4.ibm.com/software/data/db2/library/> for some suggested fixes. (Note that this information is in English only.)

SQL5043N Received on Server for NetBIOS

Symptom

The SQL5043N message is received on the server.

Possible Cause

The NetBIOS listener is not started.

Action

View the `db2diag.log` file on the server. Look for the following entries:

[] DIA3426C:

- Update the database manager configuration with a valid *nname*, and stop and start the instance.
- [] DIA3409I or DIA3420C:
- Increase the number of Network Control Blocks (NCBs), sessions, or names by the difference between the number requested and the number that could be allocated.
These values are kept in:
 - For OS/2, the `protocol.ini` file
 - For the Windows 95 and Windows 98 operating systems, the NetBIOS network control settings
 - Reduce the NetBIOS resources used by other NetBIOS applications. These resources may be restricting the DB2 NetBIOS resource request.
 - If not enough NCBs can be allocated, find out the values of the environment variables `DB2NBSENDNCBS` and `DB2NBRECVNCBS`. If their values are greater than their defaults, you may want to reduce them.
The number of NCBs that the DIA3420C message said could be allocated must be greater than the sum of these values plus the values for `DB2NBINTRLISTENS` and `DB2NBXTRANCBS`.
 - If not enough sessions can be allocated, you might try setting the environment variable `DB2NBSESSIONS` to the value that the DIA3420C message said could be allocated.
 - Increase the NetBIOS resource pool limit values in the `protocol.ini` file to satisfy all of the resource requests made by NetBIOS applications and drivers on the workstation adapter.

Problems When Using Named Pipes

SQL30082 Received

Symptom

In a workgroup environment, a user trying to access the database server using named pipe receives the SQL30082N message with a reason code of 18 (named pipe access denied).

Possible Cause

Before a named pipe can be accessed at a remote server, there must be an open session at the system level. In this case, the user authentication failed at the server. Therefore, the session is a null session that does not have the credentials to access the named pipe.

Action

Do one of the following:

- Create the client's user name and password on the remote server.
- Enable the Guest account at the remote server
- Share the network resource of the remote server. For example, perform **net use** to access the server's network drive, where you can use a user name and password valid at the remote server.

APPC Problems

Review the following checklist when experiencing problems with APPC connections:

- [] Did you follow the instructions in the *Quick Beginnings* guides for installing clients or servers? These guides give step-by-step instructions for APPC configuration.
- [] If VTAM is used, are the correct Logical Unit (LU) names defined for the server and client?
- [] Is the correct TP name defined?
- [] If SNA is used, are the correct SNA node IDs defined?
- [] Are you using appropriate APPC security in the DB2 node directory, and appropriate DB2 authentication in the DB2 database directory?

See your *Quick Beginnings* guide for details on which types of authentication and security can be used together. Note that the security setting in the DB2 node directory overrides any SNA security configuration.

Still Have Problems?

If you still have problems with your client, contact DB2 Customer Support with the following information:

- A description of the events leading up to the problem
- A full description of the problem, including any SQL messages and associated reason codes
- The db2diag.log file and any trap or dump files, for both the client and server. See "First Failure Data Capture" on page 157, for more information.
- Output from the following commands:
 - **LIST DB DIRECTORY**
 - **LIST NODE DIRECTORY**
 - **GET DBM CONFIG**
 - **db2level**

- ODBC/CLI traces for ODBC applications (see the *CLI Guide and Reference* manual).

Your DB2 Customer Support representative may also ask for and describe how to take a protocol stack trace.

Chapter 4. Troubleshooting Host Communications

You can establish connections to and from host database systems, including DB2 Universal Database for OS/390, DB2 for MVS/ESA, DB2 for VM, DB2 for VSE, or DB2 Universal Database for AS/400. Two types of host connections exist:

- Host clients can connect to a DB2 Universal Database server to query and update data. For information on setting up this type of connection, see your server's *Quick Beginnings* guide.
- DB2 Connect connections allow Windows, OS/2, and UNIX-based clients to interact with host databases. These connections are enabled with the following products:
 - DB2 Connect Personal Edition, for single users
 - DB2 Connect Enterprise Edition, for multiple users through a gateway

DB2 Connect gateway capability is also provided by DB2 Universal Database Enterprise Edition and DB2 Universal Database Enterprise - Extended Edition.

For information on DB2 Connect, see the following documents:

- *DB2 Connect Enterprise Edition for OS/2 and Windows Quick Beginnings*
- *DB2 Personal Edition Quick Beginnings* for your platform
- *DB2 Connect User's Guide*
- *Connectivity Supplement*

DB2 database communication requires Distributed Relational Database Architecture (DRDA). For information, you can order the following publications separately:

- *DRDA Connectivity Guide*
- *DRDA Problem Determination Guide* for information on error analysis in a DRDA environment

DRDA requires communication layers between systems. The *Quick Beginnings* guides provide detailed steps to establish this communication, which can be enabled in several ways:

- Systems Network Architecture (SNA) enables communication between most systems. Its layered structure allows transparent access to the data on those machines set up to use the SNA network. SNA network communication for LU 6.2 is enabled through the APPC communication protocol. For more information, see "APPC Problems" on page 59.

- TCP/IP is supported by DB2 Universal Database and by Version 5.1 of DB2 for OS/390. For more information, see “TCP/IP Problems” on page 49.

This chapter provides the following troubleshooting sections:

- “Troubleshooting Host Connections”
- “Troubleshooting Tips for Host Connections to DB2 Universal Database” on page 67

Important: These sections represent a small sampling of the information available from DB2 Customer Support. For a complete and up-to-date source of DB2 information, use the DB2 Product and Service Technical Library on the Web, at <http://www-4.ibm.com/software/data/db2/library/> (Note that this information is in English only.)

Troubleshooting Host Connections

The error message SQL30081N is the most common error returned when setting up connections. If you are receiving this error message, the following points provide troubleshooting guidance.

Direct Connection

In this case, you are connecting directly from a DRDA requestor to a DRDA server with no DB2 Connect gateway in between.

Note: A DB2 Client (CAE) cannot make a direct connection to a DRDA server. A DB2 Connect gateway is needed. For more information on troubleshooting in this scenario, see “Gateway Connection” on page 64.

TCP/IP Connection

[] Can you ping the DRDA server?

If you cannot:

- Verify the IP address
- Ensure that TCP/IP is active at both the DRDA requestor and the DRDA server
- Consult your Network Administrator

If you can: Continue to the next question.

[] Are the port number and service name properly defined?

- Did you use the DRDA default port number 446?
- Ensure the services file is updated correctly if you are using the service name instead.

[] Can any DRDA requestor connect to this DRDA server?

If none can:

- Ensure that the server DBMS is active (that the DDF is started).
- Verify the limits.
- Verify the server's zparm MAXDBAT is set to a value greater than zero. The default is zero.
- Verify the startup message appears in the logs.

If at least one can:

- Verify that you are connecting to the right DRDA server.
- If DB2 Connect is the requestor, verify the catalog information. Ensure the database is cataloged with the authentication as DCS; and that the DCS directory and the node directory are cataloged.
- The SQL1403N authentication error is always returned from a distributed environment such as UNIX, OS/2, or Windows operating system. The SQL30082N authentication error is returned from a DRDA Application Server (AS) such as DB2 for OS/390 or DB2 for AS/400. If you see a SQL1403N error when connecting to a DRDA AS and are not trying to authenticate on the gateway, you may not have the database authentication set to DCS.

SNA Connection

[] Can you manually establish a SNA link?

If you cannot:

- Verify the LAN destination address
- Verify the partner node ID and node type
- Ensure the communication controller is working

If you can: Continue to the next question.

[] Can you manually establish a SNA session?

If you cannot:

- Verify the SNA configuration. Ensure all logical unit (LU) and physical unit (PU) names are correct.
- Verify the correlating VTAM definition.
- Check the SNA configuration. It should be one of: SNA Server for AIX, Communications Server (formerly Communications Manager) for OS/2, IBM Communications Server for Windows NT, or Microsoft SNA Server for Windows NT. See *Quick Beginnings* for information on how to configure SNA.
- Contact your SNA vendor support organization.

If you can: Continue to the next question.

- [] Can any DRDA requestor connect to this DRDA server?

If none can:

- Ensure that the server DBMS is active (that the DDF is started).
- Verify the limits.
- Verify the server's zparm MAXDBAT is set to a value greater than zero. The default is zero.
- Verify the startup message appears in the logs.

If at least one can:

- Verify that you are connecting to the right DRDA server.
- If DB2 Connect is the requestor, verify the catalog information. Ensure the database is cataloged with the authentication as DCS; and that the DCS directory and the node directory are cataloged.

Gateway Connection

If you are connecting from DB2 clients to a DRDA server using a DB2 Connect gateway:

- [] Can you connect to the DRDA server from the DB2 Connect gateway?

If you cannot, see "Direct Connection" on page 62.

If you can: Continue to the next question.

- [] Is the DB2 registry variable DB2COMM set properly at the DB2 Connect gateway?

It should be set to APPC, TCPIP, or both, depending on which protocol you are using.

- [] Verify the catalog information on the client.

Please see "Chapter 3. Troubleshooting on the Client" on page 43 for additional information.

Troubleshooting Tips for DB2 Connect

This section describes how to deal with some frequent problems faced by users when trying to connect to host databases using DB2 Connect. It addresses the following topics:

- "Problems Encountered after Initial Connection" on page 65
- "Changing the Number of Connections" on page 65
- "Authentication Problems" on page 66
- "Windows Clients Do Not Connect" on page 67

Problems Encountered after Initial Connection

If you installed a client and could make an initial connection, but then experience a problem with it later, use the following checklist as a starting point to narrow down the scope of the problem.

Can you establish a host session?

- [] See “Troubleshooting Host Connections” on page 62.
- [] If you suspect a problem with your communication protocol, see “Chapter 3. Troubleshooting on the Client” on page 43.

Are there any special or unusual operating circumstances?

- [] Is this a new application?
- [] Are new procedures being used?
- [] Have any of the software products or applications been changed since the application or scenario last ran successfully?
- [] For application programs, what application programming interfaces (APIs) are called?
- [] Have other applications that use the software or communication APIs been run on the user’s system?
- [] Are there recent changes that might be affecting the system? For example, has maintenance been applied?

Is there relevant diagnostic information?

- [] Were any SQL messages or SQL states returned? To look up an SQL state or SQL code, see the *Message Reference*.
- [] Check the `db2diag.log` file on the server, particularly for SQLCA information. For more details, see “First Failure Data Capture” on page 157.

Changing the Number of Connections

To change the number of connections that a DB2 Connect Enterprise Edition gateway supports:

- Change the session limit within the mode definition.
- Define the session limit in the APPLID definition for the DB2 subsystem in VTAM (for DB2 for MVS/ESA and DB2 for VM host sessions).
- Change the `MAXDBAT` parameter in the DSNZPARM dataset for DB2 for MVS/ESA.
- Check the maximum number of database manager agents specified with the DB2 database manager configuration parameter *maxagents*.

VTAM and Communications Server will negotiate how many DRDA connections are allowed (with a minimum of two). DB2 for MVS/ESA allows only the number of connections defined in the DSNZPARM dataset.

Authentication Problems

Your operating system and communications products may affect the authentication of DB2 Connect sessions. For information, see the *Administration Guide: Implementation*.

Note that:

- User names and passwords may be case-sensitive. In particular:
 - For DB2 for MVS/ESA, user names and passwords may be case-sensitive depending on which PTFs were applied.
 - Passwords on UNIX-based environments are always case-sensitive.
 - If OS/2 uses User Profile Manager (UPM), user names and passwords are converted to uppercase. This may be an issue if you are using a DB2 Connect Enterprise Edition gateway on a UNIX machine.
- For NetBIOS connections, when you set up a connection to a host from any client via a DB2 Connect Enterprise Edition gateway, the user name and password are passed through to the target server for verification. For example, authorization is always forced to DB2 for MVS/ESA.

Because of this behavior, DCS authentication does not apply for the NetBIOS protocol: only SERVER authentication should be used.

SQL1403N When Issuing a Connect to DB2 for MVS/ESA

Symptom

The SQL1403N message occurs when a DB2 Connect client tries to connect to DB2 for MVS/ESA using DCS authentication.

Possible Cause

System tables are not set up correctly in DB2 for MVS/ESA to process the incoming request.

Action

Ensure that entries are correct in the SYSIBM.SYSLUNAMES and SYSIBM.SYSUSERNAMES tables.

For more information, see the MVS Server worksheet in the *DB2 Connect Enterprise Edition for OS/2 and Windows Quick Beginnings*.

SQL1402N and SQL30082N When Connecting to DB2 for MVS/ESA

Symptom

The SQL1402N and SQL30082N messages are received when a client tries to connect to DB2 for MVS/ESA using DB2 Connect.

Possible Cause

You may have established incorrect security authentication between the client, DB2 Connect, and the server.

Action

Specify DCS authentication on your DB2 Connect Enterprise Edition gateway.

If you use SERVER authentication with an OS/2 server, ensure that the user name and password are also defined on the OS/2 server.

Note that the Control Center always assumes SERVER authentication, so you must use the command line processor to set other authentication types.

Windows Clients Do Not Connect**Symptom**

The SQL30073N message is received with reason code X'119C' when a Windows client tries to connect to a host database.

Possible Cause

Your host does not recognize the code pages used by your client for one of the following reasons:

- The host cannot support them.
- The host was not set up to support them.
- The host requires a PTF to be applied.

Action

If possible, enable the necessary code page support on your host.

If you cannot enable this support, a workaround is to use the DB2CODEPAGE keyword in your client's configuration. For more information, see the section on configuring national language support in the *DB2 Connect Enterprise Edition for OS/2 and Windows Quick Beginnings*.

Troubleshooting Tips for Host Connections to DB2 Universal Database

See the *Quick Beginnings* for your server for detailed instructions on how to set up connections so that host clients can access DB2 Universal Database databases.

Use the following checklist as a starting-point for diagnosing problems with these connections:

- [] Are the application server and host correctly configured? (For MVS hosts, ensure that SPUFI is configured as well.)
- [] Are the appropriate authentication types being used?

The following authentication settings can be used in the database manager configuration at the DB2 Universal Database server:

- When using Communications Server/2 on OS/2 or SNA Server/6000 on AIX, SERVER or CLIENT
- In all other cases, CLIENT or DCS

- [] Is the communications layer set up properly on the machine where DB2 Universal Database resides?

For example:

- On OS/2, Communications Manager or Communications Server must be set up.
- On AIX when using SNA, the TP name used by the host client must match the TP name on the AIX machine.

- [] Is two-phase commit properly set up if it is being used?

- [] Are the same bind options used on the host and on DB2 Universal Database?

You must select bind options that are supported by DB2 Universal Database and override defaults on the host (date and time options, for example). On MVS, you may want to bind SPUFI against the DB2 Universal Database database.

Troubleshooting Installation Problems on Windows 2000

You may find you have a problem installing DB2 Connect Personal Edition on Windows 2000. The IBM Personal Communications product packaged with DB2 in Version 6 does not work on Windows 2000. If you already have a product installed using a SNA stack that is Windows 2000 ready, then you will not be affected by this limitation. Using DB2 Version 6.1 with FixPak 3 will give you a Windows 2000 ready version of IBM Personal Communications (PCOMM Version 4.2 CSD2).

DB2 Version 6.1 and later will not install successfully when done by someone who is a member of the Users Group on Windows 2000. With Windows 2000 Professional, anyone performing an installation must be a member of the Power Users Group or higher. For other Windows 2000 products, only members of the Administrator Group are able to install DB2.

Still Have Problems?

If you still have problems with your connection and you suspect a host problem, check your DB2 Universal Database for AS/400, DB2 for MVS/ESA, DB2 for VSE, or DB2 for VM documentation.

If you suspect problems with the DB2 Connect product, contact DB2 Customer Support with the following information:

- The `db2diag.log` file for gateways, clients, and DB2 Universal Database servers. For details on how to work with these files, see “First Failure Data Capture” on page 157.
- ODBC/CLI traces for ODBC applications. (See the *CLI Guide and Reference* manual.)
- Output from the **LIST DCS DIRECTORY** command.
- Output from **db2level**.

At that time, you may be requested to take SNA, DB2 Connect, or DRDA traces. See “Chapter 16. Traces” on page 177.

Chapter 5. Troubleshooting Applications

Applications are the means by which most users work with DB2 database data. When things go wrong with applications, many people are affected and your business may suffer.

Testing and debugging applications should be done in a test environment before the application is promoted and propagated to your production environment. Information on how to design, develop, compile, link, test, and debug applications is found in the following documents:

- The Application Development with DB2 Universal Database web site at <http://www-4.ibm.com/software/data/db2/udb/ad/> provides updated information on developing applications for DB2. This web site includes the latest information on developing applications for DB2 Universal Database using the embedded SQL, Java, or CLI/ODBC APIs.
- The *Application Development Guide* and <http://www-4.ibm.com/software/data/db2/java/> provide information on how to design and code application programs that use embedded SQL (Structured Query Language) statements or SQLJ (Embedded SQL for Java) to access DB2 databases in supported languages. The book and the web site provides details on how to code user-defined functions and stored procedures (including stored procedures using Java).
- The *Administrative API Reference* provides information on the use of application programming interfaces (APIs) to execute DB2 administrative functions.
- The *CLI Guide and Reference* provides information on coding with Call Level Interface (CLI), including stored procedures using CLI. CLI is a callable SQL interface based on the X/Open CLI specification, compatible with Microsoft Corporation's ODBC.
- The *Administration Guide: Implementation* provides information on authorization and the *Administration Guide: Performance* has information on tuning application performance.
- The *Application Building Guide* provides environment setup information and step-by-step instructions on how to compile, link, and run DB2 applications.

Troubleshooting Tips

This chapter describes some problems frequently encountered by developers of applications that access a DB2 server. These tips to prevent or overcome problems with applications were created from information available from DB2 Customer Support. DB2 Customer Support hears about customer problems and concerns, and so is ideally suited to provide the tips found here. This chapter presents the following topics:

- “Developing Applications”
- “Compiling and Linking Applications” on page 73
- “Running Applications” on page 74
- “Using CLI/ODBC/JDBC Applications” on page 75
- “Using SQLJ Applications” on page 79
- “Using the db2ocat Tool” on page 80

In addition, see “Developing Applications in DB2 Enterprise - Extended Edition” on page 98.

Important: For a complete and up-to-date source of DB2 information, use the DB2 Product and Service Technical Library on the Web, at <http://www-4.ibm.com/software/data/db2/library/> (Note that this information is in English only.)

Developing Applications

When developing applications, ensure that:

- [] Embedded SQL statements are syntactically and semantically correct. If possible, try to step through SQL statements manually using the command line processor before embedding them in an application. This method ensures that you are using valid SQL statements that perform the required function.
- [] You have the authorities and privileges needed to execute the SQL statements.
- [] The application is not holding locks for excessive periods of time. For more information, see “Diagnosing locking problems” on page 36.
- [] Your application is bound before you issue any static SQL statements. For information on the sqlabndx bind API, see the *Administrative API Reference*.
- [] You provide code to handle SQL errors in applications that connect to DB2.

A DB2 application should check the value of the SQLCODE or the SQLSTATE element of the SQLCA data structure following each SQL statement. If it encounters a negative SQLCODE, it should take

appropriate actions. For example, it should capture the SQLCODE and the SQLCA structure for later analysis.

For debugging purposes, the `sqlainfp()` API provides a formatted error message that you can display. For examples on how to use this API, see the *Application Development Guide*, and use the following sample programs:

- `utilapi.c` and `db2mon.c` for C
- `checkerr.cbl` for COBOL

Negative SQLCODEs that result in a terminated connection are logged in the DB2 diagnostic log (`db2diag.log`) file. The database monitor application ID is also logged with the SQLCA on the server to enable you to identify the client and application.

- [] You provide enough memory for the variables you are using.
- [] You consider code page conversions.

If you are writing programs to handle data across various operating systems, you must consider the various code pages you will be using. See the *Application Development Guide* for details.

- [] If your applications use user-defined functions, stored procedures, distributed unit of work or multi-threading, you follow the guidelines in the *Application Development Guide* and *CLI Guide and Reference manuals*.
- [] There are no incompatibilities with previous releases.

If the application is legacy code for a previous version of DB2, review the lists of incompatibilities in the *Administration Guide: Planning*.

Compiling and Linking Applications

At compile time, ensure that:

- [] Your compiler environment is correct and all include files and environment variables are current. See your compiler's documentation for details.
- [] Your compiler options are correct. See the *Application Building Guide* book for information specific to a supported compiler.
- [] No compiler error messages are received. (Correct any syntax errors before proceeding.)

At link time, ensure that:

- [] The link options and link libraries are defined properly. For information, see the instructions for your operating system in the *Application Building Guide*.

Running Applications

When you experience problems at run time, ensure that:

- [] You have EXECUTE privilege on packages that contain embedded SQL statements, and the required authorities and privileges for all dynamic SQL statements.
- [] You have looked for information about Java related applications, JDBC (Java Database Connectivity) and SQLJ (Embedded SQL for Java), on the IBM DB2 and Java page at <http://www-4.ibm.com/software/data/db2/java/>
- [] You have investigated any error messages you received. For example, embedded SQL statements, DB2 API, or CLI statements may return error messages.
- [] You step through embedded SQL statements manually from the command line processor to find problems. This technique is recommended when statements in an application run but produce the wrong results.
- [] You solve all application errors. A debugger may assist you.
- [] The application is not holding locks for excessive periods of time. See “Diagnosing locking problems” on page 36, for more information.
- [] If you are using CLI/ODBC/JDBC applications, review the considerations in “Using CLI/ODBC/JDBC Applications” on page 75.

Diagnosing a Suspended or Looping Application

After you start a query or application, you may suspect that it is either suspended (because it does not show any activity) or looping (because it shows activity, but no results are returned to the application). See “Responding to Suspensions and Loops” on page 7 for information on how to restart your system in this situation.

After you restart your system, you may find the following tools helpful to diagnose problems with applications that cause loops or suspensions:

- The debugger you use with your compiler.
- The database system monitor. See “Diagnosing locking problems” on page 36 and the *System Monitor Guide and Reference* for more information.

A function of the database system monitor is to display the status of all active agents. This information is useful when you are debugging. Ensure that statement collection is enabled before you run the application again, and immediately after you run **db2start**. Use the following command:

```
db2 UPDATE MONITOR SWITCHES USING STATEMENT ON
```

In a partitioned database environment use:


```
db2_a11 ";db2 UPDATE MONITOR SWITCHES USING STATEMENT ON"
```

The semi-colon(;) and **db2_all** command combine to improve the performance of the **UPDATE** command by issuing the command on all partitions in the database.

If you suspect that your application or query is either suspended or looping, issue the following command to capture snapshot information. This may help you identify the source of the problem.

```
db2 GET SNAPSHOT FOR ALL APPLICATIONS ON database
```

In a partitioned database environment use:

```
db2_a11 ";db2 GET SNAPSHOT FOR ALL APPLICATIONS ON database"
```

For more information on **db2_all**, see “Commands for DB2 Enterprise - Extended Edition” on page 215.

You might also consider modifying your application so that it writes to an application log file to probe where the code is suspended or looping.

Using CLI/ODBC/JDBC Applications

This section covers some common troubleshooting tips related to CLI/ODBC/JDBC applications.

Developing CLI/ODBC/JDBC Applications

When developing CLI/ODBC/JDBC applications, note the following:

- [] You must use the PATCH1/PATCH2 keyword to specify a work-around for any known compatibility problems with CLI/ODBC/JDBC applications. Set this keyword by editing the `db2cli.ini` file or (for OS/2, Windows 95, Windows 98, and Windows NT) by clicking on the **Service** tab of the CLI/ODBC Settings in the Client Configuration Assistant. See the *Installation and Configuration Supplement* guide for information on this keyword.

CLI/ODBC/JDBC Applications Fail when Connecting to Server

Symptom

An application such as Microsoft Query, Microsoft Access or Lotus Approach fails when connecting to a DB2 server. The SQL1003N message is typically received or a General Protection Fault (GPF) occurs.

Action

Ensure that:

- [] The DB2 CLI/ODBC driver is configured correctly.

You can configure the driver by using the DB2 Client Configuration Assistant:

1. Select the DB2 database alias you are configuring.
2. Click **Properties** to display the Database Properties window.
3. Click **Settings** to display the CLI/ODBC Settings window.
4. Click **Advanced**. A window appears where you can set the configuration keywords. In particular, ensure that:
 - LONGDATACOMPAT=1 if you are accessing Large Object Binary (LOB) data.
 - UNDERSCORE=0 if there are underscore characters (_) in table names.

Alternatively, edit the `db2cli.ini` file and ensure that the keywords appear in the list of parameters listed after the alias of the database you are connecting to. If they are not there, edit the file and add them on separate line.

Note: Customizing the CLI Driver may cause some applications to fail, including some DB2 Java-based utilities. It is a good practice to create multiple database aliases for different CLI/ODBC/JDBC-based applications so that the driver settings can be modified independently for each application.

- [] The DB2 CLI/ODBC driver is optimized for Lotus Approach, Microsoft Access, or Visual Basic.

You can optimize the driver using the DB2 Client Configuration Assistant:

1. Select the DB2 database alias you are configuring.
2. Click **Properties** to display the Database Properties window.
3. Click **Settings** to display the CLI/ODBC Settings window.
4. Select the **Optimize** push button. A window appears where you select one of applications shown above. The CLI driver will be automatically configured for that application using pre-defined patch values and configuration options.

- [] You have compatible Microsoft ODBC driver manager files. Updating these files can solve GPFs.

Mixed Case Names Created in ODBC Applications Causing Problems

Symptom

Table or field names in DB2 are in mixed case. They may have been created in one type of application such as an ODBC application like Lotus Approach, and then accessed using another method such as the DB2 Command Line Processor. For example, the problem with accessing tables could be seen when the following error message is returned from an SQL statement such as `SELECT * FROM table1`:

```
SQL0204N "USERID.TABLE1" is an undefined name. SQLSTATE=42704
```

This error message is given even though the table exists, and is shown in the Control Center list of tables or in a `LIST TABLES` command.

For example, the problem with accessing columns happens when the following error message is returned from an SQL statement such as `SELECT column1 FROM table2`:

```
SQL0206N "COLUMN1" is not a column in an inserted table, updated table, or any table identified in a FROM clause or is not a valid transition variable for the subject table of a trigger. SQLSTATE=42703
```

This error message is given even though:

1. COLUMN1 exists, and shows up in the Control Center

or

the SQL statement: `SELECT tablename, colname FROM SYSCAT.COLUMNS WHERE tablename='table2'` returns the appropriate information about COLUMN1.

2. The SQL statement `SELECT * FROM table2` returns all the columns, including the COLUMN1 column.

This problem can also appear if the object is created with a name that uses all uppercase characters, and the object name is specified using mixed case and quotation marks. For instance, the statement `SELECT * FROM "Org"` also returns

```
SQL0204N "USERID.Org" is an undefined name. SQLSTATE=42704.
```

Possible Cause

The DB2 object name, including table and column names, is in mixed case. The object name must be specified in the mixed case, enclosed in double quotes, or DB2 will convert them to uppercase.

This affects all SQL statements, including `SELECT` statements as well as `CREATE TABLE` and `DROP TABLE` statements.

Many ODBC applications, including Lotus Approach, preserve the case of table and column names. If you specify the table name and the

column name in mixed case when the table is created in the ODBC application then the following SQL statement is sent to DB2:

```
CREATE TABLE "test1" ("col1" CHAR(5))
```

In this case DB2 preserves the lower case name because of the double quotation marks around the table and column names. Any attempt to access this table without enclosing the name with double quotation marks will fail.

Objects in the Control Center with mixed case names are enclosed with double quotation marks.

Action

If you are using an ODBC application to create DB2 tables then specify the table and column names in upper case. They can then be accessed either with or without the double quotation marks.

Enclose all object names in double quotation marks if you need to work with tables and columns in mixed case.

For instance, if the following table is created, both SELECT statements would succeed:

```
CREATE TABLE "User1"."Table1" ("column1" CHAR(5))
SELECT * FROM "User1"."Table1"
SELECT "column1" FROM "User1"."Table1"
```

Several Error Messages and Warnings are Seen When Running ODBC Applications

Symptom

Several dialog boxes appear with error messages or warnings when running ODBC (or CLI) applications such as Lotus Approach, Microsoft Access, or Powerbuilder. The title of each dialog box indicates which function returned the error or warning.

Possible Cause

These errors and warnings are a normal part of the communication that occurs between DB2 and the applications. They do not necessarily indicate a problem with DB2 or the application.

The errors and warnings are seen when the CLI/ODBC configuration keyword POPUPMESSAGE is set to 1 in the db2cli.ini file. This keyword is also accessible in the CLI/ODBC Settings notebook which is available from the Client Configuration Assistant. The "Error popup" parameter is located on the **Service** tab.

Action

The CLI/ODBC configuration keyword POPUPMESSAGE should be set to zero for production systems.

For more information on the POPUPMESSAGE keyword, and the db2cli.ini file, see "POPUPMESSAGE" in the "DB2 CLI/ODBC Configuration Keywords Listing" section of the *CLI Guide and Reference* manual.

To configure the DB2 CLI/ODBC driver using the DB2 Client Configuration Assistant (if available on your operating system):

1. Start the DB2 Client Configuration Assistant.
2. Select the DB2 database alias you want to configure.
3. Click **Properties** to open the Database Properties window.
4. Click **Settings** to open the CLI/ODBC Settings window.
5. Click **Advanced**.
6. Click **Service** tab and select the Error popup parameter.
7. Ensure that there is no check mark next to Popup error message in the Value section.

Cannot Perform Update

Some ODBC applications (such as Lotus Approach) require unique indexes to be defined for any tables that are updated. For these applications, you must create a primary key or a unique index on any table that you are trying to update, or you will receive an error that you cannot perform the operation.

For information, see the *Application Development Guide* and the DB2/Lotus Approach Web page at <http://www-4.ibm.com/software/data/db2/db2lotus/> Enter the question, "Why does Approach open my DB2 table in read only mode?"

Using SQLJ Applications

To run Java stored procedures or user-defined functions (UDFs) on an OS/2 server, you must install DB2 on an HPFS drive on the server. This permits the stored procedure or UDF ".class" files to be placed in the %DB2PATH%\function directory without truncating the filename extension.

To build SQLJ programs, you must turn off the just-in-time compiler feature of the Java Development Kit using the following command:

```
SET JAVA_COMPILER=NONE
```

Note: SQLJ source files may include an end-of-file character (EOF) causing the SQLJ translator 'sqlj' to fail with an exception similar to the following:

```
Cursor.sqlj:122.2: Error: Java Parsing. Encountered: \u001a  
Expected: > ; "abstract" ...; "public" ...; "interface" ...;  
"#sql" ...; ";" ...;  
Total 1 error.
```

To remove the offending EOF, open the SQLJ source file in a text editor that does not automatically append an EOF, modify the file by adding a blank line to the end of the file, and resave the file.

Using the db2ocat Tool

The DB2 ODBC Catalog Optimizer tool creates optimized catalog tables for ODBC, CLI, and JDBC access.

The tool results in fewer locks on base system catalog tables for the following reasons:

- Catalog query time is reduced.
- Amount of information returned by a call to an application is reduced.

Dealing with Performance Degradation

Many applications written using ODBC or DB2 CLI interfaces make heavy use of the system catalog. While this does not usually present a problem for databases with a small number of database objects, it can lead to performance problems when using these applications with larger DB2 databases. This performance degradation can be attributed to two main factors:

- the amount of information that has to be returned to the calling application
- the length of time that locks are held on the catalog tables

Performance Enhancement with the db2ocat Tool

The DB2 ODBC Catalog Optimizer Tool solves the problem of performance degradation by creating alternative catalog tables that are optimized for ODBC access by a particular application. This utility helps database administrators to identify the subset of catalog information that is needed for a particular application, and it creates an ODBC optimized catalog that the application can use. As a result, few locks are placed on the base system catalog tables, and catalog query times can be reduced substantially, together with the amount of data returned as a result of such queries.

Notes:

1. The db2ocat tool will not prevent an application from querying the system tables if it explicitly requests preparing an SQL statement.
2. The db2ocat tool only affects driver-generated queries against the system tables that use the CLI APIs like SQLTables, SQLColumns, and SQLFunctions. It will not prevent the optimizer from accessing system catalogs to generate an access plan.

Using db2ocat for Performance Improvement

To improve performance using the db2ocat utility, do the following:

1. Create ODBC optimized catalogs on the host
 - To create an ODBC optimized catalog, run the db2ocat utility on a Windows 95, Windows 98, or Windows NT workstation. The db2ocat

utility provides a Wizard that takes you through a process of naming the catalog and specifying the tables and stored procedures that should be available to a particular application. Once the tables and stored procedures are identified, it creates ten new tables on the target database server that comprise an ODBC optimized catalog of data objects available to the application. These ten tables have a qualifier (creator) that is the name of the ODBC optimized catalog.

- Repeat this step for each application that would benefit from an ODBC optimized catalog.
- 2. Configure each workstation to point to the right ODBC-optimized catalog(s).
 - Applications query the real DB2 system catalog to obtain metadata, by default.
 - IBM DB2 ODBC, CLI, and JDBC drivers provide a parameter that can be used to point applications to different source of metadata that is optimized for access by the ODBC, CLI, and JDBC applications.
 - The CLISCHEMA parameter, is set in the `db2cli.ini` file in the subdirectory where DB2 products are installed (typically `sql1ib`). This parameter applies only to the Data Source Name (DSN) for which it is set.
 - Any application that uses a DSN for which the CLISCHEMA parameter has been set will obtain metadata from the ODBC-optimized catalog named by the CLISCHEMA.
 - Set the CLISCHEMA parameter by manually editing the `db2cli.ini` file or by using the `db2ocat` tool at each of the end-user workstations. Because `db2cli.ini` is stored on each workstation, administrators should consider strategies for distributing `db2cli.ini` to end-users especially if the number of users is large.
- 3. Keep the ODBC optimized catalog(s) current.
 - ODBC optimized catalogs, created with the `db2ocat` utility, produce a separate copy of the data extracted from the DB2 system catalog (SYSIBM). It is important to have a procedure that will propagate future additions and changes in the system catalog to the ODBC optimized catalogs.
 - The `db2ocat` utility provides a manual procedure, requiring that when catalog changes occur, you start the `db2ocat` utility on a Windows 95, Windows 98, or Windows NT workstation and click **Refresh** for each ODBC optimized catalog that may be affected by the changes.
 - Another option for keeping ODBC optimized catalogs current is available to DB2 for OS/390 users that have IBM DataPropagator Relational Capture and Apply products installed on their host system. Use this DataPropagator option allows you to automatically replicate all changes in the real system catalog to all ODBC optimized catalogs.

Notes:

1. DB2 Universal Database provides several CLI/ODBC initialization keywords that can be used to limit the amount of data that is returned by the initial API calls during the "information gathering" stage after the database is first connected.
2. CLISHEMA keywords in the `db2cli.ini` file should be set to an ODBC catalog schema. When the DB2 CLI and ODBC catalog function calls are issued to obtain catalog information, the CLISHEMA option indicates an alternative schema, tables, and index set to be searched instead of the SYSIBM (or SYSTEM, QSYS2) schemas.
3. If you specify `CLISHEMA='SERGE'`, the internal CLI/ODBC API calls that normally reference system tables will reference the following user tables instead:
 - SERGE.TABLES
 - SERGE.COLUMNS
 - SERGE.SPECIALCOLUMNS
 - SERGE.STATISTICS

Stored Procedures

A *stored procedure* is a type of application stored on a database server. A *calling procedure* (or client application) is run on the database client that calls the stored procedure using an available API method.

The stored procedure, after being called or invoked by the client application, runs on the database server and returns any results to the client application.

The calling procedure and stored procedure can be written in any language supported by DB2 on your operating system. They do not need to be written in the same language.

To debug a stored procedure, run the stored procedure in debug mode. Testing of the stored procedure can be done separately from the calling application program. You can use the debugger supplied with your language compiler to debug local FENCED stored procedures as you would with other applications. See the compiler documentation for information on using the supplied debugger.

See the *Application Development Guide* in the "Resolving Problems" section of the "Stored Procedures" chapter for more information on troubleshooting stored procedures.

Error Received When Calling a Stored Procedure**Symptom**

Received an error (Error: Expected stored procedure name. Found: ?)

when attempting to call a stored procedure where the name of the stored procedure is in a host variable.

Possible Cause

The use of host variables for the procedure name is not an SQL92 standard. SQLJ is based on SQL92 standards and therefore does not support the use of host variables in the CALL statement for the name of the procedure.

Action

You cannot use host variables in the CALL statement for the name of the procedure.

Still Have Problems?

If you still have problems with your applications:

- See the DB2 Product and Service Technical Library on the Web, at <http://www-4.ibm.com/software/data/db2/library/> (Note that this information is in English only.)
- Contact the vendor of your application or compiler if you suspect a defect or limitation.
- Contact DB2 Customer Support with the following information:
 - A description of the events leading up to the problem.
 - A full description of the problem, including any SQL messages and associated reason codes.
 - The db2diag.log, and any trap or dump files for clients, servers, and DB2 Connect Enterprise Edition gateways. For more information, see “First Failure Data Capture” on page 157.
 - An CLI/ODBC/JDBC trace for CLI/ODBC/JDBC applications. (See the *CLI Guide and Reference* manual.)
 - Output from **db2level**.
 - The db2cli.ini file.
 - The jdbcerr.log file for JDBC-related problems.
 - A description of the type of application including the primary language it is written in, such as C, C++, Visual Basic, or Java.

Chapter 6. Troubleshooting DB2 Query Patroller

The DB2 Query Patroller system is a way to control and process query requests from multiple client workstations or PCs. The server part of the system is made up of many components:

- *Request server* to service received requests.
- *Cost analyzer* to determine the relative cost to execute a query. This relative cost is used with the job scheduler component.
- *Job scheduler* to schedule the query-based information that includes the relative cost from the cost analyzer.
- *Notifier* to scan periodically for completed queries and then notify the originator of the query with the query status.
- *Log monitor* to track error information in a log file.

The agent part of the system is made up of two components:

- *Node manager* to execute any work assigned to it by the job scheduler.
- *Executor* to run the SQL statements and the system commands.

There are also other parts to the DB2 Query Patroller system:

- *Command-line interface*, enabling you to monitor and control the system by entering commands directly at the prompt, or from within shell scripts and languages such as PERL, awk, or REXX.
- *Query Administrator*, which provides an interface to manage nodes, result destinations, data sources, job queues, and to create or delete profiles for users.
- *Tracker* utility, to report on database usage history for queries managed by the system.
- *Query Enabler*, which allows queries submitted through an ODBC-compliant query tool to be managed by the system.
- *Query Monitor*, which provides an interface to monitor submitted queries.

For details about the DB2 Query Patroller system, see the *DB2 Query Patroller Administration Guide* manual.

For specifics on troubleshooting DB2 Query Patroller servers and clients, see the appendices in that manual.

Still Have Problems?

If you still have problems with DB2 Query Patroller:

- See the DB2 Product and Service Technical Library on the Web, at <http://www-4.ibm.com/software/data/db2/library/> (Note that this information is in English only.)
- Contact the vendor of your application or compiler if you suspect a defect or limitation.
- Contact DB2 Customer Support with the following information:
 - A description of the events leading up to the problem.
 - A full description of the problem, including any SQL messages and associated reason codes.
 - DB2 Query Patroller log file called `syserr.log` which is found in the path specified by the DB2 profile variable `DQP_LOG`.
 - The `db2diag.log` file, and any trap or dump files for clients, servers, and DB2 Connect Enterprise Edition gateways. See “First Failure Data Capture” on page 157, for more information.
 - Output from **db2level**.

Chapter 7. Troubleshooting DB2 Universal Database Enterprise - Extended Edition

This chapter presents troubleshooting tips for users of DB2 Universal Database Enterprise – Extended Edition.

In addition to the information in this chapter, you should also see the following publications:

- *DB2 Enterprise - Extended Edition for UNIX Quick Beginnings*
- *DB2 Enterprise - Extended Edition for Windows Quick Beginnings*
- *The Administration Guide: Planning* for conceptual information

Note: The subject of partitioned databases is complex, with a diverse set of terminology. The term *node*, for instance, can mean various things:

- A database partition, or a part of a database that consists of its own user data, indexes, configuration files, and transaction logs
- A database partition server, or the machine that stores the database partition

Troubleshooting Tips

This section describes how to deal with some problems you could face when working with DB2 Enterprise - Extended Edition. It addresses the following topics:

- “DB2 Universal Database Enterprise - Extended Edition Will Not Install” on page 88
- “Determining a Failing Node” on page 88
- “Database Manager Will Not Start” on page 88
- “Cannot Create Database” on page 91
- “Commands Cannot Be Issued” on page 92
- “Problems Backing Up On Database Partitions” on page 92
- “Problems with LOAD and with Splitting and Loading Data” on page 93
- “Problems with Indexing Data” on page 96
- “Problems during Recovery” on page 96
- “Developing Applications in DB2 Enterprise - Extended Edition” on page 98
- “Severe Error Occurs on System” on page 99

Important: This section represents a small sampling of the information available from DB2 Customer Support. For a complete and up-to-date source

of DB2 information, see the DB2 Product and Service Technical Library at <http://www.ibm.com/software/data/db2/library/>.

DB2 Universal Database Enterprise - Extended Edition Will Not Install

Symptom

The DB2 Universal Database Enterprise - Extended Edition product will not install.

Action

Ensure that:

- [] The DB2 product directory on each node has enough free disk space.
- [] All prerequisite software products are installed on the system.
- [] The same version of the product is not already installed on the system.

Determining a Failing Node

To find out which node failed, check the db2diag.log on the coordinating node for the application. If a node returns an error or warning, the node number is identified in the SQLERRD(6) field of the SQLCA. This number corresponds to the node number in the db2nodes.cfg file. (If an SQL statement or API call is successful, the node number in the SQLERRD(6) field is that of the coordinator agent.)

Database Manager Will Not Start

db2start returns with an error message having an identifier of 'N'

Symptom

The **db2start** returns with an error message having an identifier of "N". For example, the error message SQL6048N is returned.

Action

Ensure that:

- [] Your user ID has SYSADM, SYSCTRL, or SYSMANT authority. See the *Administration Guide: Implementation* for details on these authorities.
- [] You have the same user ID, group identifier, and password set for the instance owners on all nodes.
- [] You can use the **rsh** command (on UNIX platforms) from the DB2 instance ID to all nodes defined in the db2nodes.cfg file. To do this, add an entry for the node in the \$HOME/.rhosts file or the hosts.equiv file. Ensure that the file permissions on these files allow read access for all users.

Once these tasks are completed, you will be able to use UNIX commands such as `rsh date` or the equivalent `db2_all 'date'` commands.

- [] The `db2nodes.cfg` file, located in the `sql1ib` subdirectory under the `$HOME` directory, contains the correct information for all nodes defined in the system. Ensure that the host names and net names are valid.
- [] The value for the `DB2INSTANCE` environment variable is the same for every node and it matches the instance name that you are trying to start.
- [] You have the appropriate permissions. Check the permissions of the `tmp` subdirectory under the `sql1ib` subdirectory and, if required, update them so that the instance ID has write permission for the directory. If the permissions are not correct, a problem may have occurred during installation and setup of the system.

db2start returns with a critical system error message having an identifier of 'C'

Symptom

The `db2start` returns with a critical system error message having an identifier of "C". For example, the error messages `SQL0902C` or `SQL1042C` are returned.

Action

Ensure that:

- [] The product was successfully installed.
- [] FCM communication is properly enabled. You must create an entry for each DB2 logical port in the `/etc/services` file. For details on FCM, see the *DB2 Enterprise - Extended Edition Quick Beginnings* manual for your operating system.

db2start returns with no output or hangs

Symptom

The `db2start` returns with no response.

Action

Ensure that:

- [] If you are using the high-performance switch (HPS) on RS/6000 SP systems, ensure that it is up on all nodes.
- [] Check the contents of the `db2diag.log` file for possible reasons for the `db2start` failure.

- [] Ensure that the \$HOME directory of the instance owner is NFS-mounted across all nodes, and that NFS is running. If the lockd and statd daemons are not running, db2start may hang. To start the daemons, execute **rc.nfs**, and ensure that this command is in the inittab subdirectory under the etc subdirectory.

Partition does not respond or hangs

Symptom

Some partitions do not respond or hang.

Action

Ensure that:

- [] Attempt to isolate the problem by answering the following questions:
 - Do all commands hang on all partitions?
 - Do all commands hang on a particular partition?

If you respond “yes” to either of these questions, you should check the hardware in your environment including the switch.

Then consider:

- Do all db2 commands hang on all partitions?
- Do all db2 commands hang on a particular partition?

There may be a process that you suspect is causing the problem. Use the process identifier of the suspect process with the `db2_call_stack` tool.

Use the `db2_call_stack` several minutes apart and application snapshots to see if the application is making any progress. See “Trap Files” on page 170 for more information on `db2_call_stack`.

Nodes do not start or are very slow

Symptom

Some nodes cannot be started, or run extremely slowly.

Action

Ensure that:

- [] You installed the DB2 Enterprise - Extended Edition product on all nodes.

Although you can install the product on an NFS-mounted file system in a RISC/6000 cluster, performance will be compromised. You should install the product on each node.

- [] All nodes are installed at the same level of DB2 Enterprise - Extended Edition.
- [] NFS is running and there are enough NFS daemon (nfsd) processes assigned on the machine where the home directory of the instance owner resides. The number of NFS daemon (nfsd) processes should be the number of block I/O daemon (biod) processes on a machine multiplied by the number of machines in the instance. For example, if you had a four machine system with ten biods, then there would be forty NFS daemon processes on the instance home directory.

Cannot Create Database

Symptom

A database cannot be created.

Action

Determine which node is experiencing the problem. The failing node number is stored in the SQLERRD(6) field of the SQLCA returned with the **CREATE DATABASE** command. Once you have determined the failing node, ensure that:

- [] Permission on the database directory path is correct.
The instance must have permission to write into the database directory path.
- [] You are creating the database on a path that exists on all nodes.
- [] The file systems are mounted.
The file systems for the database may have been unmounted. Mount all file systems again, and then try to create the database. You should check for unmounted file systems on all nodes.
- [] There is enough disk space.
You may have created a small test database and decided that you needed very little disk space. However, when a database is created, a certain amount of disk space is required for the default table spaces and default logs. See the *Administration Guide: Planning* for details.

Commands Cannot Be Issued

Symptom

DB2 or operating system commands are not recognized.

Action

Ensure that:

- [] Your DB2 Enterprise - Extended Edition system is properly installed.
- [] You have sufficient authority to run the commands.
- [] If the home file system of the instance owner ID is mounted over Ethernet or a high performance switch (on a RS/6000 SP machine), Ethernet or the switch are running.

To determine if this is the problem, log onto the system as root and try to access a file in the instance owner's home file system, or use the **netstat** command to determine the status of the high performance switch.

Problems Backing Up On Database Partitions

Symptom

Problems have been encountered while attempting to backup data on all of the database partitions.

Action

If you want to use the `db2_all` command to back up a list of database partitions, you must first back up the catalog node for the database. After taking this back up, you can back up the remaining database partitions. For example, after the catalog node is backed up, you can issue the following `db2_all` command to back up the remaining database partitions:

```
db2_all '<<-n< db2 backup db <database-alias>'
```

Where `n` is the node number of the catalog node.

Lost Database Partitions on Windows 2000

Symptom

Partitions could disappear and data be lost when creating too many partitions in Windows 2000 when using Dynamic Disk mode.

Action

Use Basic Disk mode instead. (Basic Disk mode is what Windows NT uses.)

Database Partition Size Rounding on Windows 2000

Symptom

When creating database partitions in Basic Disk mode, the partition size is rounded (either up or down) to the nearest multiple of cylinders (minus a single track in some situations). For example, if a cylinder has approximately 7 MB then the partition size will differ +/- 3.5 MB from the requested size.

Action

You should create database partitions that are 3.5 MB larger than what is specified for the table space.

Problems with LOAD and with Splitting and Loading Data

For more information on using the LOAD utility or on using the AutoLoader to split data and load it on a node, see the *Data Movement Utilities Guide and Reference*.

For more information on the `db2atld` command, type `db2atld -h` from the `misc` subdirectory under the `sql11b` subdirectory.

Data not loaded

Symptom

LOAD does not load data

Action

Ensure that:

[] LOAD was run by a user with the correct authority.

Users in the same group as the instance owner who have SYSADM or DBADM authority can execute LOAD.

[] The table or table space being loaded is not already being used by another application.

LOAD cannot share tables, so it will not execute until it can lock all required tables. Determine why the tables are already locked, and ensure that the locks are released.

[] LOAD was started on all nodes.

You must execute LOAD on all nodes for data to be loaded in parallel. If you have a script that sends a shell to all nodes, ensure that the LOAD execution is not serialized by the remote shell.

To check if LOAD failed on a node, look at the message files:

- If you issue the LOAD command directly, you can specify the name and location of a message file using its MESSAGES option.

- If you use the **db2atld** command, message files are located in the current working directory. There is a file for each node, and each file is named `load_log.nnn`, where *nnn* is the node number as specified in the `db2nodes.cfg` file.

[] The format of the data is correct.

See the *Command Reference* for more information on using correct data formats. Note that:

- Data cannot contain non-delimited blanks. LOAD processes non-delimited blanks in a delimited ASCII (DEL) file as null, and cannot load this data into a non-nullable column. You must use a string delimiter for blank columns.
For non-delimited data, use the null indicator to indicate columns with null values.
- Data must contain decimal points in the correct position.
For data being loaded from MVS systems, you must explicitly put the decimal points in the data before loading it. Otherwise, they are assumed to be at the end of the column.

All rows rejected

Symptom

LOAD rejected all rows.

Possible Cause

Column definitions are not correct.

Action

When using METHOD L, ensure that the data column specification is correct. Shifting columns can cause truncation errors or data that does not match the table column definition.

Rows not loaded

Symptom

LOAD completed but no rows are loaded.

Possible Cause

LOAD rejected all rows.

Action

Check the `db2load` file in the temporary directory where you ran **db2atld** to see if all the rows were rejected. If some rows were loaded, see the *Data Movement Utilities Guide and Reference*.

Interrupted load operation

Symptom

A load operation fails to complete.

Possible Cause

The load utility does not start because of a user error, such as a nonexistent data file or invalid column names.

Action

Restart the load operation from the last consistency point (using the RESTART option), or reload the entire table (using the REPLACE option). Specify the same parameters as in the previous invocation so that the utility can find the necessary temporary files. For additional information on LOAD, see the *Data Movement Utilities Guide and Reference*.

Data is not split

Symptom

The **db2atld** program completed successfully, but the data is not split.

Possible Cause

The **db2atld** program was set only to analyze the data.

Action

Check the AutoLoader configuration file and ensure that the *Mode* parameter is not set to analyze data. This option does not split data. It only analyzes the data and suggests a new partitioning map. See the *Data Movement Utilities Guide and Reference* for details.

Data is not split correctly

Symptom

The **db2atld** program completed successfully, but the data is not split correctly.

Action

Ensure that:

[] Binary data is not in a character column.

The **db2atld** program cannot detect binary values in a column of any data type, unless the BINARYNUMERICS or PACKEDDECIMAL options are specified on the load command.

[] The *SplitNodes* and *OutputNodes* parameters are set correctly.

Otherwise, output data files may not be correct. See the *Data Movement Utilities Guide and Reference* for details.

Data does not match the select count

Symptom

The data that is loaded does not match the select count.

Action

Ensure that:

- [] The data is split correctly by your conversion program.

The **db2atld** program creates a header file to prevent data from being loaded to the wrong node. The header information is checked by LOAD.

If you use a conversion program to convert binary columns to the character format, the conversion may not be the same as that done by the **db2atld** program. Data will be split for the correct node, but during the conversion data may be converted to a different value and may not be hashed the same way.

- [] The partitioning columns match the header.

If the data is split using one set of partitioning keys, and is loaded into a table that was partitioned on other columns, the LOAD operation will fail. Partition column information is in the header of the split data file. If you manually alter the header, this compromises the validation measure built into the AutoLoader.

Problems with Indexing Data

Symptom

The CREATE INDEX statement fails or hangs.

Possible Cause

The table space is full, or there is insufficient log space.

Action

Ensure that there is sufficient disk space for the index pages and for sorting; and that there is sufficient log space.

For information on calculating disk size and on determining log size, see the *Administration Guide: Planning*.

Problems during Recovery

Recovery appears suspended

Symptom

A connection appears suspended.

Possible Cause

The database was restarted and is still recovering.

When crash recovery is in progress, it can take a long time to recover the database from a failure. This occurs when operations that require a large volume of logging activities were running before the database crashed. This is a normal situation.

Action

Check the db2diag.log file on all database partitions to see if crash recovery is completed. Entries in the db2diag.log will indicate when recovery is started, and when it is finished.

Check the log file directory on all database partitions for I/O activity. If there is I/O activity, then crash recovery is proceeding.

SQL1034C, SQL1042C, or SQL1072C received

Symptom

The RESTART DATABASE command returns SQL1034C, SQL1042C, or SQL1072C.

Possible Cause

InterProcess Communications (IPC) resources from the previous activity are preventing crash recovery from succeeding.

Action

Bring the database instance down using a db2stop. Check for DB2 processes on all partitions by issuing:

```
ps -ef | grep <instance name>
```

Check for IPC resources on all partitions by issuing:

```
ipcs | grep <instance name>
```

If either processes or IPC resources exist after the db2stop, issue a db2_kill to delete them. After issuing the db2_kill, check for processes and IPC resources again. If they still exist, they will have to be removed individually using the kill command for processes, and the ipcrm command for IPC resources. The ipclean command can also be used before running ipcrm on each resource.

Once all processes and IPC resources are cleared, log onto the catalog node and issue:

```
db2 restart database <database-alias>
```

Next, issue a restart to the rest of the partitions or nodes:

```
db2_all "<<-n< db2 restart database <database-alias>"
```

Where n is the node number of the catalog node.

If the SQL1034C, SQL1042C, or SQL1072C error messages persist, contact DB2 Customer Support. You will be asked for the following information:

- The db2diag.log from all database partitions
- The database partition(s) where RESTART DATABASE is failing
- A trace of the RESTART from the database partition(s) that are failing
- The database configuration from all nodes
- The database manager configuration from all nodes
- The db2nodes.cfg file
- The SQLOGCTL.LFH file from the failing database partition(s)
- The active log files from the failing database partition(s).

SQL1061W received

Symptom

The SQL1061W message is received during recovery.

Possible Cause

There are indoubt transactions that cannot complete, typically because one or more nodes cannot be started. The database is recovered and open for user connections, but the indoubt transactions are taking up memory and other resources.

Action

Check the db2diag.log file of the coordinator node to see if all nodes are started.

Resolve the indoubt transactions as soon as possible. For information, see the discussion on heuristics in the *Administration Guide: Planning*.

Developing Applications in DB2 Enterprise - Extended Edition

In a partitioned database environment, DB2 breaks up SQL statements into subsections, each of which is processed on the node that contains the relevant data. As a result, an error may occur on a node that does not have access to the application.

When developing applications for multiple nodes, consider the following:

- Errors are reported in the SQLCA.

If a node returns an error or warning, a node number that maps to the db2nodes.cfg file is placed in the SQLERRD(6) field of the SQLCA. If an SQL statement or API call is successful, the node number in the SQLERRD(6) field is that of the coordinator agent.

- Indoubt transactions may be left in the database if an application ends abnormally during 2-phase commit. An indoubt transaction pertains to global transactions when some phases complete successfully, but the system

fails before all phases can complete. The database is left in an inconsistent state. For information on handling indoubt transactions, see the *Administration Guide: Planning*.

- An application may receive a subsequent error or warning after the problem that caused the first error or warning is corrected. Only one severe error is reported to the SQLCA at a time and the error is not overwritten. Severe errors and deadlock errors are given higher priority because they require immediate action by the coordinator agent.

Applications appear to be hung or very slow

Symptom

In a partitioned database environment, SQL statements from an application are processed on different partitions. An application is not complete until all of the SQL statements on all of the partitions are complete. If one or more partitions does not complete, the application does not complete.

Action

Ensure that:

- [] You should document the particular application(s) that appear to be hung or slow. Determining which application and partition is causing the delay can be done as follows:
 - Check the application status on all partitions since the applications could be in a “lock wait”. This can be done by using the LIST APPLICATIONS command to check the application status on all partitions.
 - Check the output from the GET SNAPSHOT command for rows processed or logged.
 - Check the configuration parameter settings that affect the memory being used since this could lead to performance problems. The *Administration Guide: Performance* should also be of assistance.

Please see *Command Reference* for more information on the LIST APPLICATIONS and GET SNAPSHOT commands.

Severe Error Occurs on System

There are two distinct types of severe error:

- A DB2 process is killed because of a programming exception.

In this case, the database manager is immediately terminated at the node and active units of work are not rolled back. When other nodes detect the failure, depending on their relationship with the failing node they will attempt to recover as follows:

- If the failing database partition was the catalog node of the database, the entire database is brought down at all other nodes because all agents are forced off the node.

In this situation, you must restart all the database partitions with `db2start` and then go through the crash recovery procedure by running **RESTART DATABASE** for all databases on that partition. Crash recovery may be necessary on some of the subordinate nodes as well. See the *Command Reference* for more information.

- If the failing database partition was the coordinator node for the application, all subagents running at the node on behalf of the coordinator node are forced off the database partition and the active unit of work is rolled back.

In this situation, you must restart the coordinator node with `db2start` and then go through the crash recovery procedure by running **RESTART DATABASE** for all databases on that partition. See the *Command Reference* for more information.

- If the failing node was a subordinate node, any coordinator agent which has an active unit of work that includes the failed node rolls back the unit of work if it was not already in the COMMIT phase. If it is in the COMMIT phase, the SQL code -279 is returned to indicate that the transaction is in doubt.
- A problem caused the database to be marked as inconsistent.

Any attempt to access the database returns SQL code -1034 (SQL state 58031) or SQL code -1015 (SQL state 55032).

In this case, all coordinator agents and subagents are forced off the database. These agents then roll back the current unit of work and disconnect from the database.

Before any application on any other node can access the database at this node, you must put the database in a consistent state by running **RESTART DATABASE** against it.

The severe error SQL code -1224 (SQL state 55032) can occur for a variety of reasons. If you receive this message, check the SQLCA to determine which node failed, and then the `db2diag.log` file for additional details. For information, see "Determining a Failing Node" on page 88.

Note: In the event of a severe error where multiple machines were involved, you may find diagnostic information in their `syslog` files rather than in the `db2diag.log` file, if they were not able to obtain an NFS lock on the `db2diag.log` file.

If you know the failing partition or node, then you can use the **db2start nodenum** command. For example, if "3" is the failing node, then:

```
db2start nodenum 3
```

If the db2start fails for the individual partition or node, then you will need to run a db2stop and then a db2start on all partitions. If no nodenum is specified, all partitions defined in the node configuration file are started.

Restart the database on those partitions where it is inconsistent. To see which partitions are consistent, look at the database_consistent informational database parameter on each partition. For more information on using the RESTART DATABASE command, see the *Command Reference* manual.

Additional Enterprise - Extended Edition Considerations

There are other considerations that may assist you with problem determination and troubleshooting in a partitioned database environment. The following sections cover these considerations.

First Failure Data Capture

Information on FFDC specific to the EEE environment is presented here.

In an EEE environment, if your DIAGPATH is NFS-mounted or shared then a noticeable slow down will occur with the higher diagnostic levels chosen. Level 3 should be the highest level used in this environment. The only time level 4 should be considered for use is during:

- Initial installation or configuration
- Implementing a configuration change
- Times when errors are recurring or when many different errors are encountered.

In an EEE environment, since the *sqllib* directory is mounted on all systems there is one db2diag.log file for all partitions or nodes. By changing the DIAGPATH database manager configuration parameter for all partitions to point to a local path, a separate db2diag.log file is kept for each partition. Each db2diag.log file then has log information specific to each partition making each log file easier to read. However, should all of the db2diag.log file information need to be seen, then gathering and integrating all the data from all the partitions can be difficult.

DB2 Trace Facility

Information on the DB2 Trace Facility specific to the EEE environment is presented here.

In an EEE environment, you may need to take a trace on more than one partition to determine the error. In a multiple logical node (MLN) environment, only one trace per physical node is required. Determining which partition or node on which to run the DB2 Trace Facility (db2trc) is done using SQLCA results following an error. In the SQLCA format, SQLERRD(6) presents the node number.

Other Methods

The `db2_call_stack` command can be used to assist in the identification of loops or hangs in the EEE environment. The `db2_call_stack` command sends a “terminate” signal to every DB2 process on every node recorded in the `db2nodes.cfg` file.

The `db2_call_stack` command should be run several times with some time in between to determine if there is a loop or hang; or, if the work is being done at all. See “Trap Files” on page 170 for more information on `db2_call_stack`.

The `DB2MEMDBG` registry variable enables the tracking of memory allocation including the allocations of memory sets, memory pools, and memory blocks. It can be used to help identify memory corruption or memory leakage conditions. Memory corruption examples are:

- Allocating fewer bytes than is required
- Accessing a data structure after it was freed
- Using the wrong data pointer to access a data structure.

Memory leakage examples are:

- A slow resource leakage
- An out-of-pool memory.

The process for using this registry variable is:

1. (Following a suspected memory problem, contact DB2 Customer Support for a set of options to use with this registry variable.)
2. `db2stop`
3. `db2set DB2MEMDBG=<options>`
4. `db2start`

The output from this procedure is placed in the `db2dump` subdirectory of the `sql1ib` directory. The `memdbg.log` file contains log entries for all processes that are enabled for memory debugging. The `p_<PID>.mem` files contain per process memory debug information including corruption information.

When using `LIST APPLICATIONS` to collect data about a problem, remember to run this command on all partitions in your EEE environment. Running `LIST APPLICATIONS` from the catalog node gives information about all coordinator nodes.

When attempting to restart your database, always restart the catalog partition first and by itself. Once restarted, attempt a parallel restart of all the remaining partitions. If errors still occur during the parallel restart, then attempt to restart the partitions one at a time.

General Instructions When in a Bad State

If after checking your environmental status and you are sure that your system is completely hung, you should, with caution, consider the following process:

1. Use `db2_kill` to bring down the database.
2. Issue a `db2start` to bring up the database manager instance.
3. Issue a `db2 restart database` at the catalog partition/node to begin the process of bringing up the database.
4. Issue a `db2 restart database` at all the other partitions/nodes to complete the process of bringing up the database.

If after checking your environmental status and you determine that only one or two partitions/nodes are not working in your EEE environment, with caution, consider the following process:

1. Issue a `db2start nodenum n` to bring up the hung partition/node. Repeat this command where “n” is the node number of the partition that was hung until all of the partitions are up.
2. Issue a `db2start database` to complete the process of bringing up the database.

Hierarchy of Shutdowns

Note: You should be cautious when proceeding to the latter commands to shutdown your database environment. Using the latter commands can remove the problem evidence so that DB2 Customer Support may not be able to determine the cause of your problem. Without determining the current problem, there is little or no chance for a preventative approach to be implemented in your environment. Your environment may then be susceptible to the same problem occurring again.

Here is the list of ways to shutdown your database environment from least to most forceful:

- `db2stop`
- `db2stop force`
- `db2_kill`
- `kill -9` for processes, and `ipcrm` for IPC resources.

You should only use the most forceful commands under the direction of DB2 Customer Support.

Still Have Problems?

If you still have problems with your DB2 Enterprise - Extended Edition system, contact DB2 Customer Support with the following information:

- A description of the events leading up to the problem.

- A full description of the problem, including any SQL messages and associated reason codes.
- The `db2diag.log` file, and any trap or dump files, for each partition.
- Any `syslog` files.
- Any core files.
- The contents of the `db2nodes.cfg` file.
- The database configuration from each partition.
- The database manager configuration from each partition.
- The identification of any failing partitions.
- The `sqllogctl.lfh` file from each failing partition.
- The active log files from each failing partition.
- Output from **db2level**.
- The output from **errpt -a**.

Chapter 8. Troubleshooting DB2 Data Links Manager

This chapter provides information on the following DB2 Data Links Manager topics:

- “DB2 Data Links Manager Considerations”
- “DB2 Data Links Manager” on page 106
- “Data Access Token” on page 106
- “Recovery Data Links Manager Environment” on page 107
- “Backup Data Links Manager Environment” on page 109
- “Export Operation” on page 110
- “Import Operation” on page 110
- “Load Considerations” on page 111
- “DB2 Data Links Manager for Windows NT” on page 112.

DB2 Data Links Manager Considerations

Before using DB2 Data Links Manager, you should confirm that your environment supports DLFM. The following is a summary of the conditions in which the DB2 Data Links Manager can function.

- There is a maximum of 256 unique combinations of DATALINK columns and DLFMs. For example, the table can have a maximum of 128 DATALINK columns if the number of DLFMs configured is two.
- Data Links Manager cannot be exchanged with a DRDA server.
- Data Links Manager can be installed on systems with:
 - DB2 UDB EE (AIX)
 - DB2 PE, WE, EE (NT).
- The Data Links Manager *cannot* be used with:
 - DB2 Enterprise Extended Edition (EEE) for AIX.
 - DB2 Enterprise Extended Edition (EEE) for Windows NT.
- DATALINK values cannot point to a DB2 EEE system.
- The DATALINK data type cannot be used on DB2 EEE systems to reference files on a DB2 Data Links Manager.
- Tables with DATALINK columns can be created on any DB2 UDB server.
- DB2 Server and Data Links Manager have to be on the same release level.
 - DB2 V6 — Data Links Manager V6 support only.
 - DB2 V5.2 — Data Links Manager V5.2 support only.

- DATALINK columns cannot be part of a unique index, primary key, or foreign key.

DB2 Data Links Manager

The DB2 Data Links Manager consists of the following two components:

Data Links File Manager

Data Links File Manager registers all files on a particular Data Links Manager Server that are referred by DATALINK columns in DB2 databases. This component tracks the meta data (the administration information) for the link files. It also tracks previously linked files for point-in-time roll-forward recovery.

Data Links Filesystem Filter

Data Links Filesystem Filter filters the file system commands to ensure that registered files are not deleted or renamed. Optionally, it also filters commands to ensure that proper access authority exists. It enhances the file system of the operating system so that no file API can corrupt the integrity between the data on the DB2 Server and the external files managed by Data Links Manager.

Note: The DLFF must be applied to all file systems that are referenced by DATALINK columns.

There are some important considerations when using the DB2 Data Links Manager:

- The DB2 Database Server must have exactly the same fix pack level as the Data Links File Manager (DLFM) components.
- If the DB2 Server and the DLM are installed on the same machine, the database that is used by the Data Links File Manager (DLFM) must reside in a different instance than the DB2 Database Server Databases (the databases that contain the DATALINK columns used by applications). For example, the instance name filling the role of the DB2 Server must not be "dlfm" because it is used by Data Links File Manager.

Data Access Token

File access control is provided by a combination of DB2 and the Data Links Manager. DB2 assigns an access token dynamically, using a SELECT statement on the DATALINK column and embeds the result in the file path name.

There are several provisions for file access control to operate correctly:

- File access control is provided only if the READ PERMISSION DB option is specified on the DATALINK column when the table is created.
- If a user can query over certain meta data (such as an SQL view), they can access corresponding files.
- Any file system API or command can be used to read the file.

- The rule can be any SQL expression.
- Generation of the access token is shared secretly between DB2 and Data Links Filesystem Filter (DLFF).
- To be valid, an access token must be generated and used within a specified time interval as defined in the Data Links Access Token Expiry Interval (*dl_expint*) database configuration parameter.
- For each (select) access, a new token is generated and remains valid for the time specified by *dl_expint*.
- On most file access control mechanisms of an operating system, it is not possible to grant access based on rules for each file.
- Web addresses with embedded access tokens must be used by the application to access the files. Any attempt to open, read, or otherwise manipulate a file using Web addresses with the access token results in an access violation.

Note: For more information on Data access and permissions, see *DB2 Data Links Manager Quick Beginnings*.

Recovery Data Links Manager Environment

This section describes Data Links Manager recovery situations.

DB2 database recovery:

- Restore database backup and roll forward end of logs
- Execute the **RECONCILE** command on all tables which have DATALINK columns.

DLFS file systems recovery:

- Restore the DLFS files systems
- Execute the `db2_reconcile_aid` utility for the databases which see the file system
- The **db2_reconcile_aid** command provides a mechanism for checking and running RECONCILE.

The following table shows possible situations that require recovery operations.

Environment	Conditions	Action and Resolution
The DB2 databases on the DB2 Database Server are lost.	DB2 backup and log files are available.	On the DB2 Database Server: <ul style="list-style-type: none"> • Restore the database(s) and roll forward to end of logs. • Reconcile all table that have DATALINK values in that database(s).
DLFM_DB on the Data Links Server is lost.	Backup and log files are available.	On the Data Links Server: <ul style="list-style-type: none"> • Restore the database(s) and roll forward to end of logs. • Restore database DLFM_DB and roll forward to end of log.
Recover from a disk crash on the DLFM Server.	DB2 applications interacting with the DLFM can hang.	Use the FORCE APPLICATION command to force applications off the system.
Recover from a machine crash on the DLFM Server.	DB2 applications interacting with the DLFM can hang.	The affected DB2 Data Links Managers should be shut down by using the dlfm shutdown command. These can then be brought up again.
Recover from a disk or machine crash on the DLFM Server.	Applications can hang.	On the Data Links Server: <ul style="list-style-type: none"> • Restore the DLFS file system. On the DB2 Database Server: <ul style="list-style-type: none"> • Perform a reconcile on all tables, containing the damaged disk, by invoking the command, db2_reconcile_aid. This will check and run reconcile on tables that are potentially inconsistent with the linked file on the File Manager Server after a disk failure.

Recover the DLFM backup directory.		Restore the DLFM backup directory.
------------------------------------	--	------------------------------------

Files in a linked state, having the RECOVERY NO option set:			
Permissions	File Status on File System	Action	Result
	Not Found		DATALINK value will be set to NULL.
READ and WRITE PERMISSION FS	Found	No additional checks will be conducted to validate the correctness of the file.	
WRITE PERMISSION BLOCKED	Found	Its modification time and file size will be checked.	If there is a mismatch in the values, the DATALINK value will be set to NULL.

Files in a linked state, having the RECOVERY YES option set			
Permissions	File Status on File System	Action	Result
	Found	The file modification time is less than the file modification time at link time.	The correct version of the file will be restored from the archive server.
		The file modification time is greater than the file modification time at link time.	The DATALINK value will be set to NULL.

Backup Data Links Manager Environment

To perform a backup of the entire Data Links File Manager environment, consider the following:

- Ensure all Data Links Servers are up and running (unless you have not specified the NO LINK CONTROL option in the DATALINK definition).
- Back up the databases on the DB2 Database Server(s).
- Back up the file systems used by the Data Links Manager.

- File system(s) need to be unmounted, backed up (via the operating system), and then mounted again.
- Back up the DLFM Backup directory, which holds:
 - Images of the DLFM_DB database
 - Copies of the linked files
 - All updates if the RECOVERY OPTION is set to YES in the DATALINK column. This provides point-in-time roll-forward recovery (if you set DLFM_BACKUP_DIR_NAME=directoryname, not ADSM)

Note: The preceding discussion serves only as a general reference to this topic. Because of the complexity of this operation, you should refer to the appropriate sections of the *DB2 Data Links Manager Quick Beginnings* manuals before performing a backup or restore operation.

Export Operation

This operation, using DLFM_EXPORT is used on the Data Links server to export two sets of data:

- DATALINK data
- Linked files.

A successful export on a table with DATALINK columns produces:

- An export file for the parametric data
- Control files (one for each involved server) that list the referenced files from the exported rows.

Notes:

1. For a successful DLFM_EXPORT operation to occur, root authority is required. There may be files that need to be archived that cannot be accessed by the DLFM administrator.
2. The DB2 Data Links Manager does not have to be running in order for DLFM_EXPORT to run.

Import Operation

Steps to import data, using the utility DLFM_IMPORT, are included in *DB2 Data Links Manager Quick Beginnings*. However, there are particular conditions in which data is imported successfully and it is helpful to review these conditions during troubleshooting:

- The import utility must be used whether the archived files are being restored on the same or a different file server.
- Root authority is required as a user may decide to restore the archived files on a different file server, which may not have the same directory structure and user IDs as the file server on which the DLFM_EXPORT utility was run.

- The files exported will be owned by root in case some of the user IDs do not exist on the new machine.
- The administrator must ensure that all files have the right permissions.
- If files are imported to different Data Links Manager server or files system names or directories, the appropriate steps on the Data Link servers must be performed to register file systems and DB2 Database Servers.

On the DB2 Database Servers the appropriate steps for configuring the DB2 Database Servers must be performed if the tables are imported on another DB2 Database Server.

- If the server name or file path has changed, you can use the `DL_URL_REPLACE_PREFIX` prefix phrase in the import command which changes all paths (in the URL) in the DATALINK value.
- When the import utility executes on the target database, data related to DATALINK columns is loaded into the underlying tables using SQL INSERT.
- During the INSERT, the DATALINK column is linked to the file on the appropriate file server.

Load Considerations

Before invoking the LOAD utility, it should be noted that some options are not supported:

- The COPY option is not supported for tables with DATALINK columns
- The REPLACE option is not supported for tables with DATALINK columns
- The NONRECOVERABLE clause should not be used when DATALINK columns with the FILE LINK CONTROL attribute are present in, or being added to, the table.

If you are loading data to a table with a DATALINK column that is defined as FILE LINK CONTROL, perform the following steps before invoking the LOAD utility:

- Ensure that the DB2 Data Links Manager is installed on the file servers that will be referred to by the DATALINK column values.
- Ensure that the database is registered with the DLFM.
- Copy to the appropriate file servers all files that will be inserted as DATALINK values.
- Define the prefix name (or names) to the DLFM on the file servers. For more information, see the description on registering databases with the DLFM in the *DB2 Data Links Manager Quick Beginnings*.
- If intending to see a new file server in an input file, add the file server name (or names) to the DB2 file server configuration file.

DB2 Data Links Manager for Windows NT

If you install DB2 Data Links Manager on a clean machine, the installation program will also install a compact version of DB2 Universal Database Personal Edition for the DLFM.

With this DB2 Universal Database, there is no way to additionally install the DB2 Server on the same machine. However, if before installing DB2 Data Links Manager on the system there is already a DB2 Universal Database Personal Edition, Workgroup Edition, or Enterprise Edition, DLFM will use the existing DB2 Universal Database product.

Still Have Problems?

If you still have problems with Data Links File Manager:

- See the DB2 Product and Service Technical Library on the World Wide Web, at <http://www-4.ibm.com/software/data/db2/library/>. (Note that this information is in English only.)
- See “Appendix C. Using the DB2 Library” on page 277 for information about the location of release notes for this product.
- Contact the vendor of your application or compiler if you suspect a defect or limitation.
- Contact DB2 Customer Support with the following information:
 - A description of the events leading up to the problem.
 - A full description of the problem, including any SQL messages and associated reason codes.
 - The db2diag.log, and any trap or dump files, for clients, servers, and DB2 Connect Enterprise Edition gateways. For more information, see “First Failure Data Capture” on page 157.
 - Output from **db2level**.

Chapter 9. Troubleshooting Replication

Replication is a process of maintaining a defined set of data in more than one location. It involves copying designated changes from one location (called a *source*) to another (called a *target*), and synchronizing the data in both locations. The source and the target can be in logical servers on the same machines or on different machines in a distributed network.

DB2 DataPropagator provides the following facilities when troubleshooting problems:

- The Replication Analyzer which analyzes replication control tables, change data tables, target tables, and target table indexes for accuracy, consistency, and efficiency. Findings from the analysis are reported in a HTML file.
- Error messages and SQL states for the DB2 Control Center, the Capture program, and the Apply program
- The Apply program trail table (ASN.IBMSNAP_APPLYTRAIL on the control server database), log file (*.APP) and trace file (*.TRC). The `trcflow` or `trcerr` parameters can be used with the **asnapply** command to produce a trace.
- The Capture program trace table, log file and trace file. The `trace` parameter can be used with the **asnccp** command to produce a trace.
- The IBM Replication control tables and files.

When testing or troubleshooting replication problems, you should use these cautionary practices:

- Always work on replication scenarios in a test environment.
- Stop other replication activity while gathering information about a problem to reduce the volume of data that surrounds the actual problem.

There are two areas where problems could occur:

- During the administration of the replication environment using the Control Center or the DB2 DataJoiner Replication Administration (DJRA) tool.
- During the running of the Capture and Apply programs. For example, to get more information on ASNnnnnn errors, run `db2 ? ASNnnnnn` where nnnnn is the message number.

Details about the facilities available to identify and solve replication problems and Apply and Capture messages are found in the *Replication Guide and Reference* manual.

DB2 DataPropagator

DB2 DataPropagator allows the management of multiple copies of redundant data. To determine which log file contains the log record identified by a given log sequence number, use the **db2flsn** (Find Log Sequence Number) command.

The command **db2flsn -q input_LSN** specifies that only the log file name be printed. No error or warning messages will be printed, and status can only be determined through the return code.

Valid error codes are:

- —100 Invalid input
- —101 Cannot open LFH file
- —102 Failed to read LFH file
- —103 Invalid LFH
- —104 Database is not recoverable
- —105 LSN too big
- —500 Logical error
- 0 Successful execution
- 99 Warning: the result is based on the last known log file size. For example:

```
db2flsn 000000BE0030
```

```
Warning: The result is based on the last known log file size.
```

```
The last known log file size is 23 4K pages, starting from log extend 2.
```

```
Given LSN is contained in log file S0000001.LOG
```

Using the Log Header Control File with db2flsn

The `input_LSN` file contains a 12-byte string that represents the internal (6-byte) hexadecimal value for the log sequence number. There may be leading zeros to completely fill the 12-byte string.

The log header control file `sqllogctl.lfh` must reside in the current directory. Since this file is located in the database directory, the tool can be run from the database directory, or the control file can be copied to the directory from which the tool will be run.

The tool uses the `logfilsiz` database configuration parameter. DB2 records the three most recent values for this parameter and the first log file that is created with each `logfilsiz` value; this enables the tool to work correctly when `logfilsiz` changes.

If the specified LSN predates the earliest recorded value of `logfilsiz`, the tool uses this value, and returns a warning.

The tool can be used with database manager, prior to DB2 Universal Database Version 5.2.

For more information see the **sqlurlog** (Asynchronous Read Log) API in the *Administrative API Reference*.

Example of Using the db2flsn Command and the db2diag.log File

You notice that in your db2diag.log file a restart has taken place:

```
1999-04-06-11.51.31.237000 Instance:DB2 Node:000
PID:254(DB2SYSCS.EXE) TID:247 Appid:*LOCAL.DB2.990406154954
recovery_manager sqlpresr Probe:170 Database:SAMPLE
DIA3909W Crash recovery completed. Next LSN is "0000003E800C".
```

You can use the **db2flsn** command to investigate the relevant log file:

```
D:\DB2\NODE0000\SQL00001>db2flsn 0000003E800C
Given LSN is contained in log file S0000000.LOG
```

DB2 DataPropagator Enhancements

DB2 DataPropagator supports copying a BIGINT source to a BIGINT target between DB2 Universal Database products. For target servers that do not support BIGINT, a BIGINT column can still be copied but the target column is defined as DECIMAL.

DB2 DataPropagator supports copying large objects between DB2 Universal Database OS/390s, between DB2 Universal Database products, and between DB2 Universal Database OS/390 and DB2 Universal Database.

LOB Restrictions and Requirements

Copying LOBs between DB2 OS/390 and DB2 UDB

DB2 Connect Version 6 is required.

DB2 Universal Database does not support the role of an LOB Application Server To copy large objects from a Universal Database to DB2 OS/390, the push technique must be used.

LOB column data is not captured in the CD table

An indication of LOB column data change is captured. The actual LOB data is copied by APPLY directly from the source table to the target for those LOB columns that are changed. For this reason, the user table must have the same primary key columns as the target table. The LOB change indicator is a nullable CHAR(1) column in the source table, so it requires more than one indication column in the CD table. The value of an LOB indication column is significant for update rows only. If the LOB column data was changed in an UPDATE operation, the indicator contains U. In all other cases, it is set to null.

Before_image copy is not supported for LOB columns

LOB columns are copied only to read-only targets

The target type may not be *replica* or *replica2*.

Note: A *replica* is a target table that can be updated. Changes to this table are replicated back to the replication source table. This table is used in an update-anywhere scenario. A *replica2* is a type of update-anywhere replica without transaction semantics. Conflicts are detected row by row, not transaction by transaction as for replicas.

Target must be condensed, because only the current value is available

The apply program always copies the most current version of a LOB column directly from the source table (not the CD table), even if that column is more current than other columns in the target table. Synchronization anomalies may occur, because the LOB column value is copied from the source table and may be much more current, possibly inconsistent, with the other column values being applied.

For Capture to detect LOB changes, the user table must be created or altered specifying DATA CAPTURE CHANGES.

The LOGGED or NOT LOGGED option in the CREATE TABLE statement for Universal Database does not affect capturing LOB changes. The LOG YES/NO option in the CREATE LOB TABLESPACE statement for DB2 OS/390 does not affect capturing LOB changes. DB2 OS/390 and DB2 Universal Database use different syntaxes for supporting LOBs. In DB2 Universal Database, the proper statements are CREATE TABLE, CREATE LOB TABLESPACE, and CREATE AUXILIARY TABLE.

Still Have Problems?

If you still have problems with replication:

- See the DB2 Product and Service Technical Library on the Web, at <http://www-4.ibm.com/software/data/db2/library/> (Note that this information is in English only.)
- Contact the vendor of your application or compiler if you suspect a defect or limitation.
- Contact DB2 Customer Support with the following information:
 - A description of the events leading up to the problem.
 - A full description of the problem, including any SQL messages and associated reason codes.
 - The db2diag.log file, and any trap or dump files, for clients, servers, and DB2 Connect Enterprise Edition gateways. For more information, see “First Failure Data Capture” on page 157.
 - Output from **db2level**.

Chapter 10. Troubleshooting the Data Warehouse Center

This chapter outlines problems you might experience while using the Data Warehouse Center. Problems are listed by type with specific categories, such as installation issues. Each problem outline contains a solution that you can use to resolve the problem.

Prerequisite or related product issues

This section details errors that occur with products that are installed and configured with the Data Warehouse Center.

A DataJoiner Version 2 CREATE SERVER MAPPING statement for an Oracle data source fails

DataJoiner Version 2 CREATE SERVER MAPPING SQL statements fail on a workstation that has the Data Warehouse Center installed.

You might receive the message `DB2SYSC.EXE -- Ordinal not found` or one that implies that the SERVER MAPPING failed because it could not load the `SQL*NET` data access module.

Cause

The CREATE SERVER MAPPING statement fails because DataJoiner accesses the `ORA73.DLL` file that is provided with the Data Warehouse Center instead of the `ORA73.DLL` file in the `ORANT\BIN` directory. This condition occurs because DataJoiner looks for the Oracle DLL through the Path system variable. The system path is updated during a Data Warehouse Center installation so that it includes a directory containing a `ORA73.DLL` file that is provided by the Data Warehouse Center.

Action

Update the Path system variable. Ensure that the directory that contains the Oracle `ORA73.DLL` file (in `ORANT\BIN`) is specified first in the path (before the `VWSWIN\IWH\ODBC32` directory).

Unable to connect to a DB2 database from a Windows NT or Windows 2000 agent site

You are unable to access a DB2 source database, target database, or warehouse control database from a Windows NT or Windows 2000 agent site.

Cause

There are several reasons why you might have this problem:

- Your DB2 client is configured incorrectly.

- Your warehouse control database is not registered as an ODBC system DSN on your warehouse server and on your Data Warehouse Center administrative interface workstation.
- Your source database and target database are not registered as ODBC system DSNs on your agent site workstation.
- Your TCP/IP service is not started, or TCP/IP connectivity problems exist.

Action

Try the following solutions:

- Verify that the workstation name is entered as NNAME in your DB2 client database manager configuration.
- Ensure that the node and database are cataloged correctly, as follows:

Verify that the DB2 client catalog:

node
 host name
 service name

 database name

Matches the host DB2 catalog:

NNAME
 TCP/IP host name
 SVCENAME of the database manager
 host configuration
 database name

- Ensure that the DB2 security service is started correctly and that the DB2 service is started correctly. Check the DB2DIAG.LOG file in your DB2 installed directory for any startup errors.
- Verify that your warehouse control database is registered as an ODBC system DSN on your warehouse server workstation and all your Data Warehouse Center administrative interface workstations.
- Verify that your source database and target database are registered as ODBC system DSNs on your agent sites.
- Ensure that the TCP/IP service is started at both the client and host. To do this, check the Services application of the Control Panel program group on both workstations.
- Verify TCP/IP connectivity. Ensure that all cables are connected, that the TCP/IP service is started on both workstations, and that the LAN is functioning.

If these configurations are correct, ping the server workstation from the client workstation command prompt.

If the ping command succeeds, ensure that the server's DB2COMM environment variable is set to TCPIP.

If the ping command fails:

- Verify that the server and the client have the same entry for the server TCP/IP port number in the TCP/IP SERVICES file.
- Ensure that the entries for the DB2 server in the client SERVICES file and the server SERVICES file do not contain tab characters.

- Ensure that the DB2 server recognizes the port number. If not, update the DB2 server configuration:
 1. Enter the following command from the DB2 Command Line Processor:


```
update dbm cfg using svcname your-svc
```

where *your-svc* is the server's SERVICE name from the SERVICES file.
 2. Stop and restart DB2.
- If you do not have a domain name server, verify that the TCP/IP host name is entered in the client TCP/IP HOSTS file for the server workstation.
- If you have a domain name server, or an entry for the server in the TCP/IP HOSTS file, select the network icon in the Client NT workstation Control Panel. Verify that the TCP/IP configuration is correct. If it is correct, click the **DNS** push button and verify that the host and domain names are correct.

Cannot connect to an ODBC data source on UNIX

Cause

If an ODBC connection type is selected, ODBC uses the data source name and the required attributes in the .odbc.ini file. This control file must reside in the home directory of the user ID that is used to start the warehouse agent process.

Action

Verify that the .odbc.ini file is in the correct directory. You can find the user ID that is used to start the warehouse agent process on the Parameters page of the Agent Site notebook.

Also verify that the data source is defined correctly in the .odbc.ini file.

Unable to use TCP/IP to connect to AS/400 from Windows NT or Windows 2000

You are trying to use TCP/IP to connect to the AS/400 from a DB2 Connect for Windows NT or Windows 2000 workstation.

Cause

This configuration is not supported.

Action

Try connecting with TCP/IP from a client workstation to the DB2 Connect for Windows NT or Windows 2000 workstation, and then to the host using APPC. For more information, run the following trace on the DB2 Connect for Windows NT or Windows 2000 workstation:

1. At a command prompt, enter:
db2trc on -1 8000000
2. Reproduce the problem.
3. At a command prompt, enter:
db2trc dmp db2.trc
4. To turn off the trace, at a command prompt, enter:
db2trc off

Connecting to a Sybase source fails

You are trying to connect to a Sybase warehouse source (to import table definitions or run a step), and receive an error indicating that the Sybase Open Client Context failed.

Cause

This error can occur because of problems with the environment in which the warehouse agent runs.

Action

Verify that the warehouse agent and the Sybase client do not have conflicting language environment settings. Also, if you are using a Windows NT or Windows 2000 agent, run the warehouse agent daemon or warehouse server (if you are using the default agent) as a user process instead of a system process.

To verify and change the language environment settings:

1. Click **Start** → **Settings** → **Control Panel**.
2. Double-click the **System** icon.
3. Click the **Environment** tab.
4. Update the system environment settings by removing (or renaming) the LANG system environment variable.

For the AIX and Solaris Operating Environment agents, verify that the values specified in the LANG and LC_ALL environment variables are defined in all applicable sections of the locale.dat file (for example, [SUN] and [SUN_SRV4]). Make sure that the locale=xx_xxx entry is before the locale=C, entry in this file.

You can find the environment variables in
/usr/lpp/db2_07_01/bin/IWH.environment or
/opt/IBMdbs2/V7.1/bin/IWH.environment.

5. Verify that the **SYBASE** system environment variable points to the correct directory.
6. Click **OK**.

To run a Windows NT warehouse agent daemon or the warehouse server as a user process:

1. Click **Start** —> **Settings** —> **Control Panel**.
2. Double-click the **Services** icon.
The Services window opens.
3. Select the warehouse agent daemon or warehouse server.
4. Click **Startup**.
The Service window opens.
5. Click **This Account** in the **Log On As** area.
6. Type a valid user ID in the **This Account** field.
7. Type the password in the **Password** and **Confirm Password** fields.
8. Click **OK**.
The Service window closes.
9. In the Services window, click **Close**.
10. Restart the workstation that runs the warehouse agent daemon or warehouse server.

To run a Windows 2000 warehouse agent daemon or the warehouse server as a user process:

1. Click **Start** —> **Settings** —> **Control Panel** —> **Administrative Tools** —> **Services** .
The Services window opens.
2. Right-click the warehouse agent daemon or warehouse server and click **Properties**.
3. Click the **Log on** tab.
4. Select a Windows 2000 user on behalf of which the service should run.
5. Click **OK**.
6. In the Services window, click **Close**.
7. Restart the workstation that runs the warehouse agent daemon or warehouse server.

FTP log displays "NAMEFMT not a valid command"

Cause

This message is displayed if the FTP server system does not support the NAMEFMT command. The NAMEFMT command is an AS/400-specific FTP subcommand.

Action

It is normal for this message to be displayed. You can ignore it.

The warehouse server or warehouse agent is unresponsive when using ODBC drivers that were purchased separately

You are using ODBC drivers that you purchased separately from the Data Warehouse Center, and the warehouse server or warehouse agent is unresponsive.

Cause

Many times, ODBC drivers other than those provided with the Data Warehouse Center produce an informational, warning, or error message when a user or application connects to one of its sources. Because the Data Warehouse Center runs as a background system process, no message is displayed. However, the ODBC driver is waiting for a user response to the message.

Action

If you are experiencing these symptoms when using the Data Warehouse Center, use the ODBCTEST tool to test the connection. Connect to the ODBC source by entering the following command on a command line:

```
ODBCTEST <system dsn name> <odbc user ID> <password>
```

where *<odbc user ID>* is the ODBC user ID to use to connect to the database and *<password>* is the password for the ODBC user ID.

The ODBCTEST tool writes messages to the command window. If any ODBC-related messages appear, address the situation as described in the ODBC message, or use the ODBC drivers that come with the Data Warehouse Center.

Installation issues

This section details errors that can occur when you install or initialize the Data Warehouse Center product.

Windows NT or Windows 2000 cannot install services for Data Warehouse Center components

When you install the Data Warehouse Center, the warehouse logger and warehouse server files are not installed in the Windows NT or Windows 2000 Services folder.

Cause

You are installing from a Windows NT or Windows 2000 user ID that does not have administrative authority.

Action

Log off of Windows NT or Windows 2000 and log on with a user ID that has administrative authority.

Initialization fails at "Add install path to NT service path" message

Data Warehouse Center initialization stops at the Add install path to NT service path message.

Cause

You are installing from a Windows NT user ID that does not have administrative authority.

Action

Log off of Windows NT and log on with a user ID that has administrative authority.

Initialization fails with a "3043 Disk or Network Error" message

Data Warehouse Center initialization cannot write to the TEMP or TMP directories, or the drive is full.

Cause

Either the TEMP or TMP directory (or both) is not defined.

Action

You must have the following valid statements in the **User Environment Variables** list in your System folder.

```
TEMP = drive:\TEMP  
TMP = drive:\TEMP
```

where drive is any drive with available space for temporary files.

The System folder is located in the Control Panel folder. The TEMP directory must be defined in the **User Environment Variables** list in the System folder.

To direct the Data Warehouse Center initialization process to write to the TEMP or TMP directories:

1. Create the TEMP directory to which the statements refer.
2. Ensure that the statements are in your User Environment Variables folder.
3. Restart your workstation.
4. Initialize Data Warehouse Center again.

Data Warehouse Center files disappear after you install the product again

You installed the Data Warehouse Center again, but you cannot locate the product files.

Cause

You installed the product without restarting after the removal procedure.

Action

You must restart your workstation after you remove the product and before you install the product again. See *DB2 Universal Database Quick Beginnings* for more information.

The Data Warehouse Center will not start

You receive the error message, *SYS0005: Access denied*. As a result, you cannot access any Data Warehouse Center services.

Cause

You tried to access the Data Warehouse Center by using a different user ID than the one that you used when you installed the product.

Action

Log on to your Windows NT or Windows 2000 workstation with the same user ID that you used when you installed the Data Warehouse Center.

Initialization fails with an "Unable to connect" message

You receive the following message, *Unable to connect to Database: IWHDB01...* after a successful installation, even if you can connect to the database by using the DB2 Command Line Processor.

Cause

There are several reasons why you might get this message:

- The database is not registered as an ODBC system DSN.
- You did not start the DB2 security server.
- You installed DB2 for Windows NT or Windows 2000 with a user ID that did not have administrative authority.

Action

Try one of the following corresponding solutions:

- Register the database as an ODBC system DSN.
- Start the DB2 security server from Windows NT or Windows 2000 Services.
- Install DB2 for Windows NT or Windows 2000 again under a user ID that has administrative authority.

Initialization fails with a 9159: Win32 function RegOpenKeyEx failed message

You attempted to initialize the Data Warehouse Center after you installed the Data Warehouse Center from a Windows NT or Windows 2000 user ID that does not have administrator authority.

Cause

The Data Warehouse Center could not open a key in the Windows NT or Windows 2000 registry.

Action

Install the Data Warehouse Center again from a Windows NT or Windows 2000 user ID that has administrator authority.

Product function issues

These problems can occur while you use the Data Warehouse Center.

A Data Warehouse Center component is unable to communicate with another component

The following message information is displayed with this error:

```
Return code = 6106  
Message = 6117  
Method VWSocket::Send  
Secondary Code = 10061  
System Message = 10061
```

Cause

A Data Warehouse Center component was unable to communicate with another Data Warehouse Center component. The most common occurrence of this error is when the warehouse server cannot communicate with the agent daemon because the agent daemon is not running.

Action

On Windows NT, open the Windows NT Control Panel and double-click the **Services** icon. On Windows 2000, open the Control Panel and click **Administrative Tools** —> **Services**. Scroll down the list of services until you find **Warehouse Server**, **Warehouse Logger**, and **Warehouse Agent**. If the services are stopped, start them. (The server and logger are linked so that both start when you start one of them.)

You can also use the netstat command on Windows NT or Windows 2000 to determine the state of the warehouse agent daemon, warehouse server, and warehouse logger. Type the following command to display all active services:

```
netstat -a
```

Look to see if the services are active:

Warehouse server

```
vwkernel
```

Warehouse logger

```
vwlogger
```

Warehouse agent daemon

```
vwd
```

On UNIX, type the following command to verify that the warehouse agent daemon is active:

```
ps -ef | grep vwd
```

If the warehouse agent daemon is active, type the following command to verify that the daemon is listening from its assigned port:

```
netstat -a | grep vwd
```

The Data Warehouse Center and the AS/400 agent are unable to communicate

Cause

To communicate with each other, the Data Warehouse Center and the AS/400 agent must use the same TCP/IP port entries. If you change a port number in the Windows NT Services file, but do not change it on AS/400 (or vice-versa), communication between the Data Warehouse Center and the AS/400 agent will fail.

The Data Warehouse Center refers to ports by name rather than by number. During installation, the Data Warehouse Center assigns names to port numbers. On Windows NT or Windows 2000, the Data Warehouse Center writes the port assignments to the Services file. On AS/400, the Data Warehouse Center uses the WRKSRVTBLE command to store the port assignments. By default, the Data Warehouse Center uses these ports:

- 11000 (vwkernel)
- 11001 (vwd)
- 11002 (vwlogger)

Action

Change a port number only if that port number is already in use by another application. If you change a port number on either Windows NT or Windows 2000, or AS/400, ensure that you change the corresponding number on the other operating system.

The warehouse logger stops while the Data Warehouse Center is running

If the warehouse logger stops while the Data Warehouse Center is running, you will hear five beeps. The following message might be displayed:

```
There are no log records to display
```

Cause

If this message is displayed, the DB2 log for the warehouse control database is probably full.

Action

To fix this problem, you must increase the number of primary or secondary files for the warehouse control database.

If this message is not displayed, contact IBM Software Support.

A warehouse agent stops processing when promoting a step

The step promotion stops, but you can still access other parts of the Data Warehouse Center.

Cause

The warehouse agent is in a lock-wait state on a target.

Action

You need to free the warehouse agent workstation. To do this:

1. Open the DB2 Command Line Processor, and enter:

```
DB2 LIST APPLICATIONS
```

The DB2 agent ID for the locked agent is listed.

2. Record the DB2 agent IDs for the source and the target.
3. Enter:

```
DB2 FORCE APPLICATIONS (XXX)
```

where XXX is the DB2 agent ID for the source.

4. Repeat the previous step for the target.

Creation of an AIX or Solaris Operating Environment warehouse agent process fails

There is no response from the warehouse agent, or the warehouse agent is unable to connect to a source or target database.

Cause

There are several reasons why you might have this problem:

- The agent user ID might not be authorized to start the agent.
- The DB2 runtime library libdb2.a file might be missing from the library path.
- The warehouse agent might not be able to communicate with the warehouse server by its local name. The warehouse server cannot spawn a warehouse agent process.
- The default port numbers that are assigned to the Data Warehouse Center TCP/IP services might be in use by other services.
- The incorrect warehouse agent type might be installed for your connection type.

Action

Try the following solutions:

- Check the permission bits on the agent executable file to verify that the agent site user ID is authorized to start the agent.

- Verify that the libdb2.a file is in the library path (LIBPATH on AIX or LD_LIBRARY_PATH for the Solaris Operating Environment). If it is not, copy the file to the library path. To find the libdb2.a file, enter `echo $LIBPATH` at a command prompt.
- Verify that TCP/IP communication between the server and the warehouse agent is active. For example, ensure that the agent can communicate with the server by its local name. Update the `/etc/hosts` file in the AIX or Solaris Operating Environment host as follows:

```
ipaddress hostname.domainname
```

Also, you should ping the warehouse server from the agent site and vice-versa.

- Verify that the port numbers that the Data Warehouse Center TCP/IP services use in the Services file are not in use by other services. The default numbers for the Data Warehouse Center TCP/IP services are:
 - `vwkernel 11000/tcp`
 - `vwd 11001/tcp`
 - `vwlogger 11002/tcp`

If other TCP/IP services currently use these port numbers, change the port numbers that are assigned to the Data Warehouse Center TCP/IP services to port numbers that are not currently in use. Change the port numbers on both the agent site and the server workstation.

- Check the warehouse agent log for information.
- Ensure that the correct warehouse agent type is installed for your connection type. See the *DB2 Warehouse Manager Installation Guide* for more information.

The Data Warehouse Center stops processing if you run several steps at the same time

The Data Warehouse Center stops processing; there are no status changes in the Work In Progress window for currently running step editions.

Cause

Your target is locking up because too many warehouse agents are trying to update a target table at the same time.

Action

Change the database configuration parameter `LOCKS TIMEOUT`. The default value is `-1`, which means that DB2 will wait forever. If you are experiencing locking problems, use the DB2 Control Center or the DB2 Command Line Processor to set the wait time to 30 seconds.

Cannot import tables from AS/400

The Data Warehouse Center fails to access DB2 for AS/400 tables.

Cause

There are some problems that can occur when you use the Data Warehouse Center for Windows NT with AS/400. For example:

- Requests from any ODBC application for tables from DB2 for AS/400 can fail.
- The table filter might not work during table import.
- The table import performance might be slow, or the Data Warehouse Center might not appear to return any tables when importing.

Action

Apply DB2 server fix pack US8125D or later.

Columns defined as BINARY WITH PRECISION are interpreted as INTEGER

If your source table is an AS/400 file that contains fields that are defined as BINARY WITH PRECISION, the data types of these fields will be returned as INTEGER by DB2 for AS/400.

Cause

This is a permanent AS/400 restriction. This restriction causes the fractional portion of the BINARY WITH PRECISION data to be truncated.

Prepare of SQL statement fails**Cause**

Not enough storage is available in the application heap to process the statement.

Action

You need to increase the application heap size for the warehouse control database. See *DB2 Universal Database Quick Beginnings*.

The Data Warehouse Center displays return code 7356

This section describes some of the problems that can cause the Data Warehouse Center to display return code 7356.

Importing source tables from a remote, earlier release of an AS/400 system

When you attempt to import table definitions from a remote AS/400 system before Version 4 Release 2 to a new warehouse source, you might receive the following messages:

- From the Data Warehouse Center:

```
An agent's processing of a
command of type
<command type> failed for
edition <edition number> of
step <step name>.
```

A secondary return code is also displayed:

```
Method = VWabsCommand::ExecuteOnce  
Secondary Code = 8515  
System Message = Section number 500 not valid. Current  
high section number is 200.
```

- From the AS/400 system:

```
Message . . . . : Section number 500 not valid. Current high  
section number is 200. Reason 4.
```

Cause: This error occurs when the SQL package for the SQL Call Level Interface is older than the current version. The SQL package is located in the QGPL library on the system that is remote from the system where the import was requested.

Action: This error occurs when the SQL package for the SQL Call Level Interface is older than the current version. The SQL package is located in the QGPL library on the system that is remote from the system where the import was requested.

To correct this problem:

1. Delete the SQL package from the QGPL library on the system that is remote from the system where the import was requested. To do this, enter the following command at an AS/400 command prompt:
DLTSQLPKG SQLPKG(QGPL/QSQCLI*)
2. Import your tables again.
A new, current SQL package for the Call Level Interface is created when you import.

Preserving CCSIDs in a CREATE TABLE statement

Cause: The character coded set identifier (CCSID) is an identifier for a collection of one or more code pages and character sets, and the encoding scheme for the character set.

When you promote a step that you specify to create the target table to test mode, the character and graphic columns are created with the CCSID of the job that creates the target table during the promote operation. When you run the step, the actual CCSIDs are retrieved from the warehouse source. If your step extracts character or graphic data, an error will occur if the default CCSIDs that were created during the promotion of the step do not match the actual CCSIDs. If this error occurs, you will receive the following messages:

- From the Data Warehouse Center, you receive return code 7356.
- From the AS/400 system:


```
Message ID . . . . . : SQL0330   Severity . . . . . : 30
Message type . . . . . : Diagnostic
Date sent . . . . . : aa/aa/aa   Time sent . . . . . : bb:bb:bb

Message . . . . . : Character conversion cannot be performed.
```

aa/aa/aa is the date the message is sent.

bb:bb:bb is the time the message is sent.

Action: To preserve the AS/400 CCSID for the individual columns in character and graphic type data, you must create the target table with the CCSID and link it to the step in the process modeler before you promote the step.

Target table is not being journaled

Cause: Return code 7356 is displayed when the target table is not being journaled. To verify that this is the problem, check the AS/400 agent job log.

You should see the following message:

```
CPF4328 Member nnnnnnnnnn not journaled to journal *N
```

where nnnnnnnnnn is the name of the file/table.

Action: To resolve this problem, ensure that the target table is being journaled.

Promoting a step that uses DB2 for MVS Version 4 Release 1 as the source and an AS/400 agent hangs

Cause

During the promotion of a step with DB2 for MVS Version 4 Release 1 as the source, the Data Warehouse Center might seem to hang. This problem is caused by the IWH2AGNT job going into an ICFW wait state indefinitely.

To verify that the IWH2AGNT job is the cause of the problem:

1. At an AS/400 command prompt, enter WRKACTJOB.
2. Verify that IWH2AGNT is in an ICFW state.

Action

To resolve this problem:

1. Apply PTF UQ02568 (APAR PQ01375) to DB2 for MVS Version 4 Release 1.
2. Apply PTFs SF47985 and SF47986 to your AS/400 Version 4 Release 2 system.

No changes to the Data Warehouse Center are required.

Promoting a step that uses DB2 for OS/390 as the warehouse target and the AS/400 agent fails

When you promote a step to test mode using DB2 for OS/390 as your warehouse target, the promotion might fail with the following messages:

- An 8525 return code
- The system message Resource Limit exceeded

Cause

This problem occurs when the Data Warehouse Center tries to create the target table in the DB2 for OS/390 default database and table space, and the default table space is already full or restricted to your user ID.

Action

You can resolve this problem in two ways:

- If you want your target tables to go to your default database, you can increase the amount of space in your default database.
- If you don't want the target tables to go to the default database, you can create another database and table spaces for use by the Data Warehouse Center.

After you create the database and table spaces, complete the following tasks:

1. Define a warehouse for the database.
2. When you create a target table for your new warehouse, specify the table space to use for the table, as described in the *Data Warehouse Center Administration Guide*.

The Data Warehouse Center displays the "Could not retrieve the step RC-3142" message when creating a step with a long name

The Data Warehouse Center displays the "Could not retrieve the step RC-3142" message when creating a step with a long name.

Cause

This error occurs on a Japanese (double-byte language) operating system. It could potentially occur on other double-byte workstations. The step names can only be 80 characters long or 40 double-byte characters.

Action

Shorten the length of the name. If you requested that the Data Warehouse Center create a default table for the long-name version of the step, delete the definition of the table that exists in the metadata.

The warehouse server cannot create a warehouse agent or it did not receive a valid reply back from the warehouse agent

You are using a function in the Data Warehouse Center administrative interface that works with the warehouse agent, and receive a message that the warehouse server is not able to create a warehouse agent, or it did not receive a valid reply from the warehouse agent. (The message number is DWC07183.)

Cause

The most common causes for this error are an incorrect TCP/IP configuration or a warehouse agent start/stop timeout value that is too small. This error can also occur if system resources are too low to allow the warehouse agent daemon to create a warehouse agent, or if the warehouse agent executable file is missing or damaged.

One common TCP/IP configuration error occurs when the warehouse server host name is not known to the site on which the warehouse agent is running. The warehouse server and warehouse agent must be able to use each other's host name. Error DWC07183 occurs when the fully qualified host name of the warehouse server (for example, mymachine@www.ibm.com) is not known to the site on which the agent is running.

Action

You can confirm the host name by using the ping command to ping the server. If the ping fails, it is because of one of the following problems:

- The host name is not registered with your domain name server (DNS).
- The fully qualified name is not in the host file on the agent site.
- The numeric IP address associated with the fully qualified host name is no longer valid.

The warehouse agent start/stop timeout value determines how long the warehouse server will wait for the warehouse agent to start. You might need to raise the start/stop timeout value if:

- The server workstation is slow.
- The agent site is slow.
- The network is slow or unreliable.

To increase the warehouse agent start/stop timeout value:

1. From the main Data Warehouse Center window, right-click the **Warehouse** object and click **Properties**.
2. Click the **Server** tab.
3. Increase the agent start/stop timeout value.
4. Close the Properties notebook.
5. Stop and restart the warehouse server.

When you increase the start/stop timeout value, the warehouse server will wait longer before timing out. Timeouts occur if the warehouse agent fails to start and when the network is down between the warehouse server workstation and the agent site.

The Start/End Date/Time is incorrect in the Work In Progress window

The date and time are incorrect in the **Scheduled** and **Completed** columns of the Work In Progress window.

Cause

Your Windows NT or Windows 2000 system clock is set incorrectly.

Action

Open the Regional Settings application in the Control Panel, and change the clock setting to the correct time and date. Include the correct time zone.

DB2 reports an SQL0973N error when you open and close steps

The DB2 application heap and package cache default values can handle only a limited number of steps.

Cause

Not enough storage space is allocated to the DB2 application heap and package cache. The default settings for a database are:

applheapsz	128
pckcachesz	36

Action

Change the default settings for the DB2 application heap and package cache:

applheapsz	512
pckcachesz	128

A step that uses Microsoft Access as a source fails

You repeatedly try to run a step that uses Microsoft Access. The step fails every time.

Cause

There are several reasons that the error might occur:

- The SELECT statement in your step definition is using an unsupported function.

For example, you are accessing data from a MEMO text field in an Access database and placing it in a LONG VARCHAR field in your step. You need to use the Left() function, which is supported by Microsoft Access, instead of the Substr() function.

- The import column definition failed because the driver is not capable of providing column definitions.

Action

Try one of the following solutions:

- Check the `SELECT` statement in your step definition to ensure that functions are supported by the Microsoft Access driver.

For a mapping of the ANSI SQL data types that are supported by Microsoft Access, open the Microsoft Access Help Topics window and search for **Equivalent ANSI** on the **Find** tab. You will find a help topic titled **Equivalent ANSI SQL Data Types**.

- The Microsoft Access ODBC driver does not support import of column definitions. You can manually define the columns in the step notebook.

The Data Warehouse Center supplies incorrect table names or an ODBC OLE error occurs when using the Microsoft Excel 95/97 ODBC driver to access Excel spreadsheets

You have problems using the Microsoft Excel 95/97 ODBC driver to access Excel spreadsheets as a warehouse source. \$ is appended to the end of the table name for each worksheet in the spreadsheet, or you receive an ODBC OLE error when you access the source using the Data Warehouse Center.

Cause

The ODBC driver is generating the names of the tables.

Action

Create a named table for each of the worksheets within the spreadsheet.

To create a named table for each worksheet:

1. Select the columns and rows that you want in the table.
2. Click **Excel** → **Insert** → **Name** → **Define**.
3. Type a name (or use the default name) for the marked data.
4. Click **OK**.

You can now import tables when you define your warehouse source without selecting the **Include system tables** check box.

An file access error occurs when using the Data Warehouse Center ODBC text file driver or accessing files using warehouse programs

You are using the Data Warehouse Center ODBC text file driver or accessing files using warehouse programs, and receive an error that the file cannot be found or that you cannot access the specified drive. This problem occurs on Windows NT or Windows 2000 only.

Cause

The warehouse agent does not have access to the file. The file might be stored on a remote system that requires a user ID and password or on a LAN drive.

Action

Verify that the file being used is on a local drive, or change the setting of the warehouse agent (or the warehouse server if you are using the default agent) in the Services folder so that the agent runs on behalf of a Windows NT or Windows 2000 user rather than as a system process.

To run the service as a user process rather than a system process on Windows NT:

1. On Windows NT, click **Start** → **Settings** → **Control Panel** → **Services**.
On Windows 2000, click **Start** → **Settings** → **Control Panel** → **Administrative Tools** → **Services**.
2. Click the warehouse server or warehouse agent.
3. Click **Startup**.
Choose a Windows NT user on behalf of which the service should run.
4. Stop and restart the service.

To run the service as a user process rather than a system process on Windows 2000:

1. Click **Start** → **Settings** → **Control Panel** → **Administrative Tools** → **Services**.
The Services window opens.
2. Right-click the warehouse agent daemon or warehouse server and click **Properties**.
3. Click the **Log on** tab.
4. Select a Windows 2000 user on behalf of which the service should run.
5. Click **OK**.
6. In the Services window, click **Close**.
7. Stop and restart the service.

Column is no longer visible in the column grid

When you widen a column in a grid and then edit the column, the first column is no longer visible in the grid.

For example, when you widen and edit the **Parameter Text** column in the Parameters page of the Program notebook, the **Parameter Name** column is no longer visible in the grid.

Cause

The combined width of the columns in the grid is larger than the area available to display the columns.

Action

The missing column still exists, but it just is not visible. Press the left or right arrow key to scroll to the missing column.

Supplied warehouse program issues

This section describes problems that can occur when you use the supplied warehouse programs, and suggests solutions to resolve them.

Step running VWPRCPY fails with RC2 = 12**Cause**

Verify that the VWxxxxxxx.VWPRCPY trace file contains a message that says the FTP command file could not be created, AS/400 error = 3021.

If this is the case, the user profile under which the agent process was running does not have the authority to create members in the QIWH/FTPCMD source physical file.

Action

To fix this problem, either use a different user profile to run the agent or grant the agent user profile authority to access the QIWH/FTPCMD file.

On AS/400, the VWPFTP trace says that the job completed successfully, but data was not transferred and the FTP message log is missing**Cause**

The library or physical file name that you supplied with the FTP command file member does not exist or is invalid. Because of this, FTP executes successfully, but cannot execute the stream of commands in the file. As a result, FTP does not transfer the data. The FTP message log does not get generated.

Action

To resolve this problem, ensure that the library, physical file, and command member actually exist. If they do, check for typographical errors in the command file parameter value passed to VWPFTP. The VWPFTP trace lists the parameter values that were passed to it.

One of the OLAP Server warehouse programs cannot find the rules file or load file

You are running one of the OLAP Server warehouse programs from the Data Warehouse Center, and you receive an error message stating that the (rules or load) file cannot be found.

Cause

The location of the rules file or load file is unknown.

Action

When you define the step that uses these warehouse programs, specify whether the load file or rules file exists on the client workstation or on the server workstation. If you specify that the file is on the client workstation, you need to fully qualify the file name (drive, path, file name, and extension) when you define the warehouse source. If you specify that the file is located on the server workstation, you need to specify only the file name (do not include the drive, path, or extension) and verify that the file is located in the appropriate directory on the server workstation.

User-defined programs are not found, or they run with errors

You are running user-defined programs, and receive an error that indicates that they are not found, or other errors.

Cause

One possible cause is that the user-defined program, or services used by the program, are installed in a directory that is not listed in the PATH statement. For example, the user-defined program could be installed on a LAN drive. Another possible cause is that the user-defined program requires access to environment variables that are defined as user variables. The environment variables must be defined as system variables for the user-defined program to access them.

Action

Verify that the user-defined program is installed in a directory listed in the PATH statement. Verify that any directories that contain programs or services that the user-defined program will use are also listed in the PATH statement. Verify that all environment variables used by the user-defined program are defined as system variables, not user variables.

Use the trace level setting to validate the path and variables that the user-defined program has access to under the warehouse agent:

1. Right-click the **Warehouse** folder, and click **Properties** from the pop-up menu.
2. On the Trace Level page, use the **Agent** button to set the agent trace to trace level 4.
3. Click **OK**.
4. Stop and restart the warehouse server.
5. Run the operation that previously failed.
Two files, AGNTxxx.SET and AGNTxxx.LOG, are created at the agent site, where xxx is the process ID of the agent. For some user-defined programs, a third log file will be created, trcxxx.log.
6. Review the AGNTxxx.SET file to verify that the path is properly set and that all environment variables are present.

7. Repeat steps 1 through 4 to set the trace level to the original setting. The recommended setting is 0.

If you encounter problems using DB2 Universal Database batch programs, verify that the long form of the DB2 UDB CONNECT statement is used:

```
CONNECT TO <database> USER <user ID> USING <password>
```

Do not use:

```
CONNECT TO <database>
```

Use the long form because the warehouse agent is running as a system process and does not have access to any of the user information.

Still Have Problems?

If you still have problems with Data Warehouse Center:

- See the DB2 Product and Service Technical Library on the Web, at <http://www-4.ibm.com/software/data/db2/library/>. (Note that this information is in English only.)
- See “Appendix C. Using the DB2 Library” on page 277 for information about the location of release notes for this product.
- Contact the vendor of your application or compiler if you suspect a defect or limitation.
- Contact DB2 Customer Support with the following information:
 - A description of the events leading up to the problem.
 - A full description of the problem, including any SQL messages and associated reason codes.
 - The db2diag.log, and any trap or dump files, for clients, servers, and DB2 Connect Enterprise Edition gateways. For more information, see “First Failure Data Capture” on page 157.
 - The applicable Data Warehouse Center component trace files and start error trace files. For more information, see the *Data Warehouse Center Administration Guide*.
 - Output from **db2level**.

Chapter 11. Troubleshooting DB2 Satellite Edition

DB2 Satellite Edition enables mobile, or pervasive, computing where occasionally connected computers have locally persistent data that is easily updated and synchronized with corporate data at a central site.

Status information about each production satellite is retained in a central database known as the *satellite control database*. The status information includes problem reporting. The Satellite Administration Center is used to review the status information from the satellite control database.

If a problem has been reported then test satellites, model offices, and script files called *batches* are used to remotely administer satellites and to troubleshoot satellite problems.

Within a satellite environment, test satellites are used to test the group batches that set up and maintain the database definition that supports the end-user application. How to define a satellite as a test satellite is described in the online help available from the Satellite Administration Center.

In addition, the satellite environment has a model office which is a special member of the test satellites in a group. Typically, you have one model office for each version of the end-user application that you have deployed in the group. There are several reasons for having model offices:

- To model the initial deployment of a group.
- To test changes that are required to the database definition and data that supports a version of the end-user application that is already in production.
- To provide a representation of a typical satellite in the group. When an end-user experiences a problem, the model office, or a copy of it, can be used to reproduce the problem and to determine the best way to correct the problem.
- To verify the deployment of a new version of a new end-user application.

Model offices are key to solving problems within a satellite environment. Test satellites can be created that are copies of the model office. These test satellites can then be used to reproduce and solve any problems or to prevent problems through the testing of group batches before they are deployed to production satellites.

You can find details about troubleshooting in a satellite environment in the *Administering Satellites Guide and Reference* manual.

Replication is a major feature of the satellite environment. Problems encountered in a satellite environment may involve replication. See “Chapter 9. Troubleshooting Replication” on page 113 for information on troubleshooting replication problems.

Still Have Problems?

If you still have problems with the satellite environment:

- See the DB2 Product and Service Technical Library on the Web, at <http://www-4.ibm.com/software/data/db2/library/> (Note that this information is in English only.)
- Contact the vendor of your application or compiler if you suspect a defect or limitation.
- Contact DB2 Customer Support with the following information:
 - A description of the events leading up to the problem.
 - A full description of the problem, including any SQL messages and associated reason codes.
 - The `db2diag.log` file, and any trap or dump files, for clients, servers, and DB2 Connect Enterprise Edition gateways. For more information, see “First Failure Data Capture” on page 157.
 - Output from **db2level**.

Chapter 12. Troubleshooting NetQuestion

NetQuestion is an HTML search system that is installed automatically if you choose to install product documentation with any DB2 product. Provided your operating system meets NetQuestion's installation prerequisites, you do not have to do anything special to install and work with NetQuestion.

There are two versions of NetQuestion:

- The single-byte character set (SBCS) version is installed with documentation in languages such as English, French, German, and Spanish.
- The double-byte character set (DBCS) version is installed with documentation in Japanese, Simplified Chinese, Traditional Chinese, and Korean.

For information about troubleshooting NetQuestion, see the *Installation and Configuration Supplement*.

Still Have Problems?

If you still have problems with NetQuestion:

- See the DB2 Product and Service Technical Library on the Web, at <http://www-4.ibm.com/software/data/db2/library/> (Note that this information is in English only.)
- See "Appendix C. Using the DB2 Library" on page 277 for information about the location of release notes for this product.
- Contact the vendor of your application or compiler if you suspect a defect or limitation.
- Contact DB2 Customer Support with the following information:
 - A description of the events leading up to the problem.
 - A full description of the problem, including any SQL messages and associated reason codes.
 - The `db2diag.log` file, and any trap or dump files, for clients, servers, and DB2 Connect Enterprise Edition gateways. For more information, see "First Failure Data Capture" on page 157.
 - Output from `db2level`.

Part 2. Advanced DB2 Troubleshooting

This part addresses the following topics:

- “Chapter 13. Logs and Traces Roadmap” on page 147
- “Chapter 14. The DB2 Process Model” on page 151
- “Chapter 15. Logged Information” on page 157
- “Chapter 16. Traces” on page 177
- “Chapter 17. Diagnostic Tools for UNIX-Based Systems” on page 207
- “Chapter 18. Diagnostic Tools for OS/2” on page 217
- “Chapter 19. Diagnostic Tools for Windows 32-bit Operating Systems” on page 229
- “Chapter 20. GUI and Monitoring Tools” on page 233
- “Chapter 21. Table and Table Space Recovery” on page 241.

Chapter 13. Logs and Traces Roadmap

When you contact DB2 Customer Support, you may be requested to provide logs or perform traces to help determine the cause of problems within DB2. The following tables list the various type of logging, tracing, and alert facilities provided by DB2.

DB2 Diagnostics			
Facility	Platform	Notes	Refer to...
db2dart	All	<ul style="list-style-type: none"> For servers. Analyzes database objects for architectural integrity. 	"Miscellaneous Troubleshooting Tools" on page 15.

DB2 Logging			
Facility	Platform	Notes	See...
db2diag.log	All	<ul style="list-style-type: none"> For clients and servers. Logs errors and (depending on the diagnostic level) warnings and informational messages. 	"First Failure Data Capture" on page 157.
Alert file	All	<ul style="list-style-type: none"> For clients and servers. Contains severe DB2 errors. 	"Alerts" on page 173.
Dump files	All	<ul style="list-style-type: none"> For clients and servers. Provides additional information for some errors. 	"Dump Files" on page 169.
Trap files	All	<ul style="list-style-type: none"> For clients and servers. Files that are created by DB2 in the event of a segmentation violation or trap. 	"Trap Files" on page 170.

DB2 message files	All	<ul style="list-style-type: none"> • For servers. • Logs activities during operations such as bind, load, import and export. 	“Using Message Files” on page 172.
DB2 Performance monitor	Windows NT, Windows 95, OS/2	<ul style="list-style-type: none"> • For clients and servers. • Logs deadlocks and other performance information. 	“The Database System Monitor” on page 18, and the <i>System Monitor Guide and Reference</i> .
Administrative tools log	Windows NT, Windows 95, OS/2	<ul style="list-style-type: none"> • For clients and servers using the Control Center and other DB2 administrative tools. • Logs information for the administrative tools. 	“Using the Administrative Tools Log” on page 171.

Traces			
Facility	Platform	Notes	See...
db2trc	All	<ul style="list-style-type: none"> • For clients and servers. • Traces DB2 events. 	“Using the DB2 Trace Facility (db2trc)” on page 177.
Administrative tools trace	Windows NT, OS/2	<ul style="list-style-type: none"> • For clients and servers using the Control Center and other DB2 administrative tools. • Traces problems with the administrative tools. 	“Administration Tools Trace” on page 183.
db2drdat	All	<ul style="list-style-type: none"> • Traces DRDA requesters and servers. 	“Taking a DRDA Trace (db2drdat)” on page 185 and <i>DB2 Connect User’s Guide</i> .
ddcstrc	All	<ul style="list-style-type: none"> • Traces DB2 Connect Enterprise Edition gateways. 	<i>DB2 Connect User’s Guide</i> .
CLI trace	All	<ul style="list-style-type: none"> • Traces CLI, JDBC, or ODBC applications. 	See the <i>CLI Guide and Reference</i> manual.

SNA trace	All	<ul style="list-style-type: none"> Traces communications via SNA. 	"Taking an SNA Trace" on page 186.
-----------	-----	--	------------------------------------

Process Identification			
Facility	Platform	Notes	See...
ps	UNIX	<ul style="list-style-type: none"> For clients and servers. Shows current processes (along with process names for AIX). 	"Chapter 14. The DB2 Process Model" on page 151 and "Viewing Process Status Using the ps Command" on page 212.
pstat	OS/2	<ul style="list-style-type: none"> For clients and servers. Shows processes and their threads. 	"Chapter 14. The DB2 Process Model" on page 151 and "Using the pstat Command" on page 225.
Windows NT Performance monitor	Windows NT	<ul style="list-style-type: none"> For clients and servers. Shows processes and their threads. 	"Chapter 14. The DB2 Process Model" on page 151 and "Using Administrative Tools Provided by Windows NT" on page 230.

System Logging			
Facility	Platform	Notes	See...
System error log (syslog)	UNIX and OS/2	<ul style="list-style-type: none"> Logs alert conditions. For use by DB2 Customer Support. 	"Using the System Error Log (syslog)" on page 207 and "Using the OS/2 Logging Facility" on page 218.
Core file	UNIX	<ul style="list-style-type: none"> For UNIX-based systems. Logs severe errors. For use by DB2 Customer Support. 	"Using the Core File" on page 210.
UNIX commands	UNIX	<ul style="list-style-type: none"> Let you monitor and troubleshoot UNIX-based systems. 	"Other Commands" on page 213.

Event logger	Windows NT	<ul style="list-style-type: none"> • Logs system errors, including severe DB2 errors for Windows NT systems. • Accessible to remote system administrators. 	"Diagnostic Tools for Windows NT" on page 229.
SPM/2	OS/2	<ul style="list-style-type: none"> • Monitors the performance of OS/2. 	"Chapter 18. Diagnostic Tools for OS/2" on page 217.
Administrative tools	Windows NT Version 4.0	<ul style="list-style-type: none"> • Provide various administrative tools to diagnose and monitor Windows NT systems. 	"Diagnostic Tools for Windows NT" on page 229.
Dr. Watson	All Windows systems	<ul style="list-style-type: none"> • Logs system errors during a General Protection Fault (GPF). 	"Diagnostic Tools for Windows NT" on page 229.
Native OS/2 logging	OS/2	<ul style="list-style-type: none"> • Logs error information on OS/2 programs. 	"Using the OS/2 Logging Facility" on page 218.

Alerts			
Facility	Platform	Notes	See...
DB2 alert file	All	<ul style="list-style-type: none"> • For clients and servers. • Contains severe errors. 	"Alerts" on page 173.
SNA alerts	OS/2	<ul style="list-style-type: none"> • Sends information over an SNA network in the event of a severe error. • Can be generated by DB2 servers. 	"Using the FFST for OS/2 Utility" on page 219.
SNMP subagent	Windows NT, AIX, OS/2	<ul style="list-style-type: none"> • Sends information over an SNMP network in the event of a severe error. • Can be set up for DB2 servers. 	"SNMP Alerts" on page 173.

Chapter 14. The DB2 Process Model

Knowledge of the DB2 process model can help you determine the nature of a problem, because it helps you to understand how the database manager and its associated components interact.

Most UNIX-based environments let you see a list of the current DB2 processes. For example, you can see if an application has an agent assigned to it, if the database server is up, or if a communication protocol's listener is up.

Note: You cannot identify these processes on the OS/2 and Windows operating systems, because DB2 is implemented using a multithreaded model. If you use these operating systems, you may wish to ignore this chapter.

Description of the DB2 Process Model

The process model used by all DB2 servers facilitates the communication that occurs between database servers and client and local applications. It also ensures that database applications are isolated from resources such as database control blocks and critical database files.

UNIX-based environments use an architecture based on **processes**. For example, the DB2 communications listeners are created as processes. Intel operating systems such as OS/2 and Windows NT use an architecture based on **threads** to maximize performance. For example, the DB2 communications listeners are created as threads within the DB2 server's system controller process. The process model for DB2 describes the behavior of processes and threads.

For each database being accessed, various processes/threads are started to deal with the various database tasks (for example, prefetching, communication, and logging).

Each process/thread of a client application has a single **coordinator agent** that operates on a database. A coordinator agent works on behalf of an application, and communicates to other agents using inter-process communications (IPC) or remote communication protocols.

DB2 architecture provides a **firewall** so that applications run in a different address space from DB2. The firewall protects the database and the database manager from applications, stored procedures, and user-defined functions (UDFs). A firewall maintains the integrity of the data in the databases, because

an application programming error cannot overwrite an internal buffer or file of the database manager. It also improves reliability, because an application programming error cannot crash the database manager.

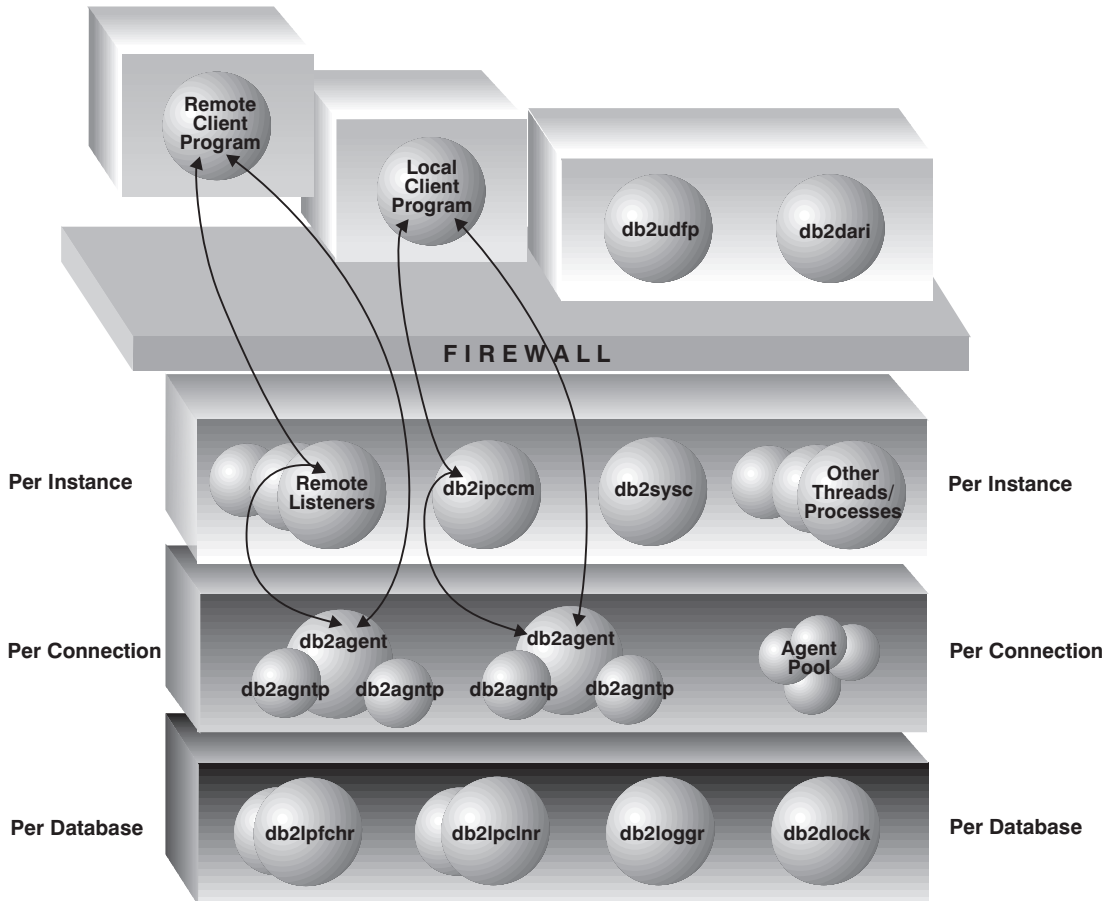


Figure 1. Process Model for DB2 Systems

The following list provides additional details on the processes/threads shown in Figure 1:

Client Programs

Client programs run remotely or on the same machine as the database server. They make their first contact with the database through a listener. A coordinator agent (**db2agent**) is then assigned to them.

Listeners

Client programs make initial contact with communication listeners, which are started when DB2 is started. There is a listener for each configured communication protocol, and an inter-process communications (IPC) listener (**db2ipccm**) for local client programs. Listeners include:

- **db2ipccm**, for local client connections
- **db2tcpdm**, for TCP/IP connections
- **db2snacm**, for APPC connections
- **db2tcpdm**, for TCP/IP discovery tool requests

To find out if there was a problem activating a listener, consult the **db2diag.log** at the server. For more information, see “Using the **db2diag.log** File to Diagnose Server Communication Problems” on page 47.

Agents

All connection requests from client applications, whether they are local or remote, are allocated a corresponding coordinator agent (**db2agent**). When the coordinator agent is created, it performs all database requests on behalf of the application.

In some environments in which the *intra_parallel* database manager configuration parameter is enabled, the coordinator agent distributes the database requests to subagents (**db2agntp**). These agents perform the requests for the application. Once the coordinator agent is created, it handles all database requests on behalf of its application by coordinating subagents (**db2agntp**) that perform requests on the database.

A coordinator agent may be:

- Connected to the database with an alias. For example, “**db2agent** (DATA1)” is connected to the database alias “DATA1”.
- Attached to an instance. For example, “**db2agent** (user1)” is attached to the instance “user1”.

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

db2udfp

Fenced user-defined functions (UDFs) run outside of the firewall.

db2dari

Fenced stored procedures run outside of the firewall.

Database Threads/Processes

The following list includes some of the important threads/processes used by each database:

- **db2pfchr**, for input and output (I/O) prefetching
- **db2pclr**, for buffer pool page cleaners
- **db2loggr**, for manipulating log files to handle transaction processing and recovery
- **db2dlock**, for deadlock detection.

Database Server Threads/Processes

The system controller (**db2sysc**) must exist in order for the database server to function. Also, the following threads/processes may be started to carry out various tasks:

- **db2resyn**, the resync agent that scans the global resync list
- **db2gds**, the global daemon spawner on UNIX-based systems that starts new processes
- **db2wdog**, the watchdog on UNIX-based systems that handles abnormal terminations
- **db2fcmdm**, the fast communications manager daemon for handling internodal communication (used only in DB2 Enterprise - Extended Edition)
- **db2pdbc**, the parallel system controller, which handles parallel requests from remote nodes (used only in DB2 Enterprise - Extended Edition)
- **db2panic**, the panic agent, which handles urgent requests after agent limits have been reached at a particular node (used only in DB2 Enterprise - Extended Edition)

Differences between Intel and UNIX

The Intel systems supported by DB2 (OS/2 and Windows) differ from UNIX-based environments in that the database engine is multi-threaded, not multi-processed. In the Intel systems, each of the dispatchable units on the agent side of the firewall is a thread under the process **db2sysc**, allowing the database engine to let the operating system perform task-switching at the thread level and not the process level. For each database being accessed, there are other threads started to deal with database tasks (for example, prefetching).

Another difference is in the handling of abends. There is no need for a "watchdog" process in Intel systems, because these systems ensure that the allocated resources are cleaned up after an abnormal termination. Thus, there is no equivalent of the db2wdog process on the Intel systems. In addition, the db2gds process/thread is not needed on the Intel systems, which have their own mechanisms for starting threads.

Viewing System Threads and Processes

Some UNIX-based platforms (notably AIX) allow you to view the status of each DB2 process. Unfortunately, it is difficult to identify individual DB2 threads on the OS/2 and Windows platforms, because of the way threads are implemented on these operating systems. For these platforms, use the db2diag.log to diagnose if any parts of the database manager are failing. See "First Failure Data Capture" on page 157, for more information.

For a listing of threads/processes, use the following:

- On UNIX-based systems: the **ps** command. For more information, see "Viewing Process Status Using the ps Command" on page 212.
- On OS/2: the **pstat** command. For more information, see "Using the pstat Command" on page 225.
- On Windows NT Version 4.0: the Task Manager. For more information, see "Using Administrative Tools Provided by Windows NT" on page 230.
- On Windows 95 and Windows 98: the Performance Monitor for the Windows operating system, with the Process and ID Process variables. For more information, see "Using DB2-Supplied Tools" on page 231.

Chapter 15. Logged Information

This chapter describes the various logging facilities available on the DB2 server or client. Typically, these facilities are available for all platforms supported by DB2.

First Failure Data Capture

First Failure Data Capture (FFDC) information is diagnostic information captured automatically by DB2 about an error when the error occurs. This information reduces the need to reproduce errors to get diagnostic information.

The information captured by FFDC includes the following:

db2diag.log

When an error occurs, the `db2diag.log` is updated with information about the error. For more details, see “Interpreting the `db2diag.log`” on page 160. For information on the `DIAGLEVEL` setting, which determines how much information is logged in the file, see “Setting Configuration Parameters Affecting FFDC” on page 158.

db2alert.log

If an error is determined to be an alert, then an entry is made in the `db2alert.log` file and to the operating system or native logging facility. See “Alerts” on page 173 for more information.

Dump files

For some error conditions, extra information is logged in external binary dump files named after the failing process ID. These files are intended for DB2 Customer Support. See “Dump Files” on page 169, for more details.

Trap files

The database manager generates a trap file if it cannot continue processing because of a trap, segmentation violation, or exception. Trap files contain a function flow of the last steps that were executed before a problem occurred. See “Trap Files” on page 170, for more information.

This diagnostic information is contained in a single location (the `DIAGPATH` specified in the database manager configuration). It contains internal traces and dumps, which contain crucial information that may help you diagnose and fix problems.

Setting Configuration Parameters Affecting FFDC

The following database manager configuration parameters affect how much FFDC information is collected and where it is kept:

- *DIAGLEVEL* specifies the level of diagnostic information you will receive. The following data is captured for each *DIAGLEVEL* value:
 - *DIAGLEVEL* (0) - No diagnostic data (this level is not recommended)
 - *DIAGLEVEL* (1) - Severe errors only
 - *DIAGLEVEL* (2) - All errors (severe and not severe)
 - *DIAGLEVEL* (3) - All errors and warnings (the default value)
 - *DIAGLEVEL* (4) - All errors, warnings, informational messages, and other internal diagnostic information

The default for *DIAGLEVEL* is 3. Change it to 4 during initial setup and during those times when you are experiencing problems.

- *NOTIFYLEVEL* specifies the type of errors that are written. The following valid values for this parameter are:
 - *NOTIFYLEVEL* (0) - No diagnostic data (this level is not recommended)
 - *NOTIFYLEVEL* (1) - Severe errors only
 - *NOTIFYLEVEL* (2) - All errors (severe and not severe) (the default value)
 - *NOTIFYLEVEL* (3) - All errors and warnings
 - *NOTIFYLEVEL* (4) - All errors, warnings, informational messages, and other internal diagnostic information

The default for *NOTIFYLEVEL* is 2.

For a satellite server, errors are written to a notification file called *instance.nfy* where the *instance* is the DB2 instance name. For all other database server types, the *NOTIFYLEVEL* parameter is only available on the Windows NT platform where the errors are written to the Windows NT event log.

The error information can be written by DB2, the Capture and Apply programs associated with replication, and user applications. For a user application to write to the notification file or the Windows NT event log, it must use the `db2AdminMsgWrite` API. For more information about this API, see the *Administrative API Reference*.

- *DIAGPATH* specifies the fully qualified path where DB2 writes its diagnostic information. This directory contains dump, error log, the notify (*.nfy*) file when working with a satellite configuration, and alert log files, and is created automatically.

Note: On Windows NT, other DB2 diagnostic information is found in the event log which can be reviewed through the Windows NT Event Viewer.

The default for DIAGPATH is a null string. It is recommended that you use this default value or use a centralized location, especially if there are multiple database instances. By default, the FFDC information is placed in the following locations:

- For Windows and OS/2:
 - If the DB2INSTPROF environment variable is not set:
db2path\db2instance (where *db2path* is the path referenced in the DB2PATH environment variable, and *db2instance* is the environment variable containing the ID of the instance owner)
 - If the DB2INSTPROF environment variable is set:
x:\db2instprof\db2instance (where *x* is the drive referenced in the DB2PATH environment variable, *db2instprof* is the instance profile directory, and *db2instance* is the environment variable containing the ID of the instance owner)
- For UNIX-based environments: `$HOME/sqlllib/db2dump`, where `$HOME` is the home directory of the instance owner.

Note: You should clean out the DIAGPATH directory periodically.

To set these configuration parameters, use the UPDATE DATABASE MANAGER CONFIGURATION command from the command line processor.

Important: You must stop DB2 and start it again before any of these configuration changes take effect.

Examples

- To obtain the most diagnostic information, type:
`db2 "UPDATE DATABASE MANAGER CONFIGURATION USING DIAGLEVEL 4"`
- To set the diagnostic path to a new directory called `x:\tmp\errors`, type:
`db2 "UPDATE DATABASE MANAGER CONFIGURATION USING DIAGPATH x:\tmp\errors"`
- To reset the diagnostic path to the default, type:
`db2 "UPDATE DATABASE MANAGER CONFIGURATION USING DIAGPATH NULL"`

For information on database manager administration commands, refer to the *Administration Guide: Implementation*, the *Command Reference*, and the *Administrative API Reference* manuals.

Interpreting the db2diag.log

The db2diag.log is an ASCII file that contains information logged by DB2. It is located in the directory specified by the DIAGPATH database manager configuration parameter. Use a text editor to view the file on the machine where you suspect a problem to have occurred.

The information in the db2diag.log includes the following:

- The location reporting an error. Application identifiers allow you to match up entries pertaining to an application on the db2diag.logs of servers and clients.
- A diagnostic message (beginning with "DIA") explaining the reason for the error.
- Any available supporting data, such as SQLCA data structures and pointers to the location of any extra dump or trap files.

There are two types of entries in the db2diag.log:

- Administrative events. These entries are valuable, since they indicate whether events such as backups and restores started and finished.
- Error information. This information is only useful if you are trying to diagnose an external symptom, or if you have already determined what the error is and are looking for more information. For example, when an application receives an unexpected SQL code, or when a database crashes, the file can contain error information, including pointers to dump files.

If the database is behaving normally, this type of information is not important and can be ignored.

Notes:

- Check the end of the file for the most recent data, because new information is always appended to the bottom of the file.
- Entries always have a timestamp. If you know when an error occurred, look for the first entry in the file marked with this time.
- The db2diag.log **grows continuously**. When it gets too large, back it up and then erase the file. A new db2diag.log is generated automatically the next time it is required by the system.

Interpreting db2diag.log Entry Headers

The following example shows the header information for a sample db2diag.log entry.

```
2000-03-06-11.53.18.001160 1 Instance:payroll 2 Node:000 3
PID:44829(db2agent (SAMPLE)) 4 TID:352 5
Appid:*LOCAL.payroll.000306140834 6
lock_manager 7 sqlplrq 8 Probe:111 9 Database:SAMPLE 10
DIA9999E 11 An internal return code occurred. Report the following:
"0xFFFFE10E". 12
```

Legend:

- 1** A timestamp for the message.
- 2** The name of the instance generating the message.
- 3** For DB2 Enterprise - Extended Edition systems with a db2nodes.cfg file, the node generating the message. (If the db2nodes.cfg file is not used, the value is "000".)
- 4** Identification of the process generating the message. In this example, the message came from the process identified as 44829. The name of this process is db2agent and it is connected to the database named SAMPLE.
Note: If the application is operating in a DUOW environment, the ID shown is the DUOW correlation token.
- 5** Identification of the table generating the message. In this example, the message came from the table identified as 352.
- 6** Identification of the application for which the process is working. In this example, the process generating the message is working on behalf of an application with the ID *LOCAL.payroll.970317140834.
To identify more about a particular application ID, either:
 - Use the **db2 list applications** command on a DB2 UDB server or **db2 list dcs applications** on a DB2 UDB Connect gateway to view a list of application IDs. From this list, you can determine information about the client experiencing the error, such as its node name and its TCP/IP address.
 - Use the **db2 get snapshot for application** command to view a list of application IDs.
- 7** The DB2 component that is writing the message. For messages written by user applications using the db2AdminMsgWrite API, the component will read "User Application".
- 8** The name of the function that is providing the message. This function operates within the DB2 subcomponent that is writing the message. For messages written by user applications using the db2AdminMsgWrite API, the function will read "User Function".

To find out more about the type of activity performed by a function, look at the fourth letter of its name. In this example, the letter "p" in the function "sqlplr" indicates a data protection problem. (Logs could be damaged, for example.)

The following list shows some of the letters used in the fourth position of the function name, and the type of activity they identify:

- b** Buffer pools
- c** Communication between clients and servers
- d** Data management
- e** Engine processes
- o** Operating system calls (such as opening and closing files)
- p** Data protection (such as locking and logging)
- r** Relational database services
- s** Sorting
- x** Indexing

- 9** Identification of the internal error that was reported.
- 10** The database on which the error occurred.
- 11** Diagnostic message indicating that an internal error occurred.
- 12** Hexadecimal representation of an internal return code. For more information, see "Interpreting Hexadecimal Codes" on page 163.

Interpreting an SQLCA Structure

For severe errors, an SQLCA structure is dumped into the db2diag.log. For details on each of the SQLCA fields, see "Appendix B. SQL Communications (SQLCA)" on page 271.

The following diagram provides an example of a db2diag.log with an SQLCA dump.

```

1997-03-16-11.53.18.001160 Instance:payroll Node:000
PID:44829(db2agent (SAMPLE)) Appid:*LOCAL.payroll.970317140834
relation_data_serv sqlrerrlg Probe:17 Database:SAMPLE
DIA9999E An internal return code occurred. Report the following : "0xFFFFE101".
Data Title :SQLCA pid(14358) 1
sqlcaid : SQLCA sqlcab: 136 sqlcode: -980 2 sqlerrml: 0
sqlerrmc: 3
sqlerrp : sqlrita
sqlerrd 4: (1) 0xFFFFE101 5 (2) 0x00000000 (3) 0x00000000
(4) 0x00000000 (5) 0x00000000 (6) 0x00000000
sqlwarn : (1) (2) (3) (4) (5) (6)
(7) (8) (9) (10) (11)
sqlstate:

```

Legend:

- 1** Beginning of the SQLCA entry.
- 2** The SQL state (when negative, an error has occurred).
- 3** Any reason codes associated with the SQL error code.
- 4** Sometimes there are several errors leading to the final SQL error code. These errors are shown in sequence in the sqlerrd area.
- 5** The hexadecimal representation of an SQL error. See “Interpreting Hexadecimal Codes”, for more information.

Interpreting Hexadecimal Codes

On OS/2 and Windows systems, some db2diag.log or SQLCA error codes are byte-reversed. If they are in the form *ffff nnnn*, they can be interpreted as is. If they are in the form *nnnn ffff*, you must byte-reverse them before they are meaningful.

To do this, switch the first four characters with the last four characters, and then the fifth and sixth characters with the seventh and eighth characters.

For example, the error code "0ae6 ffff" is translated to "ffff e60a".

When you have the error code in a meaningful form:

1. Convert it to decimal format, using a hexadecimal conversion tool. If you can locate it in the *Message Reference*, then it is an SQL code.
2. If the decimal conversion of the error code is not an SQL code, it is a return code. See “Appendix A. DB2 Internal Return Codes” on page 253 for a list of return codes.

Some Examples for Interpreting the db2diag.log

The following examples illustrate how you can use the db2diag.log to diagnose problems.

Example 1

Problem: A client application is executing against the database SAMPLE. During execution, the application always terminates abnormally.

Approach: To find the solution to the problem, first find the location of the error. Because the error occurs on the client, start by looking for the error on the client. To do this, look at the db2diag.log on the client machine.

Assume that the db2diag.log on the client machine contains no entries. Look at the db2diag.log on the DB2 server machine to see if an error has occurred on the server.

Assume that the db2diag.log on the server contains the following entries:

```
1997-03-16-20.52.27.001160 Instance:payroll Node:000
PID:44829(db2sysc (SAMPLE)) Appid:*LOCAL.payroll.970317140834
buffer_pool_services sqlbStartPools Probe:0 Database:SAMPLE
Starting the database. 1
1997-03-16-20.52.42.001160 Instance:payroll Node:000
PID:44829(db2sysc (SAMPLE)) Appid:*LOCAL.payroll.970317140834
buffer_pool_services sqlbcres Probe:1100 Database:SAMPLE
DIA3726C A invalid page checksum was found for page "". 2
ZRC=FFFFE119 3
1997-03-16-20.52.42.001160 Instance:payroll Node:000
PID:44829(db2sysc (SAMPLE)) Appid:*LOCAL.payroll.970317140834
buffer_pool_services sqlbcres Probe:1100 Database:SAMPLE
Obj={pool:2;obj:10;type:0} State=x27 4
Data Title :SQLB_OBJECT_DESC pid(104) tid(109)
0200 0a00 0200 0a00 0000 0000 0000 0000 0000 .....
0000 002e e00c 0000 0000 0000 0000 0000 .....
0000 0000 0100 0000 2700 0000 0000 0000 .....
0000 0000 0000 0000 .....
Dump File : C:\SQLLIB\DB2\104109.dmp Data : SQLB_PAGE
```

Interpretation:

- 1** An initial connection is made on the database. From the heading information for this message we see the database is the SAMPLE database.
- 2** A message indicates that there is an invalid page.
- 3** ZRC=FFFFE119 is an internal DB2 return code for the condition described in **2**. (Fortunately, there was a message in addition to this code. Example 2 will show how a hexadecimal return code may have to be used to determine an error condition.)
- 4** This information tells which object has been found to be inconsistent.
 - The pool ID is 2, indicating a table space ID of 2.

- The object ID is 10, indicating a table ID of 10. The hexadecimal value is in the SQLB_OBJECT_DESC and is shown as “0a” which is decimal 10.
- The object type is 0, indicating a data object. (Object type 1 indicates an index object.)

Solution: Restore and roll forward the database. If this is not possible, contact DB2 Customer Support.

Note: This example has shown that an error with symptoms only on the client machine may be caused by an error occurring on the DB2 server machine.

Example 2

Problem: A DB2 client application receives the SQL1042C error message when connecting to the database SAMPLE.

Approach: To find the solution to the problem, first find the location of the error. Because the error occurs on the client, start by looking at the db2diag.log on the client machine.

Assume the db2diag.log on the client machine contains no entries. Look at the db2diag.log on the DB2 server machine to see if an error has occurred on the server.

Assume the db2diag.log on the server contains the following entries:

```
1997-03-16-08.59.34.001160 Instance:payroll Node:000
PID:55543(db2syscs (SAMPLE)) Appid:*LOCAL.payroll.970317140834
buffer_pool_services sqlbStartPools Probe:0 Database:SAMPLE
Starting the database. 1
1997-03-16-08.59.35.001160 Instance:payroll Node:000
PID:55543(db2syscs (SAMPLE)) Appid:*LOCAL.payroll.970317140834
data_protection sqlpresr Probe:0 Database:SAMPLE
Crash Recovery has been initiated. 2
1997-03-16-08.59.35.001160 Instance:payroll Node:000
PID:55543(db2syscs (SAMPLE)) Appid:*LOCAL.payroll.970317140834
data_protection sqlpgole Probe:30 Database:SAMPLE
A problem occurred while verifying a database log file S0000000.LOG
RC=0ae6 ffff 3
1997-03-16-08.59.35.001160 Instance:payroll Node:000
PID:55543(db2syscs (SAMPLE)) Appid:*LOCAL.payroll.970317140834
data_protection sqlpgilt Probe:101 Database:SAMPLE
DiagData
0ae6 ffff
1997-03-16-08.59.36.001160 Instance:payroll Node:000
PID:55543(db2syscs (SAMPLE)) Appid:*LOCAL.payroll.970317140834
data_protection sqlpgilt Probe:60 Database:SAMPLE
DiagData
0ae6 ffff
1997-03-16-08.59.36.001160 Instance:payroll Node:000
PID:55543(db2syscs (SAMPLE)) Appid:*LOCAL.payroll.970317140834
data_protection sqlpgasn Probe:915 Database:SAMPLE
Marked the database log as bad. 0000 0000 4
```

Interpretation:

- 1** An initial request to connect to database SAMPLE was received. This indicates that the client request could reach the database server and communications are working.
- 2** Database crash recovery started, indicating that the database SAMPLE was not stopped normally the last time it was in use.
- 3** An error occurred with log file S0000000.LOG. The error code is shown as "0ae6 ffff".

Error codes should be in the format *ffff nnnn*. However, OS/2 and Windows architecture may byte-reverse integers. Because the error code in this example is in the form *nnnn ffff*, you must byte-reverse it before it is meaningful. For more information, see "Interpreting Hexadecimal Codes" on page 163.

The error code in this example translates to *ffff e60a*. When translated to decimal form (-6646), it is not a valid SQL code, indicating that it is

a return code rather than an SQL code. Looking up return code E60A in “Appendix A. DB2 Internal Return Codes” on page 253, we see that the file does not exist.

Database restart and recovery requires all log files that were in use at the time the database went down. This db2diag.log entry indicates that the log file S0000000.LOG cannot be found in the expected location. Restart and recovery cannot proceed.

4 Because restart and recovery could not complete, DB2 marks the database as “bad” to ensure that complete restart and recovery are done before the database is used again.

Solution: The best option to resolve this problem is to restore from a backup. Because the log file is S0000000.LOG (rather than S0001005.LOG or some other value), the database has not been log-retain enabled (or may be new). Therefore, restoring from backup is the only method of recovery.

Example 3

Problem: During a database connection, the SQL1004C message is received. This message indicates that a file system is out of storage. How do we determine which file system is involved?

Approach: First examine the text of the error message itself. (Issue db2 ? sql1004c or see the error in the *Message Reference*.) The message indicates that there is insufficient storage on a file system to process the command.

Next examine the db2diag.log on the server to find out exactly which file system is full, and to check if there is additional information.

Assume that the db2diag.log on the server contains the following entries:

```
1997-03-16-08.40.42.001160 Instance:payroll Node:000
PID:66847(db2syscs (SAMPLE)) Appid:*LOCAL.payroll.970317140834
data_protection sqlpgifl Probe:105 Database:SAMPLE
DiagData
0cd6 ffff 1
1997-03-16-08.40.43.001160 Instance:payroll Node:000
PID:66847(db2syscs (SAMPLE)) Appid:*LOCAL.payroll.970317140834
data_protection sqlpgifl 2 Probe:540 Database:SAMPLE
0cd6 ffff
1997-03-16-08.42.55.001160 Instance:payroll Node:000
PID:66847(db2syscs (SAMPLE)) Appid:*LOCAL.payroll.970317140834
data_protection sqlpgifl Probe:105 Database:SAMPLE
DiagData
0cd6 ffff
```

Interpretation:

- 1** The function sqlpgifl encounters the error code ffff d60c (remember to byte-reverse any error codes that are in the form *nnnn ffff*). Converting this code to decimal does not yield a valid SQL code. Looking up return code D60C in “Appendix A. DB2 Internal Return Codes” on page 253, we see that the disk is full.
- 2** The sqlpgifl function repeatedly encounters the error. This function is a logging function (because the fourth character is “p”. See “Interpreting the db2diag.log” on page 160, for more information. The repeated failure of this function indicates that we should first examine the log path to check available space.

Solution:

1. Determine how much space is in the log path by using **dir** for Windows or **OS/2** and **df** for UNIX-based environments.
2. Determine how much space may be required for logging. The amount of space (in bytes) required for log files can range from

$$(\text{logprimary} * (\text{logfilsiz} * 4096) + 8192$$

to

$$((\text{logprimary} + \text{logsecond}) * (\text{logfilsiz} + 2) * 4096) + 8192$$

This calculation shows the range of space that may be required by the logs, assuming log retain is not enabled.

If log retain is enabled, the logs will continue to grow unless user exits are enabled. With log retain, ensure that the log path has as much space available as possible.

3. If you determine that the file system is full, do one of the following:
 - On UNIX-based environments, extend the size of the file system containing the logs.
 - Move the logs to another log path that has enough space available to contain the primary and secondary logs.
 - Examine your logging parameters to see if their values can be reduced (see the *Administration Guide: Implementation*).

When the db2diag.log does not contain enough information to solve a problem, you will need to perform a trace. For more information, see “Example of Using db2diag.log and the DB2 Trace” on page 181.

Dump Files

Dump files are created when an error occurs for which there is additional information that would be useful in diagnosing a problem (such as internal control blocks). Every data item written to the dump files has a timestamp associated with it to help with problem determination. Dump files are in binary format and are intended for DB2 Customer Support representatives. They should be collected and sent along with the db2diag.log.

When a dump file is created or appended to, an entry is made in the db2diag.log indicating the time and the type of data written. These db2diag.log entries resemble the following:

```
1997-03-16-11.53.18.001160 Instance:payroll Node:000
PID:44829(db2agent (SAMPLE)) Appid:*LOCAL.payroll.970317140834
relation_data_serv sqlrerrlg Probe:17 Database:SAMPLE
DIA9999E An internal return code occurred. Report the following : "0xFFFFE101".
Dump File: /home/db2/sqllib/db2dump/56772.000 Data : SECTION STMT 1
```

Legend:

- 1** In this UNIX example, SECTION STMT data is stored in a file named 56772.000 located in the /home/db2/sqllib/db2dump directory.

Notes:

- For DB2 Enterprise - Extended Edition systems with a db2nodes.cfg file, the file extension identifies the node number. For example, the following entry indicates that the dump file was created by a DB2 process running on node 10:
Dump File: /home/db2/sqllib/db2dump/56772.010 Data : SECTION STMT
- For UNIX-based systems, dump files might be created in core dump directories. These files are called **DB2 core files** and are specific to DB2. Collect these files in addition to the files specified in the db2diag.log. DB2 core files are located in the path \$HOME/db2dump/core_directory, where core_directory is the core path directory name. There is one directory for each process. Directory names start with the letter "c", followed by the process identifier (pid) number of the affected process. A name extension provides the node number (for multinode systems with a db2nodes.cfg file). For example:
 - \$HOME/db2dump/c56772.000 is a directory containing a core file for the process with pid 56772.
 - \$HOME/db2dump/c56772/c56772.010 is a directory containing a core file for the process with pid 56772. It was created by a DB2 process running on node 10.

Trap Files

Under severe error conditions, the DB2 system issues a signal (for UNIX-based platforms) or exception (for Intel-based platforms) to itself. Other names for this occurrence are segment violations or traps, depending on the platform.

All signals or exceptions initiated by DB2 are reported in a **trap file**, which contains a function flow of the last steps that were executed before the system stopped. Trap files are required by DB2 Customer Support. They are located in the directory specified by the DIAGPATH database manager configuration parameter. The first letter in their names is "t", followed by a process identifier (pid). The file extension is 000 for single processors, or the node number for multi-node systems with a db2nodes.cfg file.

Examples:

- t56772.000 is a trap file for the process with pid 56772.
- t56772.010 is a trap file for the process with pid 56772. It was created by a DB2 process running on node 10.

Gathering Stack Traceback Information on UNIX-Based Systems

On UNIX-based systems, DB2 can generate a **stack traceback** for when you intentionally stop a DB2 procedure. This diagnostic information may be required by DB2 Customer Support when it is not possible to generate a trap file through a signal or exception.

A stack traceback file lists active internal functions in the order that they were called by a process, and shows the last steps of the function flow before the process was stopped. Each file reports the single signal or exception that was issued by the DB2 engine to stop or interrupt the process.

To activate a stack traceback:

1. Issue the **db2_call_stack** command to activate stack traceback.
2. Stop your DB2 instance.

The stack traceback information is located in trap files. There is one file for each process. For DB2 Enterprise - Extended Edition systems, there are files for each process on each node.

(Stack trace back information is not available for the SINIX operating system.)

Notes for DB2 Enterprise - Extended Edition:

- For systems with multiple nodes, use `db2_all db2_call_stack` to activate stack traceback on every node.

- The **db2_call_stack** command should be run by the instance owner only. Before running the command, ensure that a `/tmp/$DB2INSTANCE` directory exists at each node. The instance owner can create this directory by issuing the following command:

```
db2_all "mkdir /tmp/$DB2INSTANCE"
```

Using the Administrative Tools Log

Errors and warnings generated by DB2's graphical administrative tools are logged to an ASCII file. This file contains entries from all of the tools, including the Control Center.

Notes

- This file grows continuously. Keep this in mind, if lack of disk space is a problem.
- You can specify where the log file is stored with the following command:


```
set DBTLOG_FILE=location
```

 - If *location* is a file name with a fully qualified path, the file is stored at that location.
 - If *location* contains a file name only, the file has that name, and is located in the directory specified by the `DB2INSTPROF` registry value. If this directory is read-only or does not exist, the file is located in the directory specified by the registry value `DB2PATH`.
 - If you do not use the **set DBTLOG_FILE** command or do not specify a value for *location*, the file's name is `dbtcc.log`, and it is stored in the directory specified by the `DB2INSTPROF` or `DB2PATH` registry values (as described above).

For example, the following command will store the log file in `d:\aelog.txt`:

```
set DBTRACE_FILE=d:\aelog.txt
```

- If a log file cannot be opened for any reason, then an error message indicating this will appear on the screen.
- Occasionally, DB2 Customer Support may require more information than is in this log and request a DB2 tools trace. See "Administration Tools Trace" on page 183, for more information.

Administrative Tools Log File Sample

The following sample illustrates an administrative tools log file. Note that each entry is written to the file on a single line. In this example, the entries are truncated to fit on the page.

```

*----- QDB2/2 SQL03010 Control Center Admin Engine Log Feb 17 09:48:58 1997 1 -----*
>ERR : 2 09:49:10 3 pid=22406 4 tid=3 5 [IBM][CLI Driver] SQL1032N No start 6
>SEV : 09:49:11 pid=22406 tid=1 Nonzero sqlcode at ..\admin\aecm.cpp(357) - SQL ERROR ...
*----- End of Control Center Admin Engine Log Feb 17 09:49:28 1997 7 -----*
*----- QDB2/2 SQL03010 Control Center Admin Engine Log Feb 17 09:49:28 1997 -----*
>ERR : 09:50:11 pid=22410 tid=1 SQL Command: select schemaname, owner, remarks from sys...
>ERR : 09:50:11 pid=22410 tid=1 [IBM][CLI Driver][DB2/2] SQL0204N "SYSCAT.SCHEMATA" ...
>Warn: 09:50:12 pid=22410 tid=1 Nonzero sqlcode at ..\admin\aelist.cpp(2097) - No data ...
*----- End of Control Center Admin Engine Log Feb 17 11:44:12 1997 -----*

```

Legend:

- 1** A line indicating when the tool was started
- 2** The type of message (Informational, Warning, Error, or Severe)
- 3** A timestamp
- 4** The process ID
- 5** The thread ID
- 6** An SQL code and message text (truncated in this example)
- 7** A line indicating when the tool was closed

Using Message Files

During DB2 operations such as importing, exporting, loading, binding, or restoring data, you can specify that message files be created to contain the error, warning, and information messages associated with the operations. Specify the name of these files with the **MESSAGES** parameter. For information, see the **IMPORT**, **EXPORT**, or **LOAD** commands in the *Command Reference* or in the *Data Movement Utilities Guide and Reference* manuals.

These message files are standard ASCII text files. To print them, use the printing procedure for your operating system; to view them, use any ASCII editor.

Note:

- You can only view the contents of a message file after the operation is finished.
- For the load, import, and export utilities, a message file contains only those messages associated with the last operation.
- Each message in a message file begins on a new line and contains information provided by the DB2 message retrieval facility.

Alerts

An **alert** is an error notification issued when a severe error occurs. It can be sent to a central machine, for the attention of network administrators.

Alerts are written to both the DB2 diagnostic log (db2diag.log) and to the alert log (db2alert.log). Like the db2diag.log, the db2alert.log file contains the following information:

- The location of an alert error condition
- A diagnostic message explaining the reason for any errors

Unlike entries in the db2diag.log, however, db2alert.log entries can be viewed more readily by system administrators and others from system-wide error logging facilities:

- For AIX and other UNIX-based environments, the syslog file. See “Using the System Error Log (syslog)” on page 207.
- For OS/2, FFST/2 and SNA alerts. See “Using the FFST for OS/2 Utility” on page 219.
- For Windows NT, the event log. See “Diagnostic Tools for Windows NT” on page 229.

SNMP Alerts

If the DB2 SNMP subagent is activated, any DB2 alerts will produce a corresponding SNMP trap. Simple Network Management Protocol (SNMP) is a widely available and well-accepted protocol for distributed management, and is supported by DB2 for OS/2, DB2 for Windows NT, and DB2 for AIX.

DB2 supports SNMP with agents and subagents:

- The SystemView product provides an **SNMP agent** for the MVS, OS/2, Windows NT, and AIX operating systems. The SystemView SNMP agent provides a Desktop Management Interface (DMI) service layer through which applications such as TME or NetFinity can access information about a particular machine. The SystemView Agent also provides DMI to Distributed Protocol Interface (DPI) mapping so that:
 - Subagents at a workstation can use DMI, or DPI via SNMP, to communicate information
 - Management applications can use DMI, or DPI via SNMP, to gather information

The SystemView agent must be at the DPI Version 2 level. It is shipped with TCP/IP Version 3, and with DB2 for Windows 95 and DB2 for Windows NT.

- The **DB2 SNMP subagent** is a resident DB2 program. When invoked, it attempts to connect to the SNMP agent and to register the RDBMS MIB

sub-tree to that agent. If the SNMP agent is not present, the DB2 SNMP subagent will periodically retry to connect and register to the SNMP agent. Once invoked and connected, the DB2 SNMP subagent generates alerts to an SNMP manager in the case of a severe DB2 error (for example, a condition that requires operator intervention).

The DB2 SNMP subagent complies with the Internet Engineering Task Force (IETF) standard (RFC 1697). It also supports the attributes of the Internet Engineering Task Force Relational Database (RDBMS) Management Information Base (MIB), also known as the IETF RDBMS MIB. The IETF RDBMS MIB is an emerging industry standard for management of relational databases (RFC 1697).

Invocation

DB2 has a built-in SNMP subagent designed to work with any SystemView agent installed on the same machine, using DPI via SNMP. To configure this subagent, use the **snmpcfg** command. (Alternatively, click on the **snmpcfg** icon in the IBM SystemView Agent folder that is available on Windows NT.)

On the window that appears, you can configure the SNMP subagent to apply to a specific community name ("public" is the default), and to send trap information from this community to particular IP addresses.

To start the SNMP agent and the DB2 SNMP subagent:

For Windows NT and OS/2:

1. Start DB2.
2. Start the SystemView agent SNMP daemon from the **SNMPD** icon in the IBM SystemView Agent folder, or with the **snmpd** command.
3. Start the DB2 SNMP subagent with the command **db2snmpd**.

To stop the DB2 SNMP subagent, use the command **db2snmpd -end**.

For AIX:

1. Start DB2.
2. Start the SNMP daemon agent with the **snmpd** command. You must have root authority to issue this command.
3. Start the DB2 SNMP subagent with the command **db2snmpd**.

To stop the DB2 SNMP subagent, use the command **db2snmpd -end**.

If you have problems starting the SNMP agent or the DB2 SNMP subagent, you should contact DB2 Customer Service. DB2 Customer Service will provide guidance on how to turn on DPI internal debugging and tracing. DPI tracing enables the display of a hexadecimal dump incoming and outgoing DPI packets.

There are two other parameters that may be used on the **db2snmpd**. They are:

```
db2snmpd -h <host name> -c <community>
```

The `host name` means the host or local host name that is associated with your SNMP setup. The default is the local host's `HOSTNAME`. You should replace the local host's `HOSTNAME` with whatever host or local host name that is associated with your SNMP configuration.

The `community` means the community to which the DB2 portion of the MIB will be accessible and the community used for reporting error conditions. The default is "public". You should replace "public" with whatever community you have set up for this purpose within your SNMP configuration.

Chapter 16. Traces

This chapter describes various tracing methods you can use to capture information about a recurring and reproducible problem. To use traces, you must set them up, re-create your problem and collect the data. Normally, you would use traces only when requested by DB2 Customer Support.

Using the DB2 Trace Facility (`db2trc`)

The DB2 trace facility (`db2trc` command) lets you trace events, dump the trace data to a file, and format the data into a readable form. You may be requested by DB2 Customer Support to take a trace if the `db2diag.log` is not enough to diagnose a problem.

DB2 trace information is stored either in memory or on disk. When it is being collected, the information is recorded in chronological order.

The amount of information gathered by a trace grows rapidly. Thus, you want to capture only the error situation and avoid any other activities (for example, starting the database manager instance with `db2start` or connecting to a database with `db2 CONNECT`). When taking a trace, reproduce the *smallest* scenario that can be re-created and capture it for further analysis.

The process of performing a trace has a global effect on the behavior of a DB2 instance. The degree of performance degradation is dependent on the type of problem and on how many resources are being used to gather the trace information.

Depending on your platform, there are various ways to perform a trace. For example, on UNIX-based systems, you must use the `db2trc` command.

Before you Begin

Before attempting to use the DB2 trace facility, consider the following:

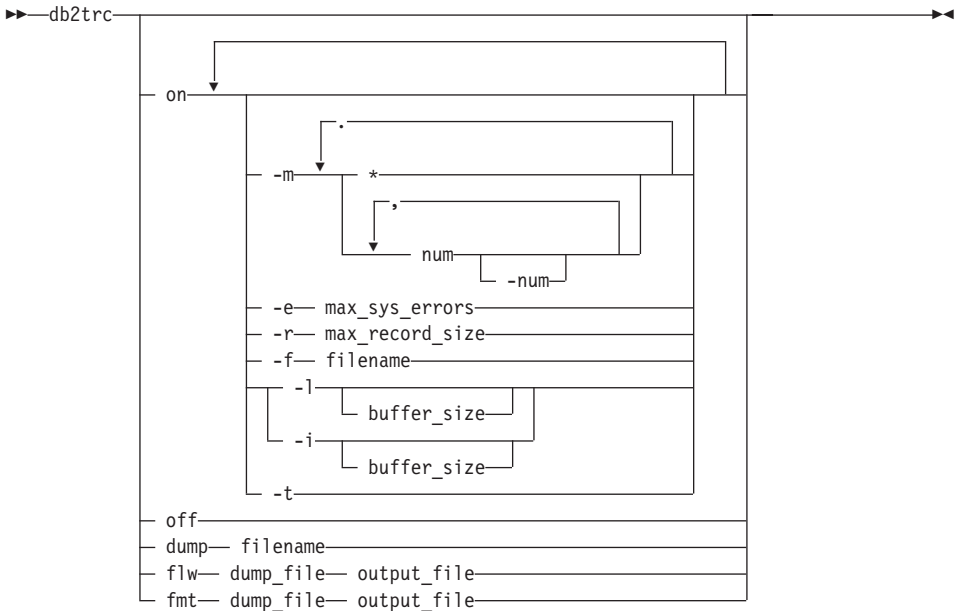
- Typically you will use the trace facility only when directed by DB2 Customer Support or by your technical support representative. Trace information should be sent to IBM for analysis **only when it is explicitly requested**.
- Only use this facility when error conditions are predictable and reproducible.
- It is recommended that system activity be at a minimum when using this utility, so that the problem can be more easily isolated.

- For DB2 Enterprise - Extended Edition, you must run a trace for each node.
- You need one of SYSADM, SYSCTRL or SYSMAINT authority.

Using Trace Facility (db2trc) Commands

The DB2 trace facility tracing is activated by the **db2trc on** command. This command has various parameters you can use to control the amount and type of data collected.

The syntax for the **db2trc** command is shown below. Explanations of the parameters follow.



Parameters

You must issue the **db2trc** command several times to turn tracing on, produce a dump file, format the dump file, and turn tracing off. The following list represents the order in which you use the parameters:

- on** Use this parameter to start the DB2 trace facility. See “Starting a DB2 Trace” on page 179, for information about the options for this parameter.
- dump** If you are tracing to memory, use this parameter to dump the trace information out to a file once you reproduce the error. The following command puts the information in the current directory in a file called db2trc.dmp:


```
db2trc dump db2trc.dmp
```

You must specify a file name with this parameter. The file is saved in the current directory unless you explicitly specify a path.

off After you dump the trace to a file, turn the trace off by typing:

```
db2trc off
```

flw | fmt

After you dump the trace to a binary file, confirm that the trace was successful by formatting it into an ASCII file. Use either the `flw` option (to sort by process or thread), or the `fmt` option (to list every event chronologically). For either option, you must specify the name of the dump file and should specify the name of an output file that will be generated. For example:

```
db2trc flw db2trc.dmp db2trc.flw
db2trc fmt db2trc.dmp db2trc.fmt
```

These options are provided for you to verify the trace. However, these files may also need to be sent to DB2 Customer Service and Support at their request.

Starting a DB2 Trace

To start the trace facility, type `db2trc on`. The default trace option values are:

-m *.*.*	Trace everything
-e -1	Collect all errors
-r 16000	Maximum record size is 16 KB
-s	Trace destination to shared memory (instead of to a file)
-l 2000000	Last trace records are retained, with a buffer size of 2 MB (the default buffer size is 512 KB instead for UNIX-based systems other than AIX)

Occasionally, you may be instructed to specify options to tailor the trace. Use the following options only as directed by DB2 Customer Support:

-m *mask*

Specifies trace record types to focus the search. The *mask* variable consists of four one-byte masks separated by periods. These masks correspond to products, event types, components, and functions, respectively, and act as a filter to accept or reject the trace record sent by DB2 for each event based on its ID.

If DB2 Customer Support requires mask values different from the default values, you will be instructed on which values to use.

-e *max_sys_errors*

Limits the number of DB2 internal system errors that the trace will hold to *max_sys_errors*. The default value is -1 (collect all errors).

-r *max_record_size*

Limits the size of trace records to *max_record_size* bytes. Longer trace records are truncated.

-s | **-f** *filename*

By default, trace output is stored in shared memory (the -s option). If the workstation is being suspended preventing you from accessing this output, you can trace to a file with the -f option. When using this output, you must specify a file where the trace output will be stored. See “Example of Tracing to a File” on page 181 for an example.

-l [*buffer_size*] | **-i** [*buffer_size*]

The option -l (lowercase “L”) specifies that the last trace records are retained (the first records are overwritten when the buffer is full). The option -i specifies that the initial trace records are retained (no more records are written to the buffer once it is full).

Use either of these options to specify the buffer size.

-t Includes timestamps. Applicable to UNIX-based environments only, where the logging of timestamps severely affects performance.

Verifying a DB2 Trace

Trace information may or may not be helpful in diagnosing an error. For example, it may not capture the error condition in the following situations:

- The trace buffer size you specified was not large enough to hold a complete set of trace events, and useful information was lost when the trace stopped writing to the file or wrapped.
- The traced scenario did not re-create the error situation.
- The error situation was re-created, but the assumption as to where the problem occurred was incorrect. For example, the trace was collected at a client workstation while the actual error occurred on a server.

To verify that a trace file can be read, format the binary trace file to show the flow control and send the formatted output to a null device. The following example shows the command to perform this task:

```
db2trc flw example.trc nul
```

The output for this command will explicitly tell you if there is a problem reading the file, and whether or not the trace was wrapped.

Examples of Using db2trc

The following examples provide several scenarios to show you how to use db2trc.

Example of Tracing to Memory

Tracing to memory is the preferred method of tracing. The size of the trace buffer is specified as 4 MB in this example.

1. Shut down all other applications.
2. Turn the trace on, using the following command:

```
db2trc on -l 4000000 -e -1
```
3. Reproduce the problem scenario. There should be as little DB2 activity as possible, to minimize event entries.
4. Dump the trace to a file, using the following command:

```
db2trc dump db2trc.dmp
```

(Dump the trace immediately after the problem occurs. Otherwise an SQL error that occurs after the error is captured may cause important information to be lost.)

5. Turn the trace off, using the following command:

```
db2trc off
```

Example of Tracing to a File

Tracing to a file is useful when the problem being re-created suspends the workstation, preventing you from dumping the trace to a file. When you trace to a file, each trace entry is written to disk when it happens. The information in the file captures those events leading up to (but not including) the suspension.

In this example, the size of the file is specified as 4 MB and the name of the file is db2trc.dmp:

1. Shut down all other applications.
2. Turn the trace on, using the following command:

```
db2trc on -l 4000000 -e -1 -f db2trc.dmp
```
3. Reproduce the problem scenario.

The db2trc.dmp file will be present when you reboot your workstation. It will contain the events leading up to the event that caused the system to crash.

Example of Using db2diag.log and the DB2 Trace

Problem: The SQL1042C error message is received when trying to connect to a database from a client. Access to the database is not possible.

Approach: To find the solution to the problem, first find the location of the error. Because the error occurs on the client, start by looking for the error on the client. To do this, look at the db2diag.log on the client machine. For more information, see “Interpreting the db2diag.log” on page 160.

Assume that the db2diag.log on the client machine contains no entries. Look at the db2diag.log on the DB2 server machine to see if an error has occurred on the server.

Assume that the db2diag.log on the server contains the following entries:

```
1997-03-16-08:54:37.001160 Instance:payroll Node:000
PID:74467(db2syscs (SAMPLE)) Appid:*LOCAL.payroll.970317140834
buffer_pool_services sqlbStartPools Probe:0 Database:SAMPLE
Starting the database. 1
1997-03-16-08:54:38.001160 Instance:payroll Node:000
PID:74467(db2syscs (SAMPLE)) Appid:*LOCAL.payroll.970317140834
data_protection sqlpresr Probe:0 Database:SAMPLE
Crash Recovery has been initiated. 2
1997-03-16-08:54:38.001160 Instance:payroll Node:000
PID:74467(db2syscs (SAMPLE)) Appid:*LOCAL.payroll.970317140834
data_protection sqlpresr Probe:0 Database:SAMPLE
Low transaction lsn: 0000 005d c00c
1997-03-16-08:54:38.001160 Instance:payroll Node:000
PID:74467(db2syscs (SAMPLE)) Appid:*LOCAL.payroll.970317140834
data_protection sqlpresr Probe:0 Database:SAMPLE
Minimum buffer lsn: 0000 005d c00c
1997-03-16-08:54:38.001160 Instance:payroll Node:000
PID:74467(db2syscs (SAMPLE)) Appid:*LOCAL.payroll.970317140834
data_management 5 sqldmund Probe:375 Database:SAMPLE
Error during undo. 3 0ae6 ffff 0ae6 ffff 0000 005e efa2 6363 4
```

Interpretation:

- 1** An initial request to connect to database SAMPLE was received. This indicates that the client request could reach the database server and communications are working.
- 2** The database is in a state where restart is required. That is, the database was not shut down cleanly the last time it was used.
- 3** An error was encountered when a transaction in the database had not been committed and was being rolled back ("undone") to make the database consistent.
- 4** The first dumped error code is ffff e60a (remember to byte-reverse any error codes that are in the form *nnnn ffff*). Converting this code to decimal does not yield a valid SQL code. Looking up return code E60A in "Appendix A. DB2 Internal Return Codes" on page 253, we see that a file does not exist.
- 5** Since the subcomponent that encountered the error is data management, we can predict that a data object is missing.

Approach: Unfortunately, the db2diag.log does not contain information to help us determine which file is missing. We must proceed by taking a trace of the error condition and scanning for the E60A error code in the trace file.

Assume that the trace file looks like the following:

```
3478 DB2 non-fatal_err oper_system_services sqloopenp (1.4.15.140)
      pid 55; tid 38; cpid 112; time 365535; trace_point 6
      433a 5c44 4232 5c53 514c 3030 3030 315c /DB2/SQL00001/
      5351 4c54 3030 3032 2e30 5c53 514c 3030 SQLT0002.0/SQL00
      3031 302e 4441 54 010.DAT 2
3479 DB2 cei_data oper_system_services sqloopenp (1.25.15.140)
      pid 55; tid 38; cpid 112; time 365535; trace_point 7
      ffff ffff
3480 DB2 cei_errcode oper_system_services sqloopenp (1.6.15.140)
      pid 55; tid 38; cpid 112; time 365535; trace_point 254
      return_code = 0xffffe60a 1 = -6646 = SQL0_FNEX
```

Interpretation:

- 1** Because trace file entries are ordered chronologically, we search backwards from the bottom of the trace file. We find the return code E60A in trace entry 3480.
- 2** Scanning backwards from trace entry 3480, we see that the file being opened was SQL00010.DAT in the table space SQLT0002.0

Solution: Restore from a backup known to be good, and roll forward to the end of the logs to ensure that no data is lost.

Taking Other Traces

There are other traces you can perform to diagnose problems. This section discusses:

- “Administration Tools Trace”
- “Taking a DB2 Connect Trace (ddcstrc)” on page 184
- “Taking a DRDA Trace (db2drdat)” on page 185
- “Taking an SNA Trace” on page 186
- “Using the DB2 CLI/ODBC/JDBC Trace Facility” on page 186.

Administration Tools Trace

Occasionally, DB2 Customer Support may request that you take an administration tools trace. This trace is intended for DB2’s graphical administrative tools (such as the Control Center, the Alert Center, and the Command Center), and provides additional information when these tools encounter a problem accessing or using a database. If you suspect the user interface itself to be the problem, take an ICLUI trace instead.

This trace is only recommended for problems that can be re-created. It provides detailed information on requests, connections, attachments to instances, CLI/ODBC/JDBC calls, DAS requests, API calls, and other activities that were happening when the problem occurs.

The format of the trace file is similar to the log file, except that more information is recorded. See “Using the Administrative Tools Log” on page 171, for more information.

Depending on how much information is written to it, the trace file can grow to the maximum file size imposed by the operating system. The file is overwritten the next time you perform an administration tools trace.

Invoking an Administration Tools Trace

To invoke the administration tools trace facility:

1. Ensure that the tool is closed.
2. Set tracing on with the following command:

```
set DBTRACE=Y
```
3. You can specify where the trace file will be stored with the following command:

```
set DBTRACE_FILE=location
```

 - If *location* is a file name with a fully qualified path, the file will be created at that location.
For example, the following command will send trace output to the file `d:\aetrace`:

```
set DBTRACE_FILE=d:\aetrace
```
 - If *location* contains a file name only, the trace file will have that name, and be located in the directory specified by the `DB2INSTPROF` registry value. If this directory is read-only or does not exist, the file is created in the directory specified by the registry value `DB2PATH`.
 - If you do not use the `set DBTRACE_FILE` command, or do not specify a value for *location*, the file's name is `dbttrace.trc`, and it is stored in the directory specified by the `DB2INSTPROF` or `DB2PATH` registry values.
4. Invoke the tool with its command. For example, use the `db2cc.exe` command to activate the trace for the Control Center.
5. Perform the procedure that caused the problem.
6. Close the tool.
7. Browse the trace file for diagnostic information. The file remains in the system until it is erased or overwritten by another trace.

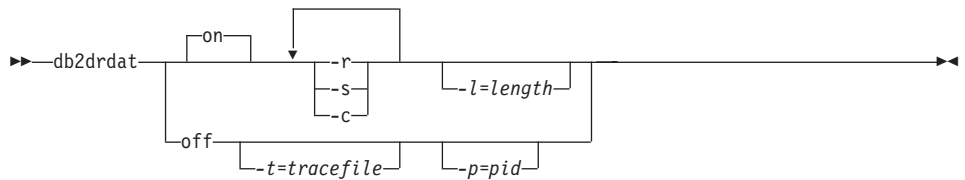
Taking a DB2 Connect Trace (ddcstrc)

The DB2 Connect trace (`ddcstrc`) is useful for diagnosing problems with the DB2 Connect product. For more information, see the *DB2 Connect User's Guide*.

Taking a DRDA Trace (db2drdat)

Use the DRDA trace (**db2drdat** command) to capture the DRDA data stream exchanged between a DRDA Application Requestor (AR) and the DRDA Application Server (AS). This tool is most often used for problem determination, but you can also use it for performance tuning in a client/server environment by determining how many sends and receives are required to execute an application. You need one of SYSADM, SYSCTRL, or SYSMOINT authority to use it.

The syntax for the **db2drdat** command is shown in the following diagram. Explanations of the parameters follow.



Parameters

- on** Turns on AS trace events (all if none are specified).
- off** Turns off AS trace events.
- r** Traces DRDA requests received from the DRDA AR. Logged information includes:
 - The type of DRDA request
 - The receive buffer
- s** Traces DRDA replies sent to the DRDA AR. Logged information includes:
 - The type of DRDA reply/object
 - The send buffer
- c** Traces the SQLCA received from the DRDA requestor. This is a formatted, easy-to-read version of *not null* SQLCAs.
Logged CPI-C error information includes:
 - Severity
 - Protocol used
 - API used
 - Local LU name
 - Failed CPI-C function
 - CPI-C return code
- l** Specifies the size of the buffer used to store the trace information.

- p Traces events only for the process with the specified process identifier (pid). If -p is not specified, all agents with incoming DRDA connections on the server are traced.
- t Specifies the destination file for the trace. If a file name is specified without a complete path, the path is determined from the current path.

Note: If *tracefile* is not specified, messages are directed to the `db2drdat.dmp` file in the current directory.

Notes:

- Do not issue the **db2trc** command while **db2drdat** is active. For information on **db2trc**, see “Using the DB2 Trace Facility (db2trc)” on page 177.
- The **db2drdat** command returns an exit code. A zero value indicates that the command completed successfully, and a nonzero value indicates that the command was not successful.
- If **db2drdat** sends the output to a file that already exists, the old file is erased unless the permissions on the file do not allow it to be erased. In this situation, the operating system returns an error.

Taking an SNA Trace

DB2 generates alerts to notify database administrators or network administrators of a serious database error or database server error. It uses two standards of alerts:

- SNA, for communication with MVS/ESA, VM, VSE, or AS/400 systems
- SNMP. For more information, see “SNMP Alerts” on page 173,

When an SNA alert is generated, it includes different kinds of information about the problem that occurred. You can use this information to help correct whatever is causing a problem with SNA and APPC communications.

For information on taking an SNA trace, use the keywords “SNA trace” to search the DB2 Product and Service Technical Library at: <http://www.ibm.com/software/data/db2/library/>. The search produces product documentation. Select the platform that is appropriate to your environment. Check the diagnosis, troubleshooting, or problem determination information and look up “tracing” or “SNA trace” from the index.

Using the DB2 CLI/ODBC/JDBC Trace Facility

Applications using CLI, ODBC, or SQLJ and the DB2 CLI Driver can have all the function calls traced to a plain text file. This can help with problem determination, database and application tuning or just to better understand what a third party application is doing.

The trace can be enabled at runtime using the CLI/ODBC Settings notebook, accessible from the Client Configuration Assistant if it is available, or by editing the `db2cli.ini` file directly. A CLI/ODBC/JDBC application can also enable the trace by setting the `SQL_ATTR_TRACE` and `SQL_ATTR_TRACEFILE` Environment or Connection attributes. These are the same attributes used by the tracing facility in the Microsoft ODBC Driver Manager.

Enabling the CLI/ODBC Trace Using the `db2cli.ini` File

The `db2cli.ini` file is located by default in the `\sql11ib\` path for Intel platforms, and the `/sql11ib/cfg/` path for UNIX platforms. The CLI/ODBC/JDBC configuration keywords used by the trace facility are:

- “TRACE” on page 199
- “TRACECOMM” on page 200
- “TRACEFILENAME” on page 200
- “TRACEFLUSH” on page 201
- “TRACEPATHNAME” on page 202
- “TRACETIMESTAMP” on page 203
- “TRACEPIDTID” on page 204

The following lines must be added to enable the trace. Keywords are NOT case-sensitive.

1. [COMMON]
2. Trace=1
3. TraceFileName=(fully qualified filename)
or
TracePathname=(fully qualified pathname)

Note: At least one of these two must be specified, but not both.

4. TraceFlush=(0 or 1) - optional
5. TraceComm=(0 or 1) - optional
6. TraceTimeStamp=(0, 1, 2, or 3) - optional
7. TracePidTid=(0 or 1) - optional

For example:

```
[COMMON]
trace=1
tracefilename=d:\temp\clitrace.txt
```

Setting **TRACE** to 0 turns tracing off. The trace file information can be left in the configuration file for the next time it is needed. See “TRACE” on page 199 for more information.

If the application does not exit or exits abnormally, the trace file will probably not be complete. Setting **TRACEFLUSH** to 1 will cause a flush to disk on

every function call (which will increase the overhead of tracing dramatically). See “TRACEFLUSH” on page 201 for more information.

To have information about each network request included in the trace file, set **TRACECOMM** to 1. See “TRACECOMM” on page 200 for more information.

If you want the timing information added to each line in the CLI trace, set **TRACETIMESTAMP** to 1, 2, or 3. Setting **TRACETIMESTAMP** to 1 provides the absolute time in seconds and milliseconds, followed by the timestamp. Setting **TRACETIMESTAMP** to 2 provides the absolute time in seconds and milliseconds. Setting **TRACETIMESTAMP** to 3 provides the timestamp.

If you want the process and thread identifier added to each line in the CLI trace, set **TRACEPIDTID** to 1.

Enabling the JDBC Trace Using the db2cli.ini File

The `db2cli.ini` file is located by default in the `\sqllib\` path for Intel platforms, and the `/sqllib/cfg/` path for UNIX platforms. Use the following CLI/ODBC/JDBC configuration keywords to enable a JDBC trace which provides more information than a regular CLI/ODBC trace. The CLI/ODBC/JDBC configuration keywords used by the trace facility are:

- “JDBCTRACE” on page 196
- “JDBCTRACEFLUSH” on page 197
- “JDBCTRACEPATHNAME” on page 198

The following lines must be added to enable the trace. Keywords are NOT case-sensitive.

1. [COMMON]
2. JDBCTrace=1
3. JDBCPathname=(fully qualified pathname)
4. JDBCFlush=(0 or 1) - optional

For example:

```
[COMMON]
jdbctrace=1
jdbcpathname=d:\temp
```

Setting **JDBCTRACE** to 0 turns tracing off. The trace file information can be left in the configuration file for the next time it is needed. See “JDBCTRACE” on page 196 for more information.

If the application does not exit or exits abnormally, the trace file will probably not be complete. Setting **JDBCTRACEFLUSH** to 1 will cause a flush to disk on every function call (which will increase the overhead of tracing dramatically). See “JDBCTRACEFLUSH” on page 197 for more information.

Locating the Resulting Files

If you used a fully qualified filename with the **TRACEFILENAME** keyword, you should have no problem locating the file. If you used a relative pathname, it will depend on what the operating system considers the current path of the application.

If you used a pathname instead of a filename with the **TRACEPATHNAME** or **JDBCTRACEPATHNAME** keyword, you will need to check the directory for a set of files created with the name set to the process id of the application and an extension that is a sequence number for each unique thread (for example, 65397.0, 65397.1, 65397.2). The file date and timestamp can be used to help locate the relevant file.

If you used a relative pathname, it will depend on what the operating system considers the current path of the application.

If there is no output file:

- Verify that the keywords are set correctly in `db2cli.ini`.
- Ensure the application is restarted. Specifically, `SQLAllocEnv()` must be called to read the `db2cli.ini` file and initialize the trace.
- Ensure the application has write access to the specified filename or path.
- Ensure that if you use **TracePathName** or **JDBCTRACEPATHNAME** then the path has already been created.
- Check if the `DB2CLIINIPATH` environment variable is specified. This environment variable changes the location from which the `db2cli.ini` file will be read.
- ODBC applications will not access the IBM CLI/ODBC/JDBC Driver until the first connect call. No trace entries will be written to the file until the application makes this connect call. See “ODBC Driver Manager Tracing” on page 195 for more details.

Reading the Trace Information

The purpose of the trace is to display the sequence of calls, the input and output arguments and the return code for each function called. The trace is intended for people familiar with the CLI/ODBC/JDBC function calls. Two important items, are the SQL statement text being executed, and any error messages that the application may not be reporting.

To locate:

- SQL statements:
Search the trace file for the strings “`SQLExecDirect`” and “`SQLPrepare`”, you will find the SQL Statement on the same line that contains the text and the “`—>`” input arrow (your editor may wrap the line).
- Errors: (queried by the application)

Search the trace file for "SQLError", the message text will be shown on the line that contains the string and the output arrow "<—".

Also, search for SQLGetDiagRec() or SQLGetDiagField() following the receipt of SQL_ERROR or SQL_SUCCESS_WITH_INFO return codes.

- Errors: (Ignored by the application)

Search for "Unretrieved error message=" This indicates that a previous call got an SQL_ERROR or SQL_SUCCESS_WITH_INFO return code, but that the application did not query for the error information.

Note: An application may expect some error messages, you should look at all the error messages in the trace file and try to determine the serious ones.

Detailed Trace File Format: Refer to the example trace file below. Note that the line numbers have been added for this discussion, and do NOT appear in the trace.

- Line 1: The build date and product signature is shown to assist IBM Service.
- Lines 2-3: First of a common two line sequence showing the arguments on input (—>) to the function call. Integer arguments may be mapped to a defined value like "SQL_HANDLE_ENV", output arguments are usually shown as pointers, with an "&" prefix.
- Lines 4-5: Two line sequence showing output (<—) results of the function call. Only output arguments are shown, and the return code on the second line following the (—>). Match this with the preceding input lines.
- Line 7: Example of an elapsed time on input. This is the time in the application between CLI/ODBC/JDBC function calls, shown in seconds.

Note: the granularity or accuracy of these timings vary between platforms.

- Line 8: Example of an elapsed time on output. This is the time in CLI/ODBC/JDBC spent executing the function.
- Lines 18-20: Both SQLDriverConnect() and SQLConnect() display the keywords set on both the input connection string and set in the db2cli.ini file.
- Line 23: The output statement handle is shown as 1:1, the first number represents the connection handle, the second the statement handle on that connection. This also applies to descriptor handles, but not to connection or environment handles, where the first number is always zero.
- Line 29: Example of SQL statement text for SQLPrepare().
- Line 43-44: Deferred arguments from SQLBindParameter() calls (lines 33 - 40). This is the data sent for each of the sql parameter markers (?) in the prepared statement (line 29).

- Lines 79-81: The output from the SQLFetch() call. (iCol = Column, rgbValue = data in char format, pcbValue=Length).
- Line 110: SQLError() output, showing message text. The pfNativeError is either the DB2 SQLCODE or -9999 if the error originated from CLI/ODBC/JDBC instead of the database server.
- Line 123: Shows an unretrieved error message. This is shown whenever a function is called using a handle which had a previous error, but was never retrieved by the application. It is effectively "lost" (to the application) at this point but is captured in the trace.

Example Trace File: This example has line numbers added to aid the discussion, line numbers do NOT appear in the trace.

```

1  Build Date: 97/05/13-Product: QDB2/6000 (4) - Driver Version: 05.00.0000
2  SQLAllocHandle( fHandleType=SQL_HANDLE_ENV, hInput=0:0,
   phOutput=&2ff7f388 )
3  —> Time elapsed - +1.399700E-002 seconds

4  SQLAllocHandle( phOutput=0:1 )
5  <— SQL_SUCCESS Time elapsed - +6.590000E-003 seconds

6  SQLAllocHandle( fHandleType=SQL_HANDLE_DBC, hInput=0:1,
   phOutput=&2ff7f378 )
7  —> Time elapsed - +1.120000E-002 seconds

8  SQLAllocHandle( phOutput=0:1 )
9  <— SQL_SUCCESS Time elapsed - +8.979000E-003 seconds

10 SQLSetConnectOption( hDbc=0:1, fOption=SQL_ATTR_AUTOCOMMIT, vParam=0 )
11 —> Time elapsed - +6.638000E-003 seconds

12 SQLSetConnectOption( )
13 <— SQL_SUCCESS Time elapsed - +1.209000E-003 seconds

14 SQLDriverConnect( hDbc=0:1, hwnd=0:0, szConnStrIn="DSN=loopback;
   uid=clitest1;pwd=*****", cbConnStrIn=-3, szConnStrOut=&2ff7e7b4,
   cbConnStrOutMax=250, pcbConnStrOut=&2ff7e7ae,
   fDriverCompletion=SQL_DRIVER_NOPROMPT )
15 —> Time elapsed - +1.382000E-003 seconds

16 SQLDriverConnect( szConnStrOut="DSN=LOOPBACK;UID=clitest1;PWD=*****";,
   pcbConnStrOut=38 )
17 <— SQL_SUCCESS Time elapsed - +7.675910E-001 seconds
18 ( DSN="LOOPBACK" )
19 ( UID="clitest1" )
20 ( PWD="*****" )

21 SQLAllocHandle( fHandleType=SQL_HANDLE_STMT, hInput=0:1,
   phOutput=&2ff7f378 )
22 —> Time elapsed - +1.459900E-002 seconds

23 SQLAllocHandle( phOutput=1:1 )
24 <— SQL_SUCCESS Time elapsed - +7.008300E-002 seconds

```

```

25 SQLExecDirect( hStmt=1:1, pszSqlStr="create table test(id integer,
    name char(20), created date)", cbSqlStr=-3 )
26     → Time elapsed - +1.576899E-002 seconds

27 SQLExecDirect( )
28     ← SQL_SUCCESS    Time elapsed - +1.017835E+000 seconds

29 SQLPrepare( hStmt=1:1, pszSqlStr="insert into test
    values (?, ?, current date)", cbSqlStr=-3 )
30     → Time elapsed - +5.008000E-003 seconds

31 SQLPrepare( )
32     ← SQL_SUCCESS    Time elapsed - +7.896000E-003 seconds

33 SQLBindParameter( hStmt=1:1, iPar=1, fParamType=SQL_PARAM_INPUT,
    fCType=SQL_C_LONG, fSQLType=SQL_INTEGER, cbColDef=4, ibScale=0,
    rgbValue=&20714d88, cbValueMax=4, pcbValue=&20714d54 )
34     → Time elapsed - +2.870000E-003 seconds

35 SQLBindParameter( )
36     ← SQL_SUCCESS    Time elapsed - +3.803000E-003 seconds

37 SQLBindParameter( hStmt=1:1, iPar=2, fParamType=SQL_PARAM_INPUT,
    fCType=SQL_C_CHAR, fSQLType=SQL_CHAR, cbColDef=20, ibScale=0,
    rgbValue=&20714dd8, cbValueMax=21, pcbValue=&20714da4 )
38     → Time elapsed - +2.649000E-003 seconds

39 SQLBindParameter( )
40     ← SQL_SUCCESS    Time elapsed - +3.882000E-003 seconds

41 SQLExecute( hStmt=1:1 )
42     → Time elapsed - +3.681000E-003 seconds
43 ( iPar=1, fCType=SQL_C_LONG, rgbValue=10, pcbValue=4, piIndicatorPtr=4 )
44 ( iPar=2, fCType=SQL_C_CHAR, rgbValue="-3", pcbValue=2, piIndicatorPtr=2 )

45 SQLExecute( )
46     ← SQL_SUCCESS    Time elapsed - +4.273490E-001 seconds

47 SQLExecute( hStmt=1:1 )
48     → Time elapsed - +5.483000E-003 seconds
49 ( iPar=1, fCType=SQL_C_LONG, rgbValue=10, pcbValue=4, piIndicatorPtr=4 )
50 ( iPar=2, fCType=SQL_C_CHAR, rgbValue="-3", pcbValue=2, piIndicatorPtr=2 )

51 SQLExecute( )
52     ← SQL_SUCCESS    Time elapsed - +1.299300E-002 seconds

53 SQLExecute( hStmt=1:1 )
54     → Time elapsed - +3.702000E-003 seconds
55 ( iPar=1, fCType=SQL_C_LONG, rgbValue=10, pcbValue=4, piIndicatorPtr=4 )
56 ( iPar=2, fCType=SQL_C_CHAR, rgbValue="-3", pcbValue=2, piIndicatorPtr=2 )

57 SQLExecute( )
58     ← SQL_SUCCESS    Time elapsed - +1.265700E-002 seconds

```

```

59 SQLExecDirect( hStmt=1:1, pszSqlStr="select * from test", cbSqlStr=-3 )
60     —> Time elapsed - +2.983000E-003 seconds

61 SQLExecDirect( )
62     <— SQL_SUCCESS   Time elapsed - +2.469180E-001 seconds

63 SQLBindCol( hStmt=1:1, iCol=1, fCType=SQL_C_LONG, rgbValue=&20714e38,
64             cbValueMax=4, pcbValue=&20714e04 )
65     —> Time elapsed - +5.069000E-003 seconds

66 SQLBindCol( )
67     <— SQL_SUCCESS   Time elapsed - +2.660000E-003 seconds

68 SQLBindCol( hStmt=1:1, iCol=2, fCType=SQL_C_CHAR, rgbValue=&20714e88,
69             cbValueMax=21, pcbValue=&20714e54 )
70     —> Time elapsed - +2.492000E-003 seconds

71 SQLBindCol( hStmt=1:1, iCol=3, fCType=SQL_C_CHAR, rgbValue=&20714ee8,
72             cbValueMax=21, pcbValue=&20714eb4 )
73     —> Time elapsed - +2.490000E-003 seconds

74 SQLBindCol( )
75     <— SQL_SUCCESS   Time elapsed - +2.795000E-003 seconds

76 SQLFetch( hStmt=1:1 )
77     —> Time elapsed - +2.660000E-003 seconds

78 SQLFetch( )
79     <— SQL_SUCCESS   Time elapsed - +9.200000E-003 seconds
80 ( iCol=1, fCType=SQL_C_LONG, rgbValue=10, pcbValue=4 )
81 ( iCol=2, fCType=SQL_C_CHAR, rgbValue="-3           ",
82   pcbValue=20 )
83 ( iCol=3, fCType=SQL_C_CHAR, rgbValue="1997-05-23", pcbValue=10 )

84 SQLFetch( hStmt=1:1 )
85     —> Time elapsed - +4.942000E-003 seconds

86 SQLFetch( )
87     <— SQL_SUCCESS   Time elapsed - +7.860000E-003 seconds
88 ( iCol=1, fCType=SQL_C_LONG, rgbValue=10, pcbValue=4 )
89 ( iCol=2, fCType=SQL_C_CHAR, rgbValue="-3           ",
90   pcbValue=20 )
91 ( iCol=3, fCType=SQL_C_CHAR, rgbValue="1997-05-23", pcbValue=10 )

92 SQLFetch( hStmt=1:1 )
93     —> Time elapsed - +4.872000E-003 seconds

94 SQLFetch( )
95     <— SQL_SUCCESS   Time elapsed - +7.669000E-003 seconds
96 ( iCol=1, fCType=SQL_C_LONG, rgbValue=10, pcbValue=4 )
97 ( iCol=2, fCType=SQL_C_CHAR, rgbValue="-3           ",
98   pcbValue=20 )

```

```

95 ( iCol=3, fCType=SQL_C_CHAR, rgbValue="1997-05-23", pcbValue=10 )
96 SQLFetch( hStmt=1:1 )
97     → Time elapsed - +5.103000E-003 seconds
98 SQLFetch( )
99     ← SQL_NO_DATA_FOUND   Time elapsed - +6.044000E-003 seconds
100 SQLCloseCursor( hStmt=1:1 )
101     → Time elapsed - +2.682000E-003 seconds
102 SQLCloseCursor( )
103     ← SQL_SUCCESS   Time elapsed - +6.794000E-003 seconds
104 SQLExecDirect( hStmt=1:1, pszSqlStr="select * foo bad sql", cbSqlStr=-3 )
105     → Time elapsed - +2.967000E-003 seconds
106 SQLExecDirect( )
107     ← SQL_ERROR   Time elapsed - +1.103700E-001 seconds
108 SQLError( hEnv=0:0, hDbc=0:0, hStmt=1:1, pszSqlState=&2ff6f19c,
           pfNativeError=&2ff6ed00, pszErrorMsg=&2ff6ed9c, cbErrorMsgMax=1024,
           pcbErrorMsg=&2ff6ed0a )
109     → Time elapsed - +2.267000E-003 seconds
110 SQLError( pszSqlState="42601", pfNativeError=-104,
           pszErrorMsg="[IBM][CLI Driver][DB2/6000] SQL0104N An unexpected
           token "foo bad sql" was found following "select * ".
           Expected tokens may include: "<space>". SQLSTATE=42601
111 ", pcbErrorMsg=163 )
112     ← SQL_SUCCESS   Time elapsed - +5.299000E-003 seconds
113 SQLError( hEnv=0:0, hDbc=0:0, hStmt=1:1, pszSqlState=&2ff6f19c,
           pfNativeError=&2ff6ed00, pszErrorMsg=&2ff6ed9c, cbErrorMsgMax=1024,
           pcbErrorMsg=&2ff6ed0a )
114     → Time elapsed - +2.753000E-003 seconds
115 SQLError( )
116     ← SQL_NO_DATA_FOUND   Time elapsed - +2.502000E-003 seconds
117 SQLExecDirect( hStmt=1:1, pszSqlStr="select * foo bad sql", cbSqlStr=-3 )
118     → Time elapsed - +3.292000E-003 seconds
119 SQLExecDirect( )
120     ← SQL_ERROR   Time elapsed - +6.012500E-002 seconds
121 SQLFreeHandle( fHandleType=SQL_HANDLE_STMT, hHandle=1:1 )
122     → Time elapsed - +2.867000E-003 seconds
123 ( Unretrieved error message="SQL0104N An unexpected token "foo bad sql"
           was found following "select * ". Expected tokens may
           include: "<space>". SQLSTATE=42601
124 " )
125 SQLFreeHandle( )
126     ← SQL_SUCCESS   Time elapsed - +4.936600E-002 seconds

```



```

127 SQLEndTran( fHandleType=SQL_HANDLE_DBC, hHandle=0:1, fType=SQL_ROLLBACK )
128     —> Time elapsed - +2.968000E-003 seconds

129 SQLEndTran( )
130     <— SQL_SUCCESS   Time elapsed - +1.643370E-001 seconds

131 SQLDisconnect( hDbc=0:1 )
132     —> Time elapsed - +2.559000E-003 seconds

133 SQLDisconnect( )
134     <— SQL_SUCCESS   Time elapsed - +8.253310E-001 seconds

135 SQLFreeHandle( fHandleType=SQL_HANDLE_DBC, hHandle=0:1 )
136     —> Time elapsed - +4.247000E-003 seconds

137 SQLFreeHandle( )
138     <— SQL_SUCCESS   Time elapsed - +4.742000E-003 seconds

139 SQLFreeHandle( fHandleType=SQL_HANDLE_ENV, hHandle=0:1 )
140     —> Time elapsed - +2.023000E-003 seconds

141 SQLFreeHandle( )
142     <— SQL_SUCCESS   Time elapsed - +4.420000E-003 seconds

```

Tracing Multi-Threaded or Multi-Process Applications

For the trace to be of any use for multi-threaded or multi-process applications, you will need to use the **TRACEPATHNAME** keyword. Otherwise the trace will be garbled if multiple threads or processes are writing to it simultaneously. See “TRACEPATHNAME” on page 202 for more information.

The files are created in the path specified with the name set to the process id of the application and an extension that is a sequence number for each unique thread (for example, 65397.0, 65397.1, 65397.2).

By having each thread write to its own file, no semaphores are needed to control access to the tracefile, which means tracing doesn’t change the behavior of a multi-thread application.

ODBC Driver Manager Tracing

It is useful to understand the difference between the ODBC trace provided by the ODBC Driver Manager and the DB2 CLI/ODBC driver (IBM ODBC Driver Tracing).

Output file formats are different. The distinction is that the ODBC trace will show the calls made by the application to the Driver Manager. The DB2 CLI trace shows the calls received from the ODBC Driver manager.

The ODBC driver manager may map application function calls to either different functions, different arguments or may delay the call.

One or more of the following may apply:

- Applications written using ODBC 2.0 functions that have been replaced in ODBC 3.0, will have the old functions mapped to the new ones by the ODBC Driver Manager.
- Some function arguments may have their values mapped from ODBC 2.0 values to equivalent ODBC 3.0 values.
- The Microsoft cursor library will map calls such `SQLExtendedFetch()` to multiple calls to `fetch`, and other supporting functions.

For these reasons you may need to enable and compare the output of both traces to get a clear picture of what is happening.

For more information, refer to the *Microsoft ODBC 3.0 Software Development Kit and Programmer's Reference*.

CLI/ODBC/JDBC Configuration Keywords

The CLI/ODBC/JDBC configuration keywords used by the trace facility are:

- "JDBCTRACE"
- "JDBCTRACEFLUSH" on page 197
- "JDBCTRACEPATHNAME" on page 198
- "TRACE" on page 199
- "TRACECOMM" on page 200
- "TRACEFILENAME" on page 200
- "TRACEFLUSH" on page 201
- "TRACEPATHNAME" on page 202
- "TRACETIMESTAMP" on page 203
- "TRACEPIDTID" on page 204

JDBCTRACE:

Keyword Description:

Turn on the DB2 JDBC trace facility.

db2cli.ini Keyword Syntax:

JDBCTRACE = 0 | 1

Default Setting:

No trace information is captured.

DB2 CLI/ODBC Settings Tab:

Service

See Also:

"JDBCTRACEFLUSH" on page 197, "JDBCTRACEPATHNAME" on page 198

Usage Notes:

When this option is on (1), JDBC trace records are appended to files in the subdirectory indicated by the JDBCTRACEPATHNAME configuration parameter.

For example, to set up a JDBC trace file that is written to disk after each trace entry:

```
[COMMON]
JDBCTRACE=1
JDBCTRACEPATHNAME=E:\JDBCTRACES\CLI
JDBCTRACEFLUSH=1
```

(This option is contained in the Common section of the initialization file and therefore applies to all connections to DB2.)

Hint Text used in the GUI:

Specify where and how the JDBC trace information is recorded. Select 'Flush after each entry' to write each entry to disk immediately. These options should only be used for debugging purposes; they will slow down the JDBC driver, and the trace information can grow quite large if left on for extended periods of time.

JDBCTRACEFLUSH:

Keyword Description:

Force a write to disk after each JDBC trace entry.

db2cli.ini Keyword Syntax:

JDBCTRACEFLUSH = 0 | 1

Default Setting:

Do not write after every entry.

DB2 CLI/ODBC Settings Tab:

Service

Only Applicable when:

the CLI/ODBC/JDBC JDBCTRACE option option is turned on.

See Also:

"JDBCTRACE" on page 196, "JDBCTRACEPATHNAME" on page 198

Usage Notes:

Set this option on (JDBCTRACEFLUSH = 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 JDBCTRACE CLI/ODBC/JDBC option is turned on. See the JDBC 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.

Hint Text used in the GUI:

Specify where and how the JDBC trace information is recorded. Select 'Flush after each entry' to write each entry to disk immediately. These options should only be used for debugging purposes; they will slow down the JDBC driver, and the trace information can grow quite large if left on for extended periods of time.

JDBCTRACEPATHNAME:

Keyword Description:

Subdirectory used to store individual DB2 JDBC trace files.

db2cli.ini Keyword Syntax:

JDBCTRACEPATHNAME = < Full subdirectory name >

Default Setting:

None

DB2 CLI/ODBC Settings Tab:

Service

Only Applicable when:

the JDBCTRACE option is turned on.

Not Applicable when:

See Also:

"JDBCTRACE" on page 196, "JDBCTRACEFLUSH" on page 197

Usage Notes:

Each thread or process that uses the same DLL or shared library will have a separate JDBC 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 JDBCTRACE option is turned on. This will be done automatically when you set this option in the CLI/ODBC/JDBC Configuration utility.

See the JDBCTRACE option for an example of using the various trace settings.

JDBC trace should only be used for debugging purposes. It will slow down the execution of the JDBC 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.

Hint Text used in the GUI:

Specify where and how the JDBC trace information is recorded. Select 'Flush after each entry' to write each entry to disk immediately. These options should only be used for debugging purposes; they will slow down the CLI/ODBC/JDBC driver, and the trace information can grow quite large if left on for extended periods of time.

TRACE:

Keyword Description:

Turn on the DB2 CLI/ODBC/JDBC 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" on page 200, "TRACEFLUSH" on page 201, "TRACEPATHNAME" on page 202

Usage Notes:

When this option is on (1), CLI/ODBC/JDBC 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/JDBC 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.)

Hint Text used in the GUI:

Specify where and how the CLI/ODBC/JDBC trace information is recorded. Select 'Flush after each entry' to write each entry to disk immediately. These options should only be used for debugging

purposes; they will slow down the CLI/ODBC/JDBC driver, and the trace information can grow quite large if left on for extended periods of time.

TRACECOMM:

Keyword Description:

Include information about each network request in the trace file.

db2cli.ini Keyword Syntax:

TRACECOMM = 0 | 1

Default Setting:

0 - No network request information is captured.

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:

the CLI/ODBC/JDBC TRACE option option is turned on.

See Also:

“TRACE” on page 199, “TRACEFILENAME”, “TRACEPATHNAME” on page 202, “TRACEFLUSH” on page 201

Usage Notes:

When **TRACECOMM** is set on (1) then information about each network request will be included in the trace file.

This option is only used when the **TRACE** CLI/ODBC/JDBC 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.

Hint Text used in the GUI:

Include information about each network request in the trace file.

TRACEFILENAME:

Keyword Description:

File used to store the CLI/ODBC/JDBC 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” on page 199, “TRACEFLUSH”, “TRACEPATHNAME” on page 202

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/JDBC 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. TRACEPATHNAME should be used instead of TRACEFILENAME for multi-threaded applications so that the thread information is kept separate.

CLI/ODBC/JDBC trace should only be used for debugging purposes. It will slow down the execution of the CLI/ODBC/JDBC 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.)

Hint Text used in the GUI:

Specify where and how the CLI/ODBC/JDBC trace information is recorded. Select 'Flush after each entry' to write each entry to disk immediately. These options should only be used for debugging purposes; they will slow down the CLI/ODBC/JDBC driver, and the trace information can grow quite large if left on for extended periods of time.

TRACEFLUSH:

Keyword Description:

Force a write to disk after each CLI/ODBC/JDBC 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 199, “TRACEFILENAME” on page 200,
“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/JDBC 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.

Hint Text used in the GUI:

Specify where and how the CLI/ODBC/JDBC trace information is recorded. Select ‘Flush after each entry’ to write each entry to disk immediately. These options should only be used for debugging purposes; they will slow down the CLI/ODBC/JDBC driver, and the trace information can grow quite large if left on for extended periods of time.

TRACEPATHNAME:**Keyword Description:**

Subdirectory used to store individual DB2 CLI/ODBC/JDBC 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 199, “TRACEFILENAME” on page 200,
“TRACEFLUSH” on page 201

Usage Notes:

Each thread or process that uses the same DLL or shared library will have a separate CLI/ODBC/JDBC 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/JDBC Configuration utility.

See the TRACE option for an example of using the various trace settings. It will be ignored if the CLI/ODBC/JDBC option TRACEFILENAME is used.

CLI/ODBC/JDBC trace should only be used for debugging purposes. It will slow down the execution of the CLI/ODBC/JDBC 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.

Hint Text used in the GUI:

Specify where and how the CLI/ODBC/JDBC trace information is recorded. Select 'Flush after each entry' to write each entry to disk immediately. These options should only be used for debugging purposes; they will slow down the CLI/ODBC/JDBC driver, and the trace information can grow quite large if left on for extended periods of time.

TRACETIMESTAMP:**Keyword Description:**

Adds timing information to each line of the DB2 CLI/ODBC/JDBC trace files.

db2cli.ini Keyword Syntax:

TRACETIMESTAMP = 0 | 1 | 2 | 3

Default Setting:

0 - No timestamp is displayed.

See Also:

“TRACE” on page 199, “TRACEFILENAME” on page 200,
“TRACEFLUSH” on page 201, “TRACEPIDTID”

Usage Notes:

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/JDBC Configuration utility.

If you want the timing information added to each line in the CLI trace, set **TRACETIMESTAMP** to 1, 2, or 3. Setting **TRACETIMESTAMP** to 1 provides the absolute time in seconds and milliseconds, followed by the timestamp. Setting **TRACETIMESTAMP** to 2 provides the absolute time in seconds and milliseconds. Setting **TRACETIMESTAMP** to 3 provides the timestamp.

Hint Text used in the GUI:

Specify where and how the CLI/ODBC/JDBC trace information is recorded. Select ‘Flush after each entry’ to write each entry to disk immediately. These options should only be used for debugging purposes; they will slow down the CLI/ODBC/JDBC driver, and the trace information can grow quite large if left on for extended periods of time.

TRACEPIDTID:**Keyword Description:**

Adds process and thread identification information to each line of the DB2 CLI/ODBC/JDBC trace files.

db2cli.ini Keyword Syntax:

TRACEPIDTID = 0 | 1

Default Setting:

0 - No process or thread identification information is displayed.

See Also:

“TRACE” on page 199, “TRACEFILENAME” on page 200,
“TRACEFLUSH” on page 201, “TRACETIMESTAMP” on page 203

Usage Notes:

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/JDBC Configuration utility.

If you want the process and thread identifier added to each line in the CLI trace, set **TRACEPIDTID** to 1.

Hint Text used in the GUI:

Specify where and how the CLI/ODBC/JDBC trace information is recorded. Select 'Flush after each entry' to write each entry to disk immediately. These options should only be used for debugging purposes; they will slow down the CLI/ODBC/JDBC driver, and the trace information can grow quite large if left on for extended periods of time.

Chapter 17. Diagnostic Tools for UNIX-Based Systems

Use the diagnostic tools described in this chapter to gather and process data that can help identify the cause of a problem you are having with your UNIX-based system. Once the data is collected, it can be examined by someone who is familiar with the problem, or provided to DB2 Customer Support for analysis.

The diagnostic tools available for UNIX-based systems include:

System error log (syslog)

DB2 logs error and warning conditions to the system error log (syslog). See “Using the System Error Log (syslog)”.

Core file

A core file is created when severe errors occur. It contains a memory image of the terminated process, and can be used to determine what function caused the error. See “Using the Core File” on page 210, for more information.

Process status utility (ps)

The **ps** command returns process status information about active processes to standard output. See “Viewing Process Status Using the ps Command” on page 212, for more information.

Other UNIX-based commands

UNIX-based systems provide various tools and utilities to diagnose problems. For a summary, see “Other Commands” on page 213.

Using the System Error Log (syslog)

DB2 logs severe error and warning conditions to the system error log (syslog). These entries are usually a subset of what appears in the db2diag.log. The exception is when a severe error occurs where the DB2 database manager was not able to write to the db2diag.log. When such errors occur, it is recommended that you look at the syslog file.

Entries are added to the syslog based on priority and on what facility caused the error or warning condition. For example, system security, the kernel, and system daemons can all cause entries to be logged.

Priority refers to the urgency of the message. Message priorities are as follows, listed from highest to lowest:

- Emergency

- Alert
- Critical
- Error
- Warning
- Notice
- Information
- Debug

DB2 adds alert messages to the system error log only for very severe situations (for example, if agents are killed or the architecture of a database is compromised). Alerts are also referenced in the `db2diag.log` and in the alert log (`db2alert.log`). For more information, see “First Failure Data Capture” on page 157.

Setting Up the System Error Log

To route alerts to the system error log, perform the following the steps:

1. Each machine can be configured to log various combinations of facilities and priorities. To begin logging DB2 error and warning conditions in the syslog, you must generate a log file by adding the following line to the `/etc/syslog.conf` file. This action requires that you have system authority.

```
user.warn    fully_qualified_file_name
```

where:

- *user* is the facility to log. This includes DB2 and any other applications logging messages with the same facility name.
- *warn* is the priority over which messages are logged. The syslog file logs error and warning conditions that are of equal and higher priority than this priority level. Available choices are:

alert Only alert messages are logged

err Error and alert messages are logged

warn Warning, error, and alert messages are logged

info Information, warning, error, and alert messages are logged

- *fully_qualified_file_name* is the file (along with its fully qualified path) where the messages will be logged and the SQLCA will be dumped. This file will **not** be created by the system; you must create it in the specified path.

2. The syslog daemon must be sent a signal to restart so that it will know about the new configuration in the `syslog.conf` file. Use:

- For AIX only:

```
refresh -s syslogd
```

- For AIX and other UNIX-based systems:

```
kill -1 pid_of_syslogd
```

where *pid_of_syslogd* is the process ID of the *syslogd* process. You can obtain this process ID by issuing the **ps -fu syslogd** command. You must have system authority in order to use the **kill -1** command.

3. Check to see if information is being logged into the syslog file by issuing:

```
ps -fu db2sysc
kill -36 db2sysc.process.id
```
4. Check the file at *fully_qualified_file_name* (as defined in the */etc/syslog.conf* file). If there is information in the file, then the system error log has been enabled to capture the information.
5. The log file may grow quickly, and you will have to reduce its size periodically. You must use **kill -1 pid_of_syslog** after you issue the following commands:

```
mv logfile logfile.old
touch logfile
```

Instead of using a **kill** command, for AIX you can include the following line in the **crontab** that you run as part of your regular system maintenance:

```
refresh -s syslogd
```

Understanding the System Error Log

Messages generated by the database manager appear in the */etc/syslog.conf* file.

An example of an entry in the system log is shown below. In this example, the entries are truncated to fit on the page.

```
Jun 18 15:02:53 bluj DB2[46827]: DB2(db2inst1.000 1)oper_system_services sqloobeep 2 reports: 3
Jun 18 15:02:53 4 bluj 5 DB2[46827 6]: extra symptom string provided: 7 RIDS/sqlsysc_
Jun 18 15:02:53 bluj DB2[46827]: data: 8 54686973 20697320 616e2065 78616d70
Jun 18 15:02:53 bluj DB2[46827]: data: 6c65206f 66206c6f 67676564 20646174
Jun 18 15:02:53 bluj DB2[46827]: data: 61
Jun 18 15:02:53 bluj DB2[46827]: 2 piece(s) of dump data provided... to file 9 /u/db2inst1/
Jun 18 15:02:53 bluj DB2[46827]: 1. 'DUMP EXAMPLE #1' has been dumped 10
Jun 18 15:02:53 bluj DB2[46827]: 2. 'DUMP EXAMPLE #2' has been dumped
```

Legend:

- 1** The instance name and, for DB2 Universal Database Enterprise - Extended Edition for the Solaris Operating Environment and DB2 Universal Database Enterprise - Extended Edition for Windows NT; systems with a *db2nodes.cfg* file, the node number.
- 2** The reporting component and function.
- 3** The first line continues with the probe ID and error and alert numbers.

4 A timestamp for when the event occurred.

5 The host name.

6 The process ID of the reporting process

Use the **ps** command to view information about the process id of the reporting process. For example, execute the following command to get information about the reporting process:

```
ps -fu 46827
```

7 A symptom string that contains additional information about where and why the problem occurred.

8 A hexadecimal dump of data that includes return codes and other information that can be interpreted by DB2 Customer Support.

9 Information about additional dump files. Larger structures and other binary data may be dumped to additional files. Typically these files are located in `/u/$DB2INSTANCE/sqllib/db2dump`. The name of the file is identified in the syslog file.

10 An entry to identify a piece of dump data.

Using the Core File

If a program terminates abnormally, a **core file** is created by the system to store a memory image of the terminated process. Errors such as memory address violations, illegal instructions, bus errors, and user-generated quit signals cause core files to be dumped.

The core file is named "core", and is placed in the directory where the application was running. Note that system core files are distinct from DB2 core files. See "Dump Files" on page 169 for information on DB2 core files.

Accessing Core File Information

The **dbx** system command helps you determine which function caused a core file to be created. This is a simple check that will help you identify whether the database manager or DB2 Connect is in error, or whether an operating system or application error is responsible for the problem.

Notes:

- To determine which function caused the core file dump to occur, enter the following command from a UNIX-based command prompt:

```
dbx program_name core_filename
```

where *program_name* is the name of the program that terminated abnormally, and *core_filename* is the name of the file containing the core file dump.

The `core_filename` parameter is optional. If you do not specify it, the default name "core" is used.

- To obtain symbolic information, compile the application using the "-g" option.
- To end the **dbx** command, type **quit** at the dbx prompt.
- For the HP-UX operating system, use the **xdb** command for similar function.
- On Version 4.1 of AIX, ensure that the full core option has been enabled using the **chdev** command or smitty.
- The **dbx** command provides much more function than is described in this section. To find out more, enter `man dbx` from a UNIX-based command prompt.

Example of the dbx Command

The following example shows how to use the **dbx** command to read the core file for a program called "main".

1. At a command prompt, enter:

```
dbx main
```

2. Output similar to the following appears on your display:

```
dbx version 3.1 for AIX.  
Type 'help' for help.  
reading symbolic information ...  
[using memory image in core]  
segmentation.violation in freeSegments at line 136  
136      (void) shmdt((void *) pcAddress[i]);  
(dbx) where  
freeSegments(numSegs = 2, iSetId = 0x2ff7f730, pcAddress = 0x2ff7f758, line  
136  
in "main.c"  
main (0x1, 2ff7f7d4), line 96 in "main.c"
```

In the above example, the name of the function that caused the core dump is "freeSegments". If the function name begins with "db2", "sql", or "ddcs", it may indicate an error in the database manager or DB2 Connect products.

The example also illustrates the use of the **where** subcommand. Enter **where** at the dbx prompt to display the program path to the point of failure. In this example, the error occurred at line 136 of freeSegments, which was called from line 96 in "main.c".

3. To end the **dbx** command, type **quit** at the dbx prompt.

Viewing Process Status Using the ps Command

The **ps** command is a UNIX-based system command that returns process status information about active processes (to standard output). Use it to look for discrepancies between DB2 processes that are running and DB2 processes that you expect to be there. (For a description of DB2 processes, see “Chapter 14. The DB2 Process Model” on page 151.)

Flags control the types of information displayed for each active process, and may be applied simultaneously to yield a cumulative effect. For information, use the **man ps** command from a system command prompt.

Example: To show all processes of the instance ID “svtldb” use:

```
ps -fu svtldb
```

The following sample shows typical output from this command. (Note that there would be additional processes shown for DB2 Enterprise - Extended Edition.)

```
svtldb 1 5112 2 27894 3 0 10:48:17 4 - 0:00 db2ipccm 5
svtldb 6135 27894 0 10:48:17 - 0:00 db2gds
svtldb 14329 27894 0 10:48:17 - 0:00 db2resyn
svtldb 15356 27894 0 10:48:17 - 0:00 db2snacm 6
svtldb 18682 27894 0 10:48:17 - 0:00 db2tccpm 6
svtldb 27894 31989 0 10:48:17 - 0:00 db2sysc 7
svtldb 33275 27894 0 10:48:17 - 0:00 db2tcpim 6
svtldb 39939 18682 7 10:48:19 - 0:00 db2agent
svtldb 56074 34761 2 10:48:37 pts/10 0:00 db2
```

Legend:

- 1** The instance ID
- 2** The process identifier (pid)
- 3** The parent process identifier
- 4** The timestamp
- 5** The name of the process. See “Chapter 14. The DB2 Process Model” on page 151 for a description of the names.
- 6** The communication listeners (in this sample, APPC and TCP/IP listeners and the TCP/IP interrupt manager) are up
- 7** The system controller process

For UNIX-based systems other than AIX and SCO OpenServer, the db2sysc process is the only process shown for all server-side processes

(for example, agents, loggers, page cleaners, and prefetchers). On Solaris, you can see these side processes with the command `/usr/ucb/ps axw`.

Other Commands

This section describes some essential UNIX-based commands for troubleshooting and performance monitoring. For details on any one of these commands, precede it with `man` on the command line.

AIX Troubleshooting Commands

The following AIX system commands are useful for DB2 troubleshooting:

errpt

The **errpt** command reports system errors such as hardware errors and network failures.

- For an overview that shows one line per error, use **errpt**
- For a more detailed view that shows one page for each error, use **errpt -a**
- For errors with an error number of "1581762B", use **errpt -a -j 1581762B**
- To find out if you ran out of paging space in the past, use **errpt | grep SYSVMM**
- To find out if there are token ring card or disk problems, check the errpt output for the phrases "disk" and "tr0"

lsps The **lsps -a** command monitors and displays how paging space is being used.

lsattr This command displays various operating system parameters. For example, use the following command to find out the amount of real memory on the node:

```
lsattr -l sys0 -E
```

This example lets you see the maximum number of processes per user.

xmperf

For AIX systems using Motif, this command starts a graphical monitor that collects and displays system-related performance data. The monitor displays 3-dimensional diagrams for each node in a single window, and is good for high-level monitoring. However, if activity is low, the output from this monitor is of limited value.

UNIX-Based Troubleshooting Commands

The following UNIX-based system commands are also useful for DB2 troubleshooting. These commands are for all UNIX-based systems, including AIX, unless otherwise noted.

- df** The **df** command lets you see if file systems are full.
- To see how much free space is in all file systems (including mounted ones), use **df**
 - To see how much free space is in all file systems with names containing "dev", use **df | grep dev**
 - To see how much free space is in your home file system, use **df /home**
 - To see how much free space is in the file system "tmp", use **df /tmp**
 - To see if there is enough free space on the machine, check the output from the following commands: **df /usr**, **df /var**, **df /tmp**, and **df /home**
- truss** Available for SVR4 UNIX-based environments such as Solaris Operating Environment, Siemens-Nixdorf, SCO OpenServer, and Silicon Graphics IRIX, this command is useful for tracing system calls in one or more processes. It is not available for AIX.
- SAM** Available for HP-UX, the System Administrative Management (SAM) tool provides information on hardware errors.
- pstack** Available for Solaris 2.5.1 or later, the `/usr/proc/bin/pstack` command displays stack traceback information. The `/usr/proc/bin` directory contains other tools for debugging processes that appear to be suspended.

Performance Monitoring Tools

The following tools are available for monitoring the performance of your UNIX-based system. For a full description of the commands, see the *AIX for RISC System/6000 Performance Monitoring and Tuning Guide*, or the equivalent guide for the system you are using.

vmstat

This command is ideal for monitoring paging rate, which can be found under the page in (pi) and page out (po) columns. Other important columns are the amount of allocated virtual storage (avm) and free virtual storage (fre).

This command is useful for determining if something is suspended or just taking a long time.

iostat This command is useful for monitoring I/O activities. You can use the read and write rate to estimate the amount of time required for certain SQL operations (if they are the only activity on the system).

This command is also useful for determining if something is suspended or just taking a long time.

netstat

This command lets you know the network traffic on each node, and the number of error packets encountered. It is useful for isolating network problems.

System file

Available for Solaris Operating Environment, the `/etc/system` file contains definitions for kernel configuration limits such as the maximum number of users allowed on the system at a time, the maximum number of processes per user, and the inter-process communication (IPC) limits on size and number of resources. These limits are important because they affect DB2 performance on a Solaris Operating Environment machine. See the *Quick Beginnings* for further information.

Commands for DB2 Enterprise - Extended Edition

For DB2 Enterprise - Extended Edition systems with a `db2nodes.cfg` file, you can run UNIX-based commands on all DB2 nodes by surrounding the commands with quotation marks and preceding them with one of the following prefix commands:

db2_all

Provides information for all logical nodes. The following example shows active applications for all logical nodes:

```
db2_all ";db2 LIST APPLICATIONS"
```

rah

Provides information for all physical nodes. This command is useful for filtering out multiple entries that might occur using `db2_all` when there are multiple logical nodes on machines. The following example lists the first three lines of the hardware error log for each physical node:

```
rah ";errpt | head -3"
```

The semicolon (;) improves performance by issuing commands simultaneously to all nodes. For information on other syntax for **db2_all** and **rah**, see the file `raw README` in the `misc` subdirectory of the `sql11ib` directory and the *Administration Guide: Implementation* in the appendix on this subject.

The following commands are also useful for DB2 Enterprise - Extended Edition systems:

spmon

If using multiple nodes on RS/6000 SP systems, you may need to check if the high performance switch (HPS) is running on all workstations.

To view the status of all nodes, use one of the following commands from the control workstation:

- **spmon -d** for ASCII output
- **spmon -g** for a graphical user interface

Alternatively, use the command **netstat -i** from a node workstation to see if the switch is down. If the switch is down, there is an asterisk (*) beside the node name. For example:

```
css0* 65520 <Link>0.0.0.0.0
```

The asterisk does not appear if the switch is up.

db2_call_stack

The **db2_call_stack** diagnostic tool writes the call stack for each DB2 process to a trap file. See “Trap Files” on page 170, for more information.

Chapter 18. Diagnostic Tools for OS/2

Use the diagnostic tools described in this chapter to gather data that can help identify the cause of a problem you are having with your OS/2 system. Once the data is collected, it can be examined by someone who is familiar with the problem or provided to DB2 Customer Support for analysis.

Some of the diagnostic tools available for OS/2 include:

OS/2 logging utility

The OS/2 logging facility provides a record of error information for OS/2 programs. See “Using the OS/2 Logging Facility” on page 218, for more information.

FFST for OS/2 and SNA alert routing

FFST for OS/2 allows you to capture error information at the point of occurrence and to route SNA generic alerts either to a specified dump file or over an SNA session to a host or to a LAN alert collection facility. For more information, see “Using the FFST for OS/2 Utility” on page 219 and “Taking an SNA Trace” on page 186.

pstat The **pstat** command lists status information on which processes are running. For more information, see “Using the pstat Command” on page 225.

WarpCenter

Available for Version 4 of OS/2 Warp Connect, the WarpCenter provides access to problem determination tools, including:

- Logging and tracing tools
- Browsers to see the hardware and software configuration, and FFST settings
- A problem determination tutorial

For more information, see your OS/2 Warp Version 4 documentation.

Standalone dump utility

The standalone dump utility records the contents of all physical memory. This utility, which runs independently of the OS/2 program, can be used to dump data when a problem is very difficult to reproduce, or when other methods of problem determination cannot solve the problem. It is seldom required by DB2 Customer Support.

Kernel debugging for advanced problem diagnosis

In severe situations, DB2 Customer Support can connect remotely to your site and use kernel debugging to diagnose the cause of a problem.

Using the OS/2 Logging Facility

The OS/2 logging facility logs historical error information to the system log file (SYSLOG). This information represents errors encountered by the OS/2 system and its programs, including DB2. Only the previous 64 KB of data is maintained in the SYSLOG file.

Throughout this section, the term *error log* refers exclusively to the OS/2 logging facility, which writes to the OS/2 system log (SYSLOG).

Preliminary Considerations

The CONFIG.SYS file specifies whether information in the OS/2 system log file is wrapped or extended when it becomes full. If it specifies "wrap", the file wraps when it is full, and new messages are stored over existing ones at the beginning of the file. If it specifies "extend", the size of the file is increased when it becomes full, and new error messages are appended to the end of the file.

The FFST for OS/2 utility also writes to the SYSLOG, and FFST for OS/2 records can be accessed by using the OS/2 logging facility. For more information, see "Using the FFST for OS/2 Utility" on page 219.

Using OS/2 Logging Facility Commands

Logging is enabled by statements in the CONFIG.SYS file that are added when the OS/2 program is installed. If logging has been disabled, see the following instructions to re-enable it.

Modifying the CONFIG.SYS File for Logging

To enable alert logging and write alerts to a file, the following two statements must be in the CONFIG.SYS file:

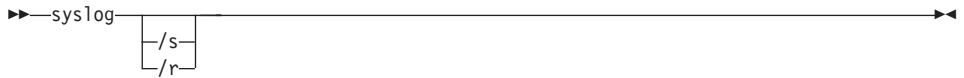
```
RUN=d:\OS2\EPW.EXE ON  
RUN=d:\OS2\EPWROUT.EXE -1
```

The drive d: represents the drive where FFST for OS/2 is installed. Note that the equal sign (=) cannot be immediately preceded or followed by a blank space.

Using the syslog Command

Use the **syslog** command to view the error log, or to suspend or resume logging. When issued without parameters, this command displays the OS/2 Error Log Formatter window, which shows a formatted version of the OS/2 system error log file and provides menus to perform actions such as suspending or resuming logging.

With parameters, the **syslog** command can be used directly to suspend or resume logging. Its syntax is as follows:



The optional parameter *s* suspends logging, while the optional parameter *r* resumes logging.

For information about the **syslog** command, type the following from an OS/2 command line:

```
help syslog
```

Notes

- When the OS/2 system log file is in use, it is locked. To make a copy of this file, you must switch to a different log file. From the OS/2 Error Log Formatter window, choose the **File** menu and then select **Change error log file**. Change to another error log, thereby unlocking the one you want to copy.
(You may also suspend system logging in order to unlock the file.)
- To choose an error log to view, select **Change error log file** from the **File** menu of the OS/2 Error Log Formatter window.
To choose the records you want displayed, select **Display Options** from the **Options** menu. You can specify a starting date and time, as well as a unique qualifier, to be more selective when viewing records in the system log.
- To print the contents of a system log file, select **Print** from the **File** menu of the OS/2 Error Log Formatter window.
The date format for the U.S.A. is printed out as MM/DD/YYYY, whereas the date format for the International Standards Organization is printed out as YYYY-MM-DD. For information about country specific date and time formats, see the *SQL Reference*.

Using the FFST for OS/2 Utility

The First Failure Support Technology/2 (FFST for OS/2) program captures error data at the point of occurrence, provides immediate problem notification to predefined locations, and furnishes unique error code identification. Because it remains passive until a software error is detected, its impact on system performance is minimal.

FFST for OS/2 provides the following functions:

- Error logging in a SYSLOG file called LOG00001.DAT. For more information about this file, see “Using the OS/2 Logging Facility” on page 218.

- System Network Architecture (SNA) generic alerts

Currently, FFST for OS/2 is accessed by LAN Adapter and Protocol Support (to log messages only) and by DB2 (to log error data and generic alerts).

Adding the FFST for OS/2 Program Group to the Desktop Manager Window

The FFST for OS/2 program group must exist on your Desktop Manager window. If it has not already been added, type the following command at the OS/2 command line:

```
EPWINST
```

This command presents you with a series of windows that allow you to specify the FFST for OS/2 configuration parameters and enter the workstation parameters. (To reset the parameters to the default values, select **Reset**.) After specifying the appropriate parameters in each window, select **OK**.

For more information about the FFST for OS/2 configuration parameters, see "Configuring and Enabling the FFST for OS/2 Utility".

Configuring and Enabling the FFST for OS/2 Utility

You can modify FFST for OS/2 operations and configuration with the **epw** command. For example, you can:

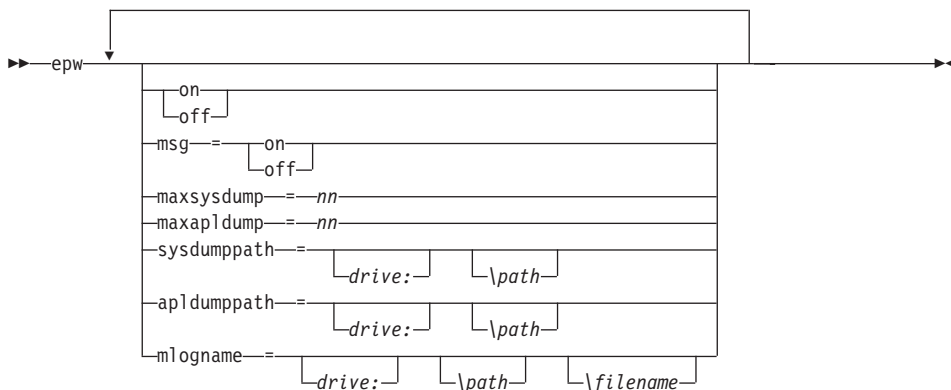
- Enable and disable FFST for OS/2
- Enable and disable the FFST for OS/2 message display
- Select the maximum number of dump data sets of a given type (either system dumps or application dumps) that may exist at one time in the current subdirectory for that type of dump
- Select the subdirectory in which to place FFST for OS/2 dump data files
- Select the path and file name of the FFST for OS/2 message log

FFST for OS/2 is enabled by the presence of the **epw** command in the CONFIG.SYS file. All of the command parameters are optional and may be specified in any combination or order.

This command operates in two modes:

- Initialization mode: FFST for OS/2 has not yet been enabled.
- Configuration mode: FFST for OS/2 is enabled (ON) and new values for the parameters are being specified.

The syntax of the FFST for OS/2 **epw** command follows:



ON | OFF

Specifies whether FFST for OS/2 will be enabled or disabled:

ON Enables all of the data-capturing capabilities of the FFST for OS/2 program, which include customized dump, error logging, generic alerts and message logging.

(This parameter is the default when you type EPW without specifying any parameters.)

OFF Disables the FFST for OS/2 program's data-capturing capabilities.

Notes:

1. In order for FFST for OS/2 to produce error logs, the operating system LOG function must be enabled. For more information, see "Using OS/2 Logging Facility Commands" on page 218.
2. The **syslog** command impacts operations. Enter **syslog /s** at the OS/2 command prompt to suspend FFST for OS/2 error logging, and **syslog /r** to resume it.

MSG={ON | OFF}

Specifies whether FFST for OS/2 will display messages:

ON Displays messages

OFF Suppresses message display

MAXSYSDUMP=nn

Specifies the maximum number of system software dumps that may occur in the current system dump directory. System software dump files are named OS2SYSxx.DMP, where xx ranges from 01 to nn.

Notes:

1. When FFST for OS/2 is initialized, xx is initialized to the index of the newest dump file in the current dump directory plus one. Whenever

xx exceeds *nn*, it is reset to 01. The value *nn* must be between 08 and 99. Its predefined default value is 32.

2. If the **epw** command is invoked in initialization mode and this keyword is not specified or is invalid, the default value of 32 will be used.
3. If the **epw** command is invoked in configuration mode and this keyword is not specified or is invalid, the current value remains unchanged.

MAXAPLDUMP=*nn*

Specifies the maximum number of software dumps that may occur in the current dump directory. System software dump files are named OS2APL*xx*.DMP, where *xx* ranges from 01 to *nn*.

Notes:

1. When FFST for OS/2 is initialized, *xx* is initialized to the index of the newest dump file in the current dump directory plus one. Whenever *xx* exceeds *nn*, it is reset to 01. The value *nn* must be between 8 and 99. Its predefined default value is 32.
2. If the **epw** command is invoked in initialization mode and this keyword is not specified or is invalid, the default value of 32 will be used.
3. If the **epw** command is invoked in configuration mode and this keyword is not specified or is invalid, the current value remains unchanged.

SYSDUMPPATH=[*drive:*][\path]

Specifies the drive and directory where a system software dump file should be placed once it is generated.

Notes:

1. If a drive is not included in the path, the boot drive will be used. The default value for the directory is OS2\SYSTEM. Dump file names cannot be set using this command.
2. If this command is invoked in initialization mode and this keyword is not specified or is invalid, the default value will be used.
3. If this command is invoked in configuration mode and this keyword is not specified or is invalid, the current value remains unchanged.

APLDUMPPATH=[*drive:*][\path]

Specifies the drive and directory where a software dump file will be placed once it is generated.

Notes:

1. If a drive is not included in the path, the boot drive will be used. The default value for the directory is OS2\SYSTEM. Dump file names cannot be set using this command.

2. If the **epw** command is invoked in initialization mode and this keyword is not specified or is invalid, the default value will be used.
3. If the **epw** command is invoked in configuration mode and this keyword is not specified or is invalid, the current value remains unchanged.

MLOGNAME=[drive:][\path][\file_name]

Specifies the fully qualified file name to be used for the message log file. If a drive is not included in the file name, the default drive will be used. (The default drive is the drive the system was booted from.)

If a directory is not specified, the default directory will be used. The default directory is OS2\SYSTEM.

If a file name is not specified, the default file name OS2MLOG.DAT will be used.

The first invocation of the **epw** command defines the values that will be used until the machine is rebooted or the values are reset.

For example, assume you enabled FFST for OS/2 by placing the following line in the CONFIG.SYS file:

```
RUN=C:\OS2\EPW.EXE MAXAPLDUMP=75
```

If you later disable FFST for OS/2 by issuing **EPW OFF**, the value for MAXAPLDUMP in a subsequent **EPW ON** command will be 75 rather than 32.

This applies to all of the **epw** keywords listed above.

Accessing FFST for OS/2 Error Records

Error records logged through FFST for OS/2 are recorded in the OS/2 System Error Log (SYSLOG). This is the same log to which the OS/2 logging facility writes error records. For information about the **syslog** command, see “Using the OS/2 Logging Facility” on page 218. You can also select **System Error Log** in the FFST for OS/2 program group window to access this log.

DB2 will always write a symptom record when using any of the FFST for OS/2 services. The symptom record includes:

- Hardware and software information
- A timestamp
- Error code information
- The name of the dump file, if one was produced
- A problem identifier generated by FFST for OS/2
- The message number and the first 32 characters of the message string if message services was requested (DB2 does not use this service)

- A symptom string, which uniquely identifies the error and is used by DB2 Customer Support to determine exactly which portion of DB2 caused the error to be logged.

This string is generated in the following format:

```
PIDS/cccccccc LVLS/111 RIDS/sssssss PCSS/nnnn PRCS/ppppppp
PRCS/scscscsc MS/mnmnmnm
```

- A message "ALERT: Y" if an alert was generated

Routing Generic Alerts

In order to route generic alerts over an SNA session to a host (such as SystemView) or to a LAN alert collection facility, you must enable the Generic Alert Router function of the FFST for OS/2 program and specify a destination. To do this, use the **epw rout** command.

Its syntax is as follows:

```
epw rout [Destination] [Option]
```

The parameters of this command are optional and positional. If only one parameter is specified, it is assumed to be the *Destination* parameter. This parameter is a numeric value to specify an action. It supports the following values:

- 1 The router is enabled, and alerts will be routed to a dump file called EPWALERT.DMP. The path is set by the SYSDUMPPATH parameter of the **epw** command.
- 0 The router is terminated if it is active.
- 1 The router is enabled, and alerts will be routed to the default destination. The alerts will be sent over an SNA session to the host using the SNA component of Communications Server. (If no *destination* value is specified, this value is the default.)
- 2 The router is enabled, and alerts will be routed over a LAN 802.2 session to a LAN alert collection facility. If you use this value, you may need to specify the *Option* parameter.

The *Option* parameter is needed only if the *destination* value is "2". It specifies the LAN adapter used to route alerts. If it is not specified, its value is assumed to be "0", and adapter 0 will be used. Values 0 through *n* are supported, where *n* is the number of LAN adapters available for your system. (The first LAN adapter is identified as Adapter 0.)

Start the router by placing a RUN= statement in the CONFIG.SYS file, or with the **start** or **detach** commands at the command line.

For example, to route the alerts to a dump file called EPWALERT.DMP, do one of the following:

- Include `RUN=EPWRROUT -1` in the `CONFIG.SYS` file
- Use the command `start epwrout -1`
- Run `detach epwrout -1` at the command line

Using the `pstat` Command

The `pstat` command returns information on which processes are running, and which threads within each process are running. This information includes current status (ready, blocked, and so on), and current priorities.

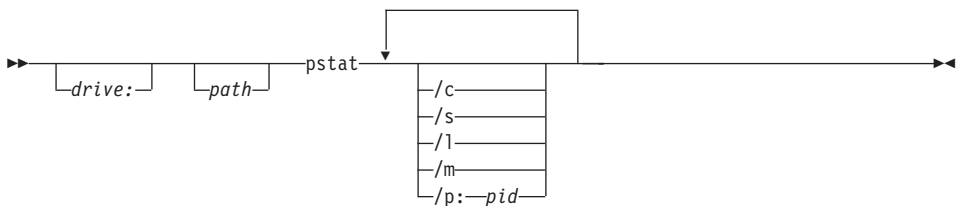
This command helps you determine:

- Which threads are suspended or waiting on a system event
- Which threads currently have lower priority compared to other threads (this may indicate poor performance)

For information on DB2 processes and threads, see “Chapter 14. The DB2 Process Model” on page 151.

The `pstat` command displays the process ID that has been assigned for tracing on a per-process basis and that is used to get global system information.

Use the `pstat` command from the OS/2 command prompt. Its syntax is as follows:



Parameters

If you use the `pstat` command without specifying any options, all process information will be displayed. If you want particular information only, use one or more of the following parameters:

- /C** This switch displays the current process- and thread-related information on the system. It returns the following information for each process:
 - Process ID
 - Parent process ID
 - Session ID
 - Process name

It returns the following information for each thread:

- Thread ID
- Thread priority
- Block ID
- Thread state (ready, blocked)

/S This switch displays system semaphore information for each thread on the system. It displays the following information for each thread:

- Process name
- Process ID
- Session ID
- Index
- Number of references
- Number of requests
- Flag
- System semaphore name

/L This switch displays the run-time libraries that are linked for each process on the system. It returns the following information for each process:

- Process name
- Process ID
- Session ID
- Library list

/M This switch displays the named shared-memory information for each process on the system:

- Handle
- Selector
- Number of references
- Shared memory name

/P: pid (where *pid* is the hexadecimal process ID to be displayed) This switch displays information related to the specified process. It returns the following information for the specified process ID:

- Process ID
- Parent process ID
- Session ID
- Process name
- Run-time link libraries
- Shared memory names

It returns the following information for each thread of the specified process ID:

- Thread ID
- Thread priority
- Thread state (ready, blocked)
- Block ID

- Owned semaphores

The following sample shows an example of **pstat** output:

Process and Thread Information							
Process ID	Parent Process ID	Session ID	Process Name	Thread ID	Priority	Block ID	State
0279	000A	12	E:\SQLLIB\BIN\DB2SYSC.EXE	01	0200	FE11E6C4	Block
				02	0200	FDFE7870	Block
				03	0200	FE11E544	Block
				04	0200	FE1191D8	Block
				05	0200	FDF58ECC	Block
				06	0200	FE034B24	Block
				07	0200	FE11DA34	Block
				08	0200	FE11DA24	Block
				09	0200	FE11DA58	Block
				0A	0200	FE11DA14	Block

Chapter 19. Diagnostic Tools for Windows 32-bit Operating Systems

Use the diagnostic tools described in this chapter to gather data that can help identify the cause of a problem when using Windows 32-bit operating systems. Once collected, the data can be examined by someone who is familiar with the problem or provided to DB2 Customer Support for analysis.

Diagnostic Tools for Windows 95 and Windows 98

The following diagnostic tools are available for Windows 95 and Windows 98 operating systems:

Dr. Watson

The Dr. Watson utility is invoked in the event of a General Protection Fault (GPF). It logs data that may help in diagnosing a problem, and saves this information to a file. You must start this utility explicitly by typing `drwatson` on the command line.

System tools

For Windows 95 and Windows 98, system information on memory allocation, hardware, and devices is available from the **System** icon in the Control Panel.

DB2-supplied tools

For Windows 95 and Windows 98 sessions, DB2 supplies administrative and development tools to help you identify DB2 problems. For more information, see “Using DB2-Supplied Tools” on page 231.

DB2 traces

DB2 traces provide some information on reproducible problems. For more information, see “Chapter 16. Traces” on page 177.

ODBC/CLI traces

CLI traces help to identify problems in CLI and ODBC applications. For more information, see the *CLI Guide and Reference* manual.

Diagnostic Tools for Windows NT

The following diagnostic tools are available for Windows NT systems:

Event log, performance monitor, and other administrative tools

The Windows NT Administrative Tools folder provides a variety of diagnostic information, including access to the event log and access to

performance information. For more information, see “Using Administrative Tools Provided by Windows NT”.

Task Manager

Available for Version 4.0 of the Windows NT operating system, the Task Manager shows all of the processes running on the Windows NT server, along with details about memory usage. Use this tool to find out which DB2 processes are running, and to diagnose performance problems. Using this tool, you can determine memory usage, memory limits, swapper space used, and memory leakage for a process.

To use Task Manager, place the cursor over the task menu, click on mouse button 2, and select **Task Manager** from the pop-up menu.

Dr. Watson

The Dr. Watson utility is invoked in the event of a General Protection Fault (GPF). It logs data that may help in diagnosing a problem, and saves this information to a file. You must start this utility by typing `drwatson` on the command line.

DB2-supplied tools

DB2 supplies administrative and development tools to help you identify DB2 problems. For more information, see “Using DB2-Supplied Tools” on page 231.

ODBC/CLI traces

CLI traces help to identify problems in CLI and ODBC applications. For more information, see the *CLI Guide and Reference* manual.

SNA server tracing

If you have the SNA server installed, it provides a tracing facility. To use it, go to the SNA Server window.

SNMP

SNMP allows you to capture error data at the point of occurrence. For more information, see “SNMP Alerts” on page 173.

Using Administrative Tools Provided by Windows NT

The Windows NT operating system provides administrative tools that you can use for problem and performance diagnosis. For Version 4.0, these tools are available from the **Administrative Tools** menu, accessible from **Programs** in the **Start** menu. For Version 3.5.1, they are available from the Control Panel.

For information on:

- System errors, select **Event Viewer**.

Severe DB2 errors are transmitted to the application event log, which is accessible from the event viewer. For more information, see “Using the Windows NT Event Viewer” on page 231.

- Performance statistics for the local machine and for remote connections, select **Performance Monitor**. Use this tool to find out and monitor information about connections, threads, and memory.
- User accounts, select **User Manager**.
- Diagnosing problems, select **Windows NT Diagnostics**. This tool provides information on the system and its resources, memory, network, and environment.
- Which version of the operating system you are using, select **OS Version**.
- The type of hardware your Windows NT session is operating on, select **Hardware level**.

Using the Windows NT Event Viewer

DB2 information is written to the Windows NT event log, which can be accessed by system administrators or other personnel. Because this information can be monitored remotely, critical errors can be detected earlier.

You can view the event log from the Event Viewer, accessible from the **Administrative Tools** menu. View the application log (rather than the security or system logs) to view DB2-related information.

Often, information in the event log will see the db2diag.log file. Therefore it is recommended that you save both files before you contact DB2 Customer Support.

Taking Screen Captures

A feature provided by Windows 32-bit operating systems is the ability to take screen captures. If there is a window with information that you want to keep, perform the following steps:

1. Make the window the active window.
2. Press down the following three keys at the same time: Ctrl-Alt-PrintScreen.
3. Open the Paintbrush program.
4. The screen capture is in the clipboard. Paste it into the Paintbrush program.
5. Save the image to a file as a bitmap, to be used for problem diagnosis.

Using DB2-Supplied Tools

DB2 supplies administrative and development tools to help you identify DB2 problems on Windows 32-bit operating systems:

- The DB2 database manager lets you identify remote connections, threads (with the Process and ID Process variables), and other performance-related information. For more information, see “The Database System Monitor” on page 18.

- The db2diag.log identifies serious problems. For more information, see “First Failure Data Capture” on page 157.

Chapter 20. GUI and Monitoring Tools

This chapter discusses how to troubleshoot GUI and monitoring tools. It includes the following topics:

- “Database System Monitor”
- “Control Center” on page 235
- “Troubleshooting Page” on page 238
- “Stored Procedure Builder” on page 238
- “DB2 Index Advisor” on page 239

Database System Monitor

Built into the database manager is the ability to collect data about its operation and performance, and that of the applications using it. There are two ways to access the monitor data collected by the database manager:

- Snapshot monitoring, which provides you with information for a specific point in time.
- Event monitors, which automatically log monitor data to files or to a named pipe when specific events occur. Event monitors allow you to collect information about transient events that are difficult to monitor through snapshots.

This chapter briefly describes these methods to help you diagnose the cause of poor system and application performance. For more specific information on the Database System Monitor, see the *System Monitor Guide and Reference*.

Using the Snapshot Monitor

Taking a snapshot gives you information for a specific point in time. A snapshot is a picture of the current state of activity in the database manager for a particular object or group of objects. Using the Snapshot Monitor you can make a specific snapshot requests to assist you with troubleshooting a problem. This section explains the how to use the Snapshot Monitor to capture this information.

The Snapshot Monitor provides two categories of information for each level being monitored:

- State information tells you:
 - The current status of the database
 - Information on the current or most recent unit of work
 - The list of locks being held by an application

- The status of an application
- The current number of connections to a database
- The most recent SQL statement performed by an application
- Run-time values for configurable system parameters
- Counters accumulate counts for activities from the time monitoring started until the time a snapshot is taken:
 - The number of deadlocks that have occurred
 - The number of transactions performed on a database
 - The amount of time an application has waited for locks

Many snapshot request types and logical data groupings can be created. Snapshots can be requested at different levels from tables through the database manager. For more information, see the *System Monitor Guide and Reference*. For a listing of snapshot request types and logical data groupings, see "Snapshot Requests" in the *System Monitor Guide and Reference*.

To make a snapshot request, use the **db2GetSnapshot()** command or invoke the `GetSnapshot()` API. The API allows you to specify several snapshot requests. For a coding example of `db2GetSnapshot()`, see "Making a Snapshot Request," in the *System Monitor Guide and Reference*.

A snapshot returns its data as a self-describing stream in the user-supplied buffer. Data is returned in logical groupings. Each item returned by a snapshot request contains fields that specify its size and type for processing. For more information, see "Reading the Snapshot" in the *System Monitor Guide and Reference*.

Using the Event Monitor

You can request the database manager to automatically log data when specific events occur. The event monitor writes out database system monitor data to either a file or a named pipe when one of the following events occurs:

- End of a transaction
- End of a statement
- A deadlock
- Start of a connection
- End of a connection
- Database activation
- Database deactivation
- End of a statement's subsection (when a database is partitioned)

An event monitor effectively provides the ability to obtain a trace of the activity on a database. This section briefly describes how to review the

contents and format of the trace produced by an event monitor. Using the CREATE EVENT MONITOR statement, you can modify the trace to help diagnose your problem.

Output Records

The output of an event monitor is a binary stream of logical data groupings that are exactly the same for both pipe and file event monitors. You can format this trace using the db2evmon productivity tool.

Records in a trace can be divided into the following types:

- Monitor information, which identifies the version level of the event monitor
- Prologue information, which is generated when an event monitor is activated
- Actual content information, which is generated as events occur
- Epilog information, which is generated when a database is deactivated

Each record includes the application handle and application ID. These allow you to correlate each record with the application for which the record was generated. The application handle (agent_id) is unique system-wide for the duration of the application. The application ID is a string identifier that includes a time stamp and is guaranteed to remain unique, even after stopping and restarting the database manager.

For more specific information, see "Output Records" in the *System Monitor Guide and Reference*.

Event monitors return data in a self-describing format. That allows you to perform several useful tasks:

- Read the log stream header
- Read the log header
- Read the data stream
- Swap bytes in numerical values

For more information, see "Reading an Event Monitor Trace" in the *System Monitor Guide and Reference*.

Control Center

The Control Center is used to manage systems, DB2 Universal Database instances, DB2 for OS/390 subsystems, databases, and database objects, such as tables and views. In the Control Center, you can display all of your systems, databases, and database objects and perform administration tasks on them.

If you have problems starting the Control Center, see "Control Center Installation and Configuration." You can find this document in the file `x:\sql11ib\cc\prime\readme.htm` where *x* is the drive where you installed DB2.

The Control Center Administration Engine Log (`db2cc.log`) records information about the errors generated while using the Control Center. The log is always active while the Control Center is active. The log file is kept in the home directory of the executable that invokes the Control Center, that is, in the `bin` subdirectory of the `sql11ib` subdirectory. The log can be viewed and updated using an ASCII file editor.

The log file records the error message type, a time stamp, a process identifier (PID), a thread identifier (TID), and an SQL error message. The process ID and the thread ID are used to identify the operating system that generated the log. By combining information from the log files with the Control Center trace information, the DB2 Service and Support personnel are able to determine which Control Center task caused the error. This information is only of use to the DB2 Service and Support personnel.

See "Taking Other Traces" on page 183 for more information on taking a Control Center trace.

If you encounter other problems using the Control Center, you can search for troubleshooting tips using the search utility for DB2 online information.

Using Search Utility for DB2 Online Information

Troubleshooting tips for the Control Center are available through the search utility for DB2 online information. To use the **Search** utility, perform the following steps:

1. Start the Control Center. (There are different ways to do this depending on your operating system. For example, you could:
 - Use the `db2cc` command on the command line
 - (On OS/2) Select the Control Center icon from the DB2 folder
 - (On Windows NT) Select **Start** → **Programs** → **DB2 for Windows NT** → **Control Center**.
2. Press **F1** to display online help.
3. Click **Search** near the top of the online help. This opens the DB2 online information search form.
4. Type `troubleshooting` in the search field.
5. Select **Online help** from the information list.
6. Click the **Search** button. This will open a list of search results.
7. Select a topic to view troubleshooting tips.

Troubleshooting Tips

This section describes how to deal with some frequently encountered problems faced by users working with the Control Center.

Online Help Not Coming Up on OS/2

Symptom

When you press **F1** or press the **Help** button on a window, the online help does not come up, or you get an error message and the online help does not come up.

Action

Ensure that the subdirectory that contains the Netscape executable (by default in <drive>\netscape\program\netscape.exe) is included in your PATH environment.

Online Help Not Coming Up on UNIX Based Systems

Symptom

When you press **F1** or press the **Help** button on a window, the online help does not come up, or you get an error message and the online help does not come up.

Action

Ensure that Netscape 4.51 is installed, and that the directory that contains the Netscape script file (for example, /usr/bin) is included in your PATH environment variable.

SQL4400N Received When Attempting Commands in the Control Center

Symptom

The error message SQL4400N is received when attempting to run certain commands through the Control Center. As an example, the BACKUP command may cause this error.

Possible Cause

The user ID being used does not have the correct authority on the DAS instance.

Action

The user must manually attach using a user ID which has the correct authority on the DAS. Validation of the manual attach does not get validated immediately. The Control Center stores the authority and uses it when attempting a particular task (like a backup).

DBA4745N Received When Attempting to Administer a EEE AIX V5.2 Server

Symptom

The error message DBA4745N is received when attempting to

administer a DB2 UDB Enterprise - Extended Edition for AIX Version 5.2 server from the Control Center on the client.

Possible Cause

The **db2ccmlst** process run by the DB2AS user ID was not started; and the **db2ccmsvr** port was not defined in the services file on the server.

Action

A port for graphical tool administration with an EEE system must be allocated. The **db2ccmlst** process must be started on each node.

Troubleshooting Page

Use the Information Center to find instructions, answers, and other information about DB2 Universal Database. The Troubleshooting page of the Information Center includes descriptions of error messages and associated recovery actions, as well as "how to" instructions for troubleshooting. The Web page of the Information Center provides links to IBM sites on the World Wide Web where you can find up-to-date information about problems and solutions.

To view information available on the Troubleshooting page:

1. Start the Control Center. (There are different ways to do this depending on your operating system. For example, you could:
 - Use the main product folder
 - Use the task bar in the Control Center
 - Use the **Help** menu of the Control Center
 - (On Windows NT) Select **Start** —>**Programs** —>**DB2 for Windows NT** —>**Control Center** —>**Information Center**.
2. Click the Troubleshooting page.
3. Click a category, problem, or message; then, click the **View** push button.

Stored Procedure Builder

You can use the DB2 Stored Procedure Builder (SPB) to assist in the creation of Java stored procedures. The SPB is a graphical application that supports the rapid development of DB2 stored procedures. There are design assistants within the SPB to guide you through basic design patterns, to help you create SQL queries, and to estimate the performance cost of invoking a stored procedure.

Using the Debug Settings notebook in the SPB, you can view all the stored procedures that you have the authority to change, add, or remove. You can also debug entries in the stored procedures debug table. If you have DBADM

authority, or if you created the selected stored procedure, you can grant authorization to other users so that they can debug the stored procedure. See the online help for more information on using the SPB to debug stored procedures.

DB2 Index Advisor

The DB2 Index Advisor is a tool to assist you in choosing an optimal set of indices for your table data. There are different ways to get to this tool:

- You can access this tool through the Control Center by selecting the Indexes folder, clicking mouse button 2, and selecting **Create -> Index using SmartGuide**.
- You can access this tool from the command line by entering `db2advls`.
- You can use self-directed methods involving the EXPLAIN modes and PREP options.
- You can use the Call Level Interface (CLI) when writing applications.

The Index SmartGuide is the recommended method to use because it is simple, straight-forward, and easy to use. It provides a visual interface for accessing the Advise Facility.

You can use the Index Advisor to evaluate current indexes or to recommend indexes to use for a particular set of SQL statements processed over a given period of time.

See the “SQL Advise Facility” section in the “SQL Explain Facility” chapter of the *Administration Guide: Implementation* manual for more information on how to use the Index Advisor.

Chapter 21. Table and Table Space Recovery

This chapter discusses how to troubleshoot table and table space problems. It addresses the following topics:

- “Recovery Concepts”
- “Dropped Table Point-in-Time Recovery” on page 243
- “Table Space OFFLINE” on page 245
- “Table Space Usability” on page 246.

Finally, there is a section that discusses the database directories where current information about the database objects (including table spaces and tables) is kept.

Recovery Concepts

There are some concepts you should be aware of when concerned with solving problems that may occur when backing up and recovering tables and table spaces. These concepts include:

- “Recoverable and Non-Recoverable Databases”
- “Database Logs” on page 242
- “Point of Recovery” on page 243

For additional conceptual information on logging and recovery, see *Administration Guide: Planning*.

For detailed information on backup, restore, roll-forward, and logging, see *Administration Guide: Implementation*.

Recoverable and Non-Recoverable Databases

If you can recreate data easily, the database holding that data can be a non-recoverable database. For example:

- Tables that hold data from an outside source that is used for read-only applications (and the data is not mixed with existing data) should be considered for placement within a non-recoverable database.
- Tables with small amounts of data. Here recovery is not a problem. Rather, there is just not enough logging done for the data to justify the added complexity of managing log files and rolling forward after a restore.
- Large tables where small numbers of rows are periodically added. Again, there is not enough volatility to justify managing log files and rolling forward after a restore operation.

If you cannot recreate data easily, the database holding that data should be a recoverable database. The following are examples of data that should be part of a recoverable database:

- Data that you cannot recreate. This includes data whose source is destroyed after the data is loaded, and data that is manually entered into tables.
- Data that is modified by application programs or workstation users after it is loaded into the database.

Database Logs

All databases have logs associated with them. These logs keep records of database changes. If a database needs to be restored to a point beyond the last full, offline backup, then logs are required to roll the data forward to the point of failure.

There are two types of DB2 logging: *circular* and *archive*, each providing a different level of recovery capability.

Circular logging is the default behavior when a new database is created. With this type of logging, only full, offline backups of the database are valid. As the name suggests, circular logging uses a "ring" of online logs to provide recovery from transaction failures and system crashes. The logs are used and retained only to the point of ensuring the integrity of current transactions. Circular logging does not allow you to roll forward a database through prior transactions from the last full backup. Recovery from media failures and disasters is done by restoring from a full, offline backup. All changes since the last backup are lost. The database must be offline (inaccessible to users) when a full backup is taken. Since this type of restore recovers your data to the specific point in time of the full backup, it is called *version recovery*.

Active logs are used during crash recovery to prevent a failure (system power or application error) from leaving a database in an inconsistent state. The RESTART DATABASE command uses the active logs, if needed, to move the database to a consistent and usable state. During crash recovery, changes recorded in these logs that were not committed because of the failure are rolled back. Changes that were committed but were not physically written from memory (buffer pool) to disk (database containers) are redone. These actions ensure the integrity of the database. The ROLLFORWARD DATABASE command may also use the active logs, if needed, during a point-in-time recovery, or a recovery to the end of the logs. Active logs are located in the database log path directory.

Archived logs are used specifically for roll-forward recovery. They can be:

online archived logs

When changes in the active log are no longer needed for

normal processing, the log is closed, and becomes an archived log. An archived log is said to be *online* when it is stored in the database log path directory.

offline archived logs

An archived log is said to be *offline* when it is no longer found in the database log path directory. You can also store archived logs in a location other than the database log path directory by using a user exit program. (For additional information, see “User Exit for Database Recovery” in the *Administration Guide: Implementation*.)

Point of Recovery

The version and roll-forward recovery methods provide different points of recovery. The version method involves making an offline, full database backup copy of the database at scheduled times. With this method, the recovered database is only as current as the backup copy that was restored. For instance, if you make a backup copy at the end of each day, and you lose the database midway through the next day, you will lose a half-day of changes.

In the roll-forward recovery method, changes made to the database are retained in logs. With this method, you first restore the database or table spaces using a backup copy; then you use the logs to reapply changes that were made to the database since the backup copy was created.

With roll-forward recovery enabled, you can take advantage of online backup and table space level backup. For full database and table space roll-forward recovery, you can choose to recover to the end of the logs, or to a specified point in time. For instance, if an application corrupted the database, you could start with a restored copy of the database, and roll forward changes up to just before that application started. No units of work written to the logs after the time specified are reapplied.

You can also roll forward table spaces to the end of the logs, or to a specific point in time.

Dropped Table Point-in-Time Recovery

This feature allows you to recover dropped tables to the table space. Prepare every table for Point-in-Time (PIT) recovery in a case before a table is dropped.

Accidentally dropping a table

Sometimes a table is accidentally dropped. Once committed, the drop statement cannot be rolled back. To restore the table you must perform a full database restore and roll forward to a point in time

prior to the drop. This action requires that the entire database be made unavailable until the end of the roll forward. The transactions made after the table drop will be lost. You cannot use the table space restore and roll forward.

If dropped table recovery is enabled

If dropped table recovery is enabled, when a DROP TABLE statement is executed, DB2 writes an additional log entry in the log files. This entry contains the following items:

- Name of the table being dropped
- Time stamp
- GXID (Global Transaction ID). This number is unique for each transaction and is constant across partitions.
- TID (Table Space ID)
- FID (Table ID)

History File Entry

DB2 also writes an entry to the history file, which contains the DDL statement used to re-create the table, as well as a copy of the log record. The administrative information provided by the history file includes the following:

- ROLLFORWARD
- ALTER TABLESPACE
- REORG
- RUNSTATS
- DROP TABLE information

To recover a dropped table

1. Obtain a special ID that identifies the dropped table. This ID can be obtained from the history file by the command **LIST HISTORY DROPPED TABLE ALL**. This command displays a list of tables that have been dropped, their names, the time at which they were dropped, a special ID string to identify the tables during the rollforward phase, and the DDL commands used to re-create the table.
2. Restore a backup taken before accidentally dropping the table.
3. ROLLFORWARD DATABASE ... RECOVER DROPPED TABLE TableID to /expdir. Rollforward to the end of logs using the option RECOVER DROPPED TABLE <tableID> TO <export_dir>, where Table ID is the string obtained in step (1) and export_dir is the directory where the output files are written.
4. CREATE TABLE. Re-create the table using the CREATE TABLE DDL statement obtained in step number (1).

5. Import the data exported during the roll forward phase into the table.

Notes:

1. Only a single dropped table can be recovered at a time. To recover multiple accidentally dropped tables, the recovery sequence will have to be attempted multiple times.
2. The RECOVERABLE TABLES option (ON or OFF) is only allowed on REGULAR table spaces
3. The table is not re-created; rather, the data in the dropped table is extracted from the log files and written in DEL ASC format to the specified export directory.

Table Space OFFLINE

Before DB2 Universal Database Version 6, if a regular, long, or temporary table space was damaged or missing and circular logging was used, a connection to the database failed. If the problem was not corrected, the database would have to be restored from a backup.

DB2 Universal Database Version 6 has addressed this shortcoming. When a regular, long or temporary table space is damaged or missing, the table space reverts to OFFLINE state and the connection to the database will succeed, even if circular logging is used. This change improves your ability to handle temporary table spaces, because now there is nothing to recover from a temporary table space, except in a REORG situation. For example, once a connection is established, there are ways to bring your database up:

- If you have multiple temporary table spaces, further operations requiring temporary table spaces will choose the good ones.
- If you have only one temporary table space and it is bad, right after the connection you can create a new temporary table space and drop the bad one.

Archive Logging

ARCHIVE LOGGING before DB2 Universal Database Version 6

If log retain is on (log archival), and a regular, long, or temporary table space is damaged or missing, the connect succeeds and the table space is put in a rollforward pending state.

If the table space is accessed while the database is up, an error is returned (SQL0290N Table space access is not allowed).

For regular tables spaces, dropping may be possible, even if this is the only user table space.

If a table spans multiple table spaces, dropping is not possible.

If it is a temporary table space and there are no others, then anything that requires a temporary table to be created will fail with an error. For example, if you try to create an index over a table, you may get "SQL0290N Table space access is not allowed". The reorganization of a table using this temporary table space may return "SQL2216N SQL error —290 was found while reorganizing a database table." If the temporary table space is the only one, you cannot drop it. DB2 prevents a user from dropping the only temporary table space in the database. In this case, the error message is "SQL0283N Temporary table space "TEMPSPACE1" cannot be dropped because it is the only temporary table space with a 4K page size in the database". The solution is to create a new temporary table space before dropping the old one.

ARCHIVE LOGGING after DB2 Universal Database Version 6

For log retain, the current log is applied without change, except OFFLINE status is added.

Table Space Usability

After fixing the problem that caused DB2 Universal Database to revert to an OFFLINE table space state, perform the following steps to return the state to ONLINE:

1. Disconnect all applications and connect to the database again. You will see that the table space is now fine and can be taken out of the OFFLINE state.
2. Use the ALTER TABLESPACE ... SWITCH ONLINE statement to bring the table space up while the rest of the database is still up and deployed.
 - If the table space can be brought up successfully after issuing the command, or if the table space was not in the OFFLINE state to begin with, DB2 will return a SQLCODE of 0.
 - If the table space cannot be brought up successfully because there are still problems with one or more of the containers, DB2 will return a SQLCODE of —293.

RESTART ... DROP PENDING TABLESPACE

Making a successful database restart

```
RESTART DATABASE database-alias
[USER user-name [USING password]]
[DROP PENDING TABLESPACES tablespace-name1,
tablespace-name2, tablespace-nameN]
```

In case of circular logging, a damaged or missing table space (either temporary or permanent) causes a restart database failure. Option

DROP PENDING TABLESPACES can be used by explicitly telling the database manager to make the database restart successfully, even if you may lose these table spaces.

A list of troubled table space names can be found in db2diag.log if a restart database fails because of container problems.

If there is only one temporary table space in the database and it is in DROP PENDING state, a new temporary table space must be created right after a successful database restart.

Restarting the database on a single node

If the database is only restarted on a single node within an MPP system, a message may be returned on a subsequent database query indicating that the database needs to be restarted on other nodes. This requirement occurs because the database partition on a node on which the query depends must also be restarted. Restarting the database on all nodes solves the problem.

RESTART ... DROP PENDING Table Space State

If a problem occurs with containers of table spaces, specified in DROP PENDING option of RESTART DATABASE command, during the restart process, the corresponding table space is in DROP PENDING state after the successful restart.

Once a table space is in DROP PENDING state, the only operation available is to drop it.

DROP PENDING table space state applies only to circular logging.

Table Space Usability

DROP TABLESPACE allows you to drop a table space that shares a table with other table spaces.

Dropping the table space

You can drop the table space that has a table spanned across several DMS table spaces. For example, if long data, indexes, and regular data of the same table are stored in different DMS table spaces, this command allows you to drop the related tables spaces at one time.

Accepting a list of one or more table space names

The DROP TABLESPACE statement accepts a list of one or more table space names (separated by commas). If one of the table spaces in the list to be dropped contains data that spans another table space, then that other table space must also be in the list. You cannot drop them, and an error is returned.

RESTART ... DROP PENDING

In DB2 Universal Database Version 6, new table space status is valuable to circular logging. For log retain, the current logic is applied without change. However, for circular logging, there are two situations to be considered.

Crash recovery is necessary

If the user can fix the table space, a later restart is successful. Otherwise, the table space is in DROP PENDING state right after the restart. This DROP PENDING state may apply to multiple table spaces if necessary. This change requires new options for the **RESTART** command:

```
db2 RESTART sample [USER user1 [USING passwd]]
      [DROP PENDING TABLESPACES (tsname1, tsname2, ...tsnameN)]
```

Crash recovery is not necessary

If the table space is in a consistent state, a new **CONNECT** tries to bring the table space online as usual. A successful table space rollforward drops its OFFLINE state if applicable.

Database Directories

When a database is created, DB2 creates a separate subdirectory to store control files (such as log header files) and to allocate containers to default table spaces. Objects associated with the database are not always stored in the database directory; they can be stored in various locations, including devices.

The database is created in the instance that is defined by the **DB2INSTANCE** environment variable, or in the instance to which you have explicitly attached (using the **ATTACH** command). For an introduction to instances, see "Creating an Instance" in the *Administration Guide: Implementation*.

The naming scheme used on UNIX based systems is:

```
specified_path/$DB2INSTANCE/NODEnnnn/SQL00001
```

The naming scheme used on OS/2 and the Windows operating systems:

```
D:\$DB2INSTANCE\NODEnnnn\SQL00001
```

where

- `specified_path` is the optional, user-specified location to install the instance.
- `NODEnnnn` is the node identifier in a partitioned database environment. The first node is `NODE0000`.
- `"D:"` is a "drive letter" identifying the volume on which the root directory is located.

SQL00001 contains objects associated with the first database created, and subsequent databases are given higher numbers: SQL00002, and so on.

The subdirectories are created in a directory with the same name as the database manager instance to which you are attached when you create the database. (On OS/2 and the Windows operating systems, the subdirectories are created under the root directory for a volume that is identified by a "drive letter".) These instance and database subdirectories are created within the path specified on the CREATE DATABASE command, and the database manager maintains them automatically. Depending on your platform, each instance might be owned by an instance owner, who has system administrator (SYSADM) authority over the databases belonging to that instance.

To avoid potential problems, do not create directories that use the same naming scheme, and do not manipulate directories that have already been created by the database manager.

Database Files

The following files are associated with a database:

File Name	Description
SQLDBCON	This file stores the tuning parameters and flags for the database. See <i>Administration Guide: Performance</i> for information about changing database configuration parameters.
SQLLOGCTL.LFH	This file is used to help track and control all of the database log files.
Syyyyyyy.LOG	<p>Database log files, numbered from 0000000 to 9999999. The number of these files is controlled by the <i>logprimary</i> and the <i>logsecond</i> database configuration parameters. The size of the individual files is controlled by the <i>logfilsiz</i> database configuration parameter.</p> <p>With circular logging, the files are reused and the same numbers remain. With archive logging, the file numbers increase in sequence as logs are archived and new logs are allocated. When 9999999 is reached, the number wraps.</p> <p>By default, these log files are stored in a directory called SQLLOGDIR. SQLLOGDIR is found in the SQLn subdirectory.</p>
SQLINSLK	This file helps to ensure that a database is used by only one instance of the database manager.

- SQLTMPLK** This file helps to ensure that a database is used by only one instance of the database manager.
- SQLSPCS.1** This file contains the definition and current state of all table spaces in the database.
- SQLSPCS.2** This file is a backup copy of SQLSPCS.1. Without one of these files, you will not be able to access your database.
- SQLBP.1** This file contains the definition of all buffer pools used in the database.
- SQLBP.2** This file is a backup copy of SQLBP.1. Without one of these files, you will not be able to access your database.
- DB2RHIST.ASC** This file is the database history file. It keeps a history of administrative operations on the database, such as backup and restore operations.
- DB2RHIST.BAK** This file is a backup copy of DB2RHIST.ASC.

Notes:

1. Do *not* make any direct changes to these files. They can only be accessed indirectly using the documented APIs and by tools that implement those APIs, including the command line processor and the Control Center.
2. Do not move these files.
3. Do not remove these files.
4. The only supported means of backing up a database or a table space is through the **sqlubkp** (Backup Database) API, including the command line processor and Control Center implementations of that API.

Part 3. Appendixes

Appendix A. DB2 Internal Return Codes

The following list is provided to help you interpret some of the hexadecimal return codes you may encounter when analyzing a db2diag.log.

The list represents a substantial set of the return codes you may encounter. If you encounter a return code that is not in the list, contact DB2 Customer Support.

Return codes should be in the form "ffff nnnn", where nnnn represents one of the values listed in the following chart. If return codes appear in the form "nnnn ffff" instead, you must byte-reverse them. For more information, see "Interpreting Hexadecimal Codes" on page 163.

Table 1. DB2 Internal Return Codes

Internal Return Code	SQL State	Reason Code	Description
80D3	-4977		Invalid export directory specified
80D4	-4978		Dropped table cannot be accessed
80D5	-4979		Unable to export the dropped table data
80D6	-1620	1	Unable to flush event monitor because it wasn't started
80D7	-1620	2	Unable to flush event monitor because it is running at a pre-version 6 output level
811E	-996		Invalid user-specified directory
812F	-290		Access not allowed: table space is quiesced
8130	-290		Access not allowed: table space is load-pending
8131	-290		Access not allowed: table space is delete-pending
8132	-290		Access not allowed: table space is backup-pending
8133	-290		Access not allowed: table space is rollforward-pending
8134	-290		Access not allowed: table space is rollforward-in-progress
8135	-290		Access not allowed: table space is recovery-pending
8136	-290		Access not allowed: table space is disabled
8137	-290		Access not allowed
8138	-291		Invalid state transition
8139	-294		Container is already being used
813A	-295		Container names are too long

Table 1. DB2 Internal Return Codes (continued)

Internal Return Code	SQL State	Reason Code	Description
813B	-297		Path name is too long
813C	-298		Bad container path
813D	-299		Duplicate container
813E	-257		Raw device is not allowed
813F	-258		Add containers pending
8146	-1442		Bad container size
8173			No dirty buffers
81A7			Invalid alternate
8201	+100		End of file reached
8203	-952		I/O Interrupt
8212	-804		Invalid request
8303	-952		Interrupt
8380			No term
8381			No interrupt
8382			No interrupt
8383			Incompatible release level
8384			Operation not supported
8385			Server/requester migration level incompatible
8386			ASP protocol error
8387			Bad ASP object OBJDSS
8388			Bad FDOCA object
8389			SNA protocol error
838A			Invalid SQL request
838B			Cursor already open
838C			Cursor not open
838D			Syntax error
838E			Invalid FDOCA descriptor
8390			End of SQLDTAGRP
8394			Parameter Error
8395			Value Error
8396			SQLDA too small
8397			Invalid RPYDSS

Table 1. DB2 Internal Return Codes (continued)

Internal Return Code	SQL State	Reason Code	Description
8398			Parser syntax error
8399			FDOCA Error
839A			End of input
839B			Code point not found
839C			SQLCODE set in parse
839D			Data descriptor mismatch
839E			Required parameter not received
839F			Bad format
83A0			Duplicate parameters detected
83A3			Translation substitution
8401	+100		End of file reached
8403	-952		Interrupt
8404	-950		Active cursor
8406	-508		Invalid cursor position
842E	-659		Check-pending state
8443	-804		General purpose validation error
844B	-1477		Table not available because forward recovery encountered no log operation
8451	-680		Too many columns
8502	-911		Deadlock encountered
8503	-1044		Interrupt
8544	-911		Lock timeout with transaction rollback
8550	-913		Lock timeout with statement rollback
856D	-902		SQLCA has been built and saved in a component-specific control block
8574	-1035		File open error
8575	-1015		Database needs recovery
8576	-1042		Deadlock start error
8577			Deadlock stop error
8578	-1034		Recovery failed
8579			Conditional conflict
85A1			Backup pending

Table 1. DB2 Internal Return Codes (continued)

Internal Return Code	SQL State	Reason Code	Description
85A2			Recovery pending
85A4	+993		Invalid new log path
85A5	+995		Invalid current log path
85A6	-1267		Exist file not found
85AA			Table space rollforward stopped
85AB			I/O error encountered
85AC			Duplicate entry
85B3			Missing log extent
85B4			Log extent is of a different size
85B5	-1472		System clock difference exceeds max_time_diff on connect (log synchronization)
85B6	-1473		System clock difference exceeds max_time_diff on commit
85B7	+1474		System clock difference exceeds max_time_diff on commit (read-only)
85B8	-276		Restore pending
85B9	-1275		Invalid stop time for rollforward
85BA	-1276		Invalid time to stop rollforward
85BB	-4970		Rollforward is missing log files
85BC	-4971		Rollforward log is already truncated
85BD	-4972		Rollforward log path is full
85BE	-4973		Rollforward log mismatch
85BF	-4974		Rollforward query error
85C0	-4975		Rollforward cancelled
85C1	-4976		Rollforward not issued on catalog
85C3	-4906		Table space rollforward has invalid table space set
85C4	-1278		Table space rollforward required for rollback
85C5	-1280		Invalid stop time for table space rollforward
8659			Term characters not provided
865A			Conversion table not loaded
865B			No conversion table exists
865C			Invalid target code page
865D			Invalid source code page

Table 1. DB2 Internal Return Codes (continued)

Internal Return Code	SQL State	Reason Code	Description
865F	-5123		Invalid code page
8660	-5124		Invalid country code
8661	-5125		Incompatible CP and CC
8662			Invalid stored procedure
8663			Invalid name
8664			Expired password
8665			Bad password
8666			User or group description over 8 characters
8667			User or group ID over 8 characters
8668			Password over 8 characters
8669			Bad group ID
866A			Bad user ID
866E			Not a descendant
866F			SMG started in background
8670			Invalid thread ID
867F	-10nn		Dropped current directory
86A8			Range too large
86A9			Invalid user name
86AA			Owner died
86AB			Path error
86AC			Programming error
86AD			Exit list full
86AE			Exit list not found
86AF			Services not initialized
870B	-138		Invalid SUBSTR parameter
8711	-910		Table or table space drop pending
8712	-804		Invalid request
8713	-804		Bad request context
8714	-804		Invalid section number
8719	-811		Non-unique answer
871D	-994		Savepoint error

Table 1. DB2 Internal Return Codes (continued)

Internal Return Code	SQL State	Reason Code	Description
871E	-996		Invalid directory, involving the path length of the REORG directory
874A	-1476		Forced rollback on unit of work because of an error on number of log tables
876B			Authorization error
876C			Authorization with different ID
876D	-902		SQLCA has been built and saved in a component-specific control block
877E			RDS error
8803	-1044		Interrupt
8905	-803		Duplicate key
8971			Index scan incomplete
897A			Index end of file
897B			Index internal error
897D			User ID suspended
8A72			Compilation error
8B07	-302		Output truncated
8B08	-413		Values out of range
8B09	-304		Conversion overflow converting date or time to character string
8B0A	-303		Incompatible types
8B0C	-309		Null input invalid
8B0D	-305		Null output invalid
8B0E	-804		Invalid output type
8B0F	-822		Invalid input address
8B10	-802		Math overflow
8B15	-311		Negative SQLDA length
8B16	-301		Incompatible types (dynamic)
8B17	-180		Time or date syntax error
8B18	-181		Time or date range error
8B1A	-404		String column overflow
8B1B	-406		Numeric column overflow
8B1C	-407		Non-nullable column

Table 1. DB2 Internal Return Codes (continued)

Internal Return Code	SQL State	Reason Code	Description
8B1F	-183		Date/time arithmetic result out of range
8B20	-182		Date/time arithmetic duration out of range
8B21	-176		Translate scalar run-time error
8B22	-436		Null term missing on input CSTR
8B47	-304		Conversion overflow (numeric value)
8B48	-302		Numeric value out of range
8B49	-801		Divide by zero operation not allowed
8B4C	-420		Invalid input format
8B4D	-410		Float string too long
8B4E			Null-only truncation
8B4F			Date is truncated
8C03	-952		Interrupt
8C05	-803		Duplicate key
9301			Allocation failure
9302			Conversion failure
9303			Transaction processor (TP) not started
9304			TP limit reached
9305			Remote TP exit normal
9306			Allocation failed, attempting retry
9307			Allocation failed, no retry attempted
9308			General allocation error
9309			Conversion failed, no retry attempted
930A			TP not started, no retry attempted
930B			Remote TP abend
930C			Communications Manager not loaded
930D			Communications Manager abend
930E			Bad security on conversation
930F			Unknown APPC error
9310			Communications Manager parameter bad
9311			Communications Manager bad request state
9312			External communications error
9313			Remote TP send error

Table 1. DB2 Internal Return Codes (continued)

Internal Return Code	SQL State	Reason Code	Description
9332			Invalid name or name number
9333			Invalid session ID
9334			Invalid command
9335			Invalid data
9336			Adapter error
9337			Reset error
9338			Other NetBIOS error
9364			Node environment is corrupted
960C			Communications Manager not loaded
960F			Generic APPC error
9617			Already connected
9618			Connection in progress
9619			Connection refused
962A			Connection timed out
962B			Address already in use
962C			No connection
962D			Socket is bound already
962E			Socket is not bound yet
962F			Socket not writable/readable yet
9630			Partial message was sent
9631			Partial message was received
9C14			FCM communication error
9C15			FCM node not found
9C16	-1229		Node recovery
A602	-901		Invalid memory address
A603	-901		General memory management error
A604	-901		Memory management error: invalid size
AB01	-901		Internal error
AC01	-901		Internal program error
C107	-986		File error
C109	-8100		Segmented tables, page number too high
C119	-995		EMP indirect not found

Table 1. DB2 Internal Return Codes (continued)

Internal Return Code	SQL State	Reason Code	Description
C11A	-995		EMP map information ended
C11B	-995		EMP map information not found
C11C	-996		Mapping information should exist, but cannot be found
C11D	-292		Cannot create file
C11E	-293		Container not accessible
C180			Agent file close error
C201	-970		Access denied
C211			Seek error
C212			Unknown media error
C213			File not found
C214			File already exists
C215			File in use
C216			Invalid file name
C217			Commit failed
C218			Undo failed
C40A	-659		Maximum object size reached
C47E			Index needs to be recreated
C47F			File renamed
C57F			File renamed
C601	-970		Access denied
C602	-972		Change disk
C603	-972		Not a DOS disk
C604	-974		Drive locked
C605	-976		Device not ready
C606	-978		File write-protected
C607	-986		File error
C608	-902		Delete directory error
C67C			Device is busy
C721	-2423		Missing index during offline backup
C90A			Maximum object size reached
CE0B	-1614	1	Encountered an unknown event monitor target type
CE0C	-1614	2	The event monitor target path was not sent

Table 1. DB2 Internal Return Codes (continued)

Internal Return Code	SQL State	Reason Code	Description
CE0D	-1614	3	Access to event monitor target path was denied
CE0E	-1614	4	Event monitor target path is not the name of a pipe
CE0F	-1614	5	No process has opened the event monitor target pipe for reading
CE10	-1614	6	Encountered an unexpected I/O error.
D085		21	Authentication failed due to no user licenses available
D107	-960		No more file tokens
D121	-289		Container full
D122			Exceeded maximum quiescers
D123	-296		Table space limit exceeded
D124	-259		Map too big
D20C	-968		Disk full
D21A	-930		No memory: UNDO heap
D315			No memory on requester
D316			No memory on server
D31D			Vectored I/O request too big
D408	-962		Maximum tables in file
D40D	-912		Too many locks
D411	-902		Maximum long field file size
D505	-930		Memory allocation failure
D509	-964		Log file full
D50A	-912		Lock list full
D57F	-1004		Disk full log file
D601	-954		No memory heap (for application)
D602	-956		No memory heap (for database)
D603	-101		No memory heap
D604	-930		Share buffer exceeded
D605	-930		Memory allocation failure
D606	-958		Too many open files
D607	-960		No more file tokens
D60B	-955		No memory sort heap
D60C	-968		Disk full

Table 1. DB2 Internal Return Codes (continued)

Internal Return Code	SQL State	Reason Code	Description
D60D	-912		Too many locks
D60E	-973		No memory
D610	-902		No memory BSU heap
D612	-953		No memory: AgentHeap
D613	-957, -959		No memory: ComHeap
D614	-961, -962		No memory: RSHeap
D615			No memory: Users/groups
D616	-930		No memory: DrIdx heap
D617	-930		No memory: ASP heap
D619	-930		No memory: Queue heap
D61A	-930		No memory: UND heap
D61B	-930		No memory: Lock heap
D61C	-930		No memory: System heap
D61D	-930		No memory
D61E	-930		No memory
D620	-930		No memory
D625	-930		No memory
D62B	-930		No memory
D67D			Shared memory set exists
D67E	-9		No memory for DosLoadMod
D905	-930		Memory allocation failure
D90F	-990		Index structure problem
DC26	-6042		No FCM MSG_ANCHOR
DC27	-6040		No FCM buffer
DC28	-6041		No FCM connection entry
DC29	-6043		No FCM request block
DC2A	-902		No high priority buffer
E101	-980	1	Bad page
E10A	-980	10	File does not exist
E10E	-982		Bad signature
E119	-1034		Page CHECKSUM error
E11A	-1035		Bad database, won't flush it

Table 1. DB2 Internal Return Codes (continued)

Internal Return Code	SQL State	Reason Code	Description
E11B	-980	27	Both primary and shadow ORFs are bad
E11C	-980	28	Primary ORF is bad
E11D	-980	29	Secondary ORF is bad
E11E	-980	30	Both primary and shadow SSFs are bad
E11F	-980	31	Primary SSF bad
E120	-980	32	Secondary SSF bad
E40B	-980	11	Object does not exist
E50D	-980	13	Bad log file
E510	-5123		Error in log control file
E511	-1258		Log control file not found
E512	-1259		I/O error accessing log control file
E513	-1260		Database not recoverable
E514	-1261		Recovery not pending
E515	-1263		Invalid log extent file
E516	-1264		Log extent file does not belong to the database
E517	-1265		Log extent file is the incorrect version
E518	-1266		Point in time prior to recovery
E521	-1034		Recovery failed
E522	-1269		Error while retrieving file during forward recovery
E57F	-1036		Adjust log file error
E602	-980	2	CRC error
E603	-980	3	Disk error
E604	-980	4	General failure
E605	-980	5	Read fault
E606	-980	6	Seek error
E607	-980	7	Sector not found
E608	-980	8	Unknown media error
E609	-980	9	Write fault
E60A	-980	10	File does not exist
E60C	-980	12	Cannot open file
E60F	-980	15	Network access denied
E623	-931		Too many open system files

Table 1. DB2 Internal Return Codes (continued)

Internal Return Code	SQL State	Reason Code	Description
F051	-1042		Invalid log record encountered during redo or undo: unknown component
F102	-902	2	BPS logic error
F103	-902	3	Invalid buffer pointer
F104	-902	4	No buffers
F109	-902	9	Data does not exist
F10A	-902	10	File already exists
F10B	-902	11	Unfixed buffer page
F10C	-902	12	Invalid file token
F10D	-902	13	Invalid file type
F110	-902	16	Invalid mode parameter
F117	-902	23	Invalid reference
F121	-902	33	RAM semaphore error
F124	-902	36	Bad configuration file
F136	-902	54	Bad header
F13B	-902	59	File not found in the reorg linked list
F149	-902	73	Bad database path
F156	-902	86	Page already exists
F176	-902	118	Invalid pool ID
F17D			State already off
F210	-902	16	Invalid mode parameter
F21B	-902	27	Bad transaction ID
F225	-902	37	Invalid file handle
F331	-902	49	Bad selector
F33C	-902	60	Tokenizer stack overflow
F345			The server STARTDBM failed
F34A	-902	74	Node manager thread failed
F34B	-902	75	Database is bad
F34C	-902	76	Parser stack overflow
F34D	-902	77	Token buffer overflow
F34E	-902	78	Bad value in code page table
F34F	-902	79	Configuration mismatch

Table 1. DB2 Internal Return Codes (continued)

Internal Return Code	SQL State	Reason Code	Description
F401	-902	1	DMS data file error
F418	-902	24	Bad table handle
F419	-902	25	Bad record ID
F42D	-902	45	Reallocation error
F42E	-902	46	Set signal error
F42F	-902	47	Invalid mode
F432	-902	50	Floating point error
F451	-1042		Invalid log record encountered during redo or undo: bad record length
F455	-902	85	Program error
F47A	-902	122	Record deleted
F47C	-902	124	Missing defaults
F51B	-902	27	Bad transaction ID
F51C	-902	28	Log file overflow
F51D	-902	29	Fatal logic error
F51E	-902	30	No active transaction
F51F	-902	31	Maximum save points
F520	-902	32	No active save point
F527	-902	39	Bad record type
F528	-902	40	Transaction ID table overflow
F529	-902	41	Invalid LSN
F52A	-902	42	Transaction already started
F550	-902	80	Database in recovery mode
F605	-1068		Message file not found
F606	-1068		Message not found
F609	-902	9	Data does not exist
F60A	-902	10	File already exists
F60B	-902	11	Unfixed buffer page
F60C	-902	12	Invalid file token
F60D	-902	13	Invalid file type
F60E	-902	14	Lock violation
F60F	-902	15	Directory overflow

Table 1. DB2 Internal Return Codes (continued)

Internal Return Code	SQL State	Reason Code	Description
F610	-902	16	Invalid mode parameter
F611	-902	17	Invalid path
F612	-902	18	Invalid page number
F613	-902	19	Sector boundary error
F614	-902	20	System internal error
F616	-902	22	File sharing error
F617	-902	23	Invalid reference
F61A	-902	26	Invalid selector
F621	-902	33	RAM semaphore error
F622	-902	34	Access error
F624	-902	36	Bad configuration file
F625	-902	37	Invalid file handle
F626	-902	38	No file descriptor
F631	-902	49	Bad selector
F635	-902	53	Conditional failure
F637	-902	55	Infinite retry
F638	-902	56	Stored procedure not found
F639	-902	57	Invalid drive
F63A	-902	58	Bad heap ID
F63D	-902	61	Duplicate queue
F63E	-902	62	Bad queue handle
F63F	-902	63	Queue message too big
F640	-902	64	No message in queue
F641	-902	65	Message not sent
F642	-902	66	Queue does not exist
F643	-902	67	Max queue limit
F644	-902	68	Invalid queue name
F690			DB2NODE environment variable has a bad value
F691			The db2nodes.cfg file contains an error
F730	-902	48	Invalid file
F733	-902	51	Invalid compile request
F85F	-902	95	Generic LOB manager error

Table 1. DB2 Internal Return Codes (continued)

Internal Return Code	SQL State	Reason Code	Description
F860	-902	96	Cannot redo operation
F861	-902	97	Beginning segment (BSEG) already trimmed
F862	-902	98	Insufficient space
F863	-902	99	No mini directory
F864	-902	100	Invalid LM descriptor
F865	-902	101	Invalid address
F866	-902	102	LF space exhausted
F867	-902	103	BSEG size/address conflict
F868	-902	104	BSEG not free or allocated
F869	-902	105	Incorrect BSEG size
F86A	-902	106	BSEG not free
F86B	-902	107	Bad count array
F86C	-902	108	Lock error
F86D	-902	109	Not found error
F86E	-902	110	Value out of bounds
F86F	-902	111	Unexpected NULL value
F870	-902	112	Encountered fatal error
F871	-902	113	Bad state
F872	-902	114	Request too big
F873			No slots for threads
F874			Thread not waiting
F875			Owner died
F87B			Too many active threads
F912	-902	18	Invalid page number
F915	-902	21	Memory allocation error
F92B	-902	43	Index token does not exist
F92C	-902	44	Key not found
F952	-902	82	SMP problems
F953	-902	83	Invalid database release
F954	-902	84	Program error
FB2E	-902	46	Set signal error
FC07	-902	7	Severe internal error

Table 1. DB2 Internal Return Codes (continued)

Internal Return Code	SQL State	Reason Code	Description
FC21	-902	33	RAM semaphore error
FC7E	-902	126	Component interface error
FC7F	-902	127	FCM programming error
FC80	-902	128	FCM daemon not available
FC81	-902	129	FCM node configuration file error
FC88	-902	136	BDS communication error
FC89	-902	137	BDS partner error
FC8F	-1445		No Context
FD8A	-902	138	Invalid partition map ID
FD8B	-902	139	Not able to fetch from catalog
FD8C	-902	140	Invalid data type
FD8D			Invalid partition number
FD8E			PMP is of a 1-node nodegroup

Appendix B. SQL Communications (SQLCA)

An SQLCA is a collection of variables that is updated at the end of the execution of every SQL statement. A program that contains executable SQL statements (except for DECLARE, INCLUDE, and WHENEVER) and is precompiled with option LANGLEVEL SAA1 (the default) or MIA must provide exactly one SQLCA, though more than one SQLCA is possible by having one SQLCA per thread in a multi-threaded application.

When a program is precompiled with option LANGLEVEL SQL92E, an SQLCODE or SQLSTATE variable may be declared in the SQL declare section or an SQLCODE variable can be declared somewhere in the program.

An SQLCA should not be provided when using LANGLEVEL SQL92E. The SQL INCLUDE statement can be used to provide the declaration of the SQLCA in all languages but REXX. The SQLCA is automatically provided in REXX.

Viewing the SQLCA Interactively

To display the SQLCA after each command you use in the command line processor, use the command **db2 -a**. The SQLCA is then provided as part of the output for subsequent commands. The SQLCA is also dumped in the db2diag.log.

SQLCA Field Descriptions

Table 2. Fields of SQLCA

Name ¹	Data Type	Field Values
sqlcaid	CHAR(8)	An "eye catcher" for storage dumps containing 'SQLCA'. The sixth byte is 'L' if line number information is returned from parsing an SQL procedure body.
sqlcabc	INTEGER	Contains the length of the SQLCA, 136.

1. The field names shown are those present in an SQLCA that is obtained via an INCLUDE statement.

Table 2. Fields of SQLCA (continued)

Name ¹	Data Type	Field Values
sqlcode	INTEGER	<p>Contains the SQL return code. For specific meanings of SQL return codes, see the message section of the <i>Message Reference</i>.</p> <p>Code Means</p> <p>0 Successful execution (although one or more SQLWARN indicators may be set).</p> <p>positive Successful execution, but with a warning condition.</p> <p>negative Error condition.</p>
sqlerrml	SMALLINT	<p>Length indicator for <i>sqlerrmc</i>, in the range 0 through 70. 0 means that the value of <i>sqlerrmc</i> is not relevant.</p>
sqlerrmc	VARCHAR (70)	<p>Contains one or more tokens, separated by X'FF', that are substituted for variables in the descriptions of error conditions.</p> <p>This field is also used when a successful connection is completed.</p> <p>When a NOT ATOMIC compound SQL statement is issued, it may contain information on up to 7 errors.</p> <p>For specific meanings of SQL return codes, see the message section of the <i>Message Reference</i>.</p>
sqlerrp	CHAR(8)	<p>Begins with a three-letter identifier indicating the product, followed by five digits indicating the version, release, and modification level of the product. For example, SQL07010 means DB2 Universal Database Version 7 Release 1 Modification level 0.</p> <p>If SQLCODE indicates an error condition, then this field identifies the module that returned the error.</p> <p>This field is also used when a successful connection is completed.</p>
sqlerrd	ARRAY	<p>Six INTEGER variables that provide diagnostic information. These values are generally empty if there are no errors, except for sqlerrd(6) from a partitioned database.</p>

Table 2. Fields of SQLCA (continued)

Name ¹	Data Type	Field Values
sqlerrd(1)	INTEGER	<p>If connection is invoked and successful, contains the maximum expected difference in length of mixed character data (CHAR data types) when converted to the database code page from the application code page. A value of 0 or 1 indicates no expansion; a value greater than 1 indicates a possible expansion in length; a negative value indicates a possible contraction. ^a</p> <p>On successful return from an SQL procedure, contains the return status value from the SQL procedure.</p>
sqlerrd(2)	INTEGER	<p>If connection is invoked and successful, contains the maximum expected difference in length of mixed character data (CHAR data types) when converted to the application code page from the database code page. A value of 0 or 1 indicates no expansion; a value greater than 1 indicates a possible expansion in length; a negative value indicates a possible contraction. ^a If the SQLCA results from a NOT ATOMIC compound SQL statement that encountered one or more errors, the value is set to the number of statements that failed.</p>
sqlerrd(3)	INTEGER	<p>If PREPARE is invoked and successful, contains an estimate of the number of rows that will be returned. After INSERT, UPDATE, and DELETE, contains the actual number of rows affected. If compound SQL is invoked, contains an accumulation of all sub-statement rows. If CONNECT is invoked, contains 1 if the database can be updated; 2 if the database is read only.</p> <p>If CREATE PROCEDURE for an SQL procedure is invoked and an error is encountered parsing the SQL procedure body, contains the line number where the error was encountered. The sixth byte of sqlcaid must be 'L' for this to be a valid line number.</p>
sqlerrd(4)	INTEGER	<p>If PREPARE is invoked and successful, contains a relative cost estimate of the resources required to process the statement. If compound SQL is invoked, contains a count of the number of successful sub-statements. If CONNECT is invoked, contains 0 for a one-phase commit from a down-level client; 1 for a one-phase commit; 2 for a one-phase, read-only commit; and 3 for a two-phase commit.</p>

Table 2. Fields of SQLCA (continued)

Name ¹	Data Type	Field Values
sqlerrd(5)	INTEGER	<p>Contains the total number of rows deleted, inserted, or updated as a result of both:</p> <ul style="list-style-type: none"> • The enforcement of constraints after a successful delete operation • The processing of triggered SQL statements from activated triggers. <p>If compound SQL is invoked, contains an accumulation of the number of such rows for all substatements. In some cases when an error is encountered, this field contains a negative value that is an internal error pointer. If CONNECT is invoked, contains an authentication type value of 0 for a server authentication; 1 for client authentication; 2 for authentication using DB2 Connect; 3 for DCE security services authentication; 255 for unspecified authentication.</p>
sqlerrd(6)	INTEGER	<p>For a partitioned database, contains the partition number of the partition that encountered the error or warning. If no errors or warnings were encountered, this field contains the partition number of the coordinator node. The number in this field is the same as that specified for the partition in the db2nodes.cfg file.</p>
sqlwarn	Array	<p>A set of warning indicators, each containing a blank or W. If compound SQL is invoked, contains an accumulation of the warning indicators set for all substatements.</p>
sqlwarn0	CHAR(1)	<p>Blank if all other indicators are blank; contains W if at least one other indicator is not blank.</p>
sqlwarn1	CHAR(1)	<p>Contains W if the value of a string column was truncated when assigned to a host variable. Contains N if the null terminator was truncated.</p> <p>Contains A if the CONNECT or ATTACH is successful and the authID for the connection is longer than 8 bytes.</p>
sqlwarn2	CHAR(1)	<p>Contains W if null values were eliminated from the argument of a function. ^b</p>
sqlwarn3	CHAR(1)	<p>Contains W if the number of columns is not equal to the number of host variables.</p>
sqlwarn4	CHAR(1)	<p>Contains W if a prepared UPDATE or DELETE statement does not include a WHERE clause.</p>
sqlwarn5	CHAR(1)	<p>Reserved for future use.</p>
sqlwarn6	CHAR(1)	<p>Contains W if the result of a date calculation was adjusted to avoid an impossible date.</p>

Table 2. Fields of SQLCA (continued)

Name ¹	Data Type	Field Values
sqlwarn7	CHAR(1)	Reserved for future use. If CONNECT is invoked and successful, contains 'E' if the DYN_QUERY_MGMT database configuration parameter is enabled.
sqlwarn8	CHAR(1)	Contains W if a character that could not be converted was replaced with a substitution character.
sqlwarn9	CHAR(1)	Contains W if arithmetic expressions with errors were ignored during column function processing.
sqlwarn10	CHAR(1)	Contains W if there was a conversion error when converting a character data value in one of the fields in the SQLCA.
sqlstate	CHAR(5)	A return code that indicates the outcome of the most recently executed SQL statement.

Note:

- a See the "Character Conversion Expansion Factor" section of the "Programming in Complex Environments" chapter in the *Application Development Guide* for details.
- b Some functions may not set SQLWARN2 to W even though null values were eliminated because the result was not dependent on the elimination of null values.

Order of Error Reporting

The order of error reporting is as follows:

1. Severe error conditions are always reported. When a severe error is reported, there are no additions to the SQLCA.
2. If no severe error occurs, a deadlock error takes precedence over other errors.
3. For all other errors, the SQLCA for the first negative SQL code is returned.
4. If no negative SQL codes are detected, the SQLCA for the first warning (that is, positive SQL code) is returned.

For DB2 Enterprise - Extended Edition, the exception to this rule occurs if a data manipulation operation is issued on a table that is empty on one partition, but has data on other nodes. The SQLCODE +100 is only returned to the application if agents from all partitions return SQL0100W, either because the table is empty on all partitions or there are no more rows that satisfy the WHERE clause in an UPDATE statement.

DB2 Enterprise - Extended Edition Usage of the SQLCA

In DB2 Universal Database Enterprise - Extended Edition, one SQL statement may be executed by a number of agents on different partitions, and each agent may return a different SQLCA for different errors or warnings. The coordinator agent also has its own SQLCA.

To provide a consistent view for applications, all SQLCA values are merged into one structure and SQLCA fields indicate global counts. For example:

- For all errors and warnings, the *sqlwarn* field contains the warning flags received from all agents.
- Values in the *sqlerrd* fields indicating row counts are accumulations from all agents.

Note that SQLSTATE 09000 may not be returned in all cases of an error occurring while processing a triggered SQL statement.

Appendix C. Using the DB2 Library

The DB2 Universal Database library consists of online help, books (PDF and HTML), and sample programs in HTML format. This section describes the information that is provided, and how you can access it.

To access product information online, you can use the Information Center. For more information, see “Accessing Information with the Information Center” on page 291. You can view task information, DB2 books, troubleshooting information, sample programs, and DB2 information on the Web.

DB2 PDF Files and Printed Books

DB2 Information

The following table divides the DB2 books into four categories:

DB2 Guide and Reference Information

These books contain the common DB2 information for all platforms.

DB2 Installation and Configuration Information

These books are for DB2 on a specific platform. For example, there are separate *Quick Beginnings* books for DB2 on OS/2, Windows, and UNIX-based platforms.

Cross-platform sample programs in HTML

These samples are the HTML version of the sample programs that are installed with the Application Development Client. The samples are for informational purposes and do not replace the actual programs.

Release notes

These files contain late-breaking information that could not be included in the DB2 books.

The installation manuals, release notes, and tutorials are viewable in HTML directly from the product CD-ROM. Most books are available in HTML on the product CD-ROM for viewing and in Adobe Acrobat (PDF) format on the DB2 publications CD-ROM for viewing and printing. You can also order a printed copy from IBM; see “Ordering the Printed Books” on page 287. The following table lists books that can be ordered.

On OS/2 and Windows platforms, you can install the HTML files under the `sql1lib\doc\html` directory. DB2 information is translated into different

languages; however, all the information is not translated into every language. Whenever information is not available in a specific language, the English information is provided

On UNIX platforms, you can install multiple language versions of the HTML files under the `doc/%L/html` directories, where `%L` represents the locale. For more information, refer to the appropriate *Quick Beginnings* book.

You can obtain DB2 books and access information in a variety of ways:

- “Viewing Information Online” on page 290
- “Searching Information Online” on page 294
- “Ordering the Printed Books” on page 287
- “Printing the PDF Books” on page 286

Table 3. DB2 Information

Name	Description	Form Number PDF File Name	HTML Directory
DB2 Guide and Reference Information			
<i>Administration Guide</i>	<i>Administration Guide: Planning</i> provides an overview of database concepts, information about design issues (such as logical and physical database design), and a discussion of high availability.	SC09-2946 db2d1x70	db2d0
	<i>Administration Guide: Implementation</i> provides information on implementation issues such as implementing your design, accessing databases, auditing, backup and recovery.	SC09-2944 db2d2x70	
	<i>Administration Guide: Performance</i> provides information on database environment and application performance evaluation and tuning.	SC09-2945 db2d3x70	
	You can order the three volumes of the <i>Administration Guide</i> in the English language in North America using the form number SBOF-8934.		
<i>Administrative API Reference</i>	Describes the DB2 application programming interfaces (APIs) and data structures that you can use to manage your databases. This book also explains how to call APIs from your applications.	SC09-2947 db2b0x70	db2b0

Table 3. DB2 Information (continued)

Name	Description	Form Number PDF File Name	HTML Directory
<i>Application Building Guide</i>	Provides environment setup information and step-by-step instructions about how to compile, link, and run DB2 applications on Windows, OS/2, and UNIX-based platforms.	SC09-2948 db2axx70	db2ax
<i>APPC, CPI-C, and SNA Sense Codes</i>	Provides general information about APPC, CPI-C, and SNA sense codes that you may encounter when using DB2 Universal Database products.	No form number db2apx70	db2ap
	Available in HTML format only.		
<i>Application Development Guide</i>	Explains how to develop applications that access DB2 databases using embedded SQL or Java (JDBC and SQLJ). Discussion topics include writing stored procedures, writing user-defined functions, creating user-defined types, using triggers, and developing applications in partitioned environments or with federated systems.	SC09-2949 db2a0x70	db2a0
<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.	SC09-2950 db2l0x70	db2l0
<i>Command Reference</i>	Explains how to use the Command Line Processor and describes the DB2 commands that you can use to manage your database.	SC09-2951 db2n0x70	db2n0
<i>Connectivity Supplement</i>	Provides setup and reference information on how 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. This book also details how to use DRDA application servers with DB2 Connect application requesters.	No form number db2h1x70	db2h1
	Available in HTML and PDF only.		

Table 3. DB2 Information (continued)

Name	Description	Form Number PDF File Name	HTML Directory
<i>Data Movement Utilities Guide and Reference</i>	Explains how to use DB2 utilities, such as import, export, load, AutoLoader, and DPROP, that facilitate the movement of data.	SC09-2955 db2dmx70	db2dm
<i>Data Warehouse Center Administration Guide</i>	Provides information on how to build and maintain a data warehouse using the Data Warehouse Center.	SC26-9993 db2ddx70	db2dd
<i>Data Warehouse Center Application Integration Guide</i>	Provides information to help programmers integrate applications with the Data Warehouse Center and with the Information Catalog Manager.	SC26-9994 db2adx70	db2ad
<i>DB2 Connect User's Guide</i>	Provides concepts, programming, and general usage information for the DB2 Connect products.	SC09-2954 db2c0x70	db2c0
<i>DB2 Query Patroller Administration Guide</i>	Provides an operational overview of the DB2 Query Patroller system, specific operational and administrative information, and task information for the administrative graphical user interface utilities.	SC09-2958 db2dwx70	db2dw
<i>DB2 Query Patroller User's Guide</i>	Describes how to use the tools and functions of the DB2 Query Patroller.	SC09-2960 db2wwx70	db2ww
<i>Glossary</i>	Provides definitions for terms used in DB2 and its components. Available in HTML format and in the <i>SQL Reference</i> .	No form number db2t0x70	db2t0
<i>Image, Audio, and Video Extenders Administration and Programming</i>	Provides general information about DB2 extenders, and information on the administration and configuration of the image, audio, and video (IAV) extenders and on programming using the IAV extenders. It includes reference information, diagnostic information (with messages), and samples.	SC26-9929 dmbu7x70	dmbu7
<i>Information Catalog Manager Administration Guide</i>	Provides guidance on managing information catalogs.	SC26-9995 db2dix70	db2di

Table 3. DB2 Information (continued)

Name	Description	Form Number PDF File Name	HTML Directory
<i>Information Catalog Manager Programming Guide and Reference</i>	Provides definitions for the architected interfaces for the Information Catalog Manager.	SC26-9997 db2bix70	db2bi
<i>Information Catalog Manager User's Guide</i>	Provides information on using the Information Catalog Manager user interface.	SC26-9996 db2aix70	db2ai
<i>Installation and Configuration Supplement</i>	Guides you through the planning, installation, and setup of platform-specific DB2 clients. This supplement also contains information on binding, setting up client and server communications, DB2 GUI tools, DRDA AS, distributed installation, the configuration of distributed requests, and accessing heterogeneous data sources.	GC09-2957 db2iyx70	db2iy
<i>Message Reference</i>	Lists messages and codes issued by DB2, the Information Catalog Manager, and the Data Warehouse Center, and describes the actions you should take. You can order both volumes of the Message Reference in the English language in North America with the form number SBOF-8932.	Volume 1 GC09-2978 db2m1x70 Volume 2 GC09-2979 db2m2x70	db2m0
<i>OLAP Integration Server Administration Guide</i>	Explains how to use the Administration Manager component of the OLAP Integration Server.	SC27-0787 db2dpx70	n/a
<i>OLAP Integration Server Metaoutline User's Guide</i>	Explains how to create and populate OLAP metaoutlines using the standard OLAP Metaoutline interface (not by using the Metaoutline Assistant).	SC27-0784 db2upx70	n/a
<i>OLAP Integration Server Model User's Guide</i>	Explains how to create OLAP models using the standard OLAP Model Interface (not by using the Model Assistant).	SC27-0783 db2lpx70	n/a
<i>OLAP Setup and User's Guide</i>	Provides configuration and setup information for the OLAP Starter Kit.	SC27-0702 db2ipx70	db2ip
<i>OLAP Spreadsheet Add-in User's Guide for Excel</i>	Describes how to use the Excel spreadsheet program to analyze OLAP data.	SC27-0786 db2epx70	db2ep

Table 3. DB2 Information (continued)

Name	Description	Form Number PDF File Name	HTML Directory
<i>OLAP Spreadsheet Add-in User's Guide for Lotus 1-2-3</i>	Describes how to use the Lotus 1-2-3 spreadsheet program to analyze OLAP data.	SC27-0785 db2tpx70	db2tp
<i>Replication Guide and Reference</i>	Provides planning, configuration, administration, and usage information for the IBM Replication tools supplied with DB2.	SC26-9920 db2e0x70	db2e0
<i>Spatial Extender User's Guide and Reference</i>	Provides information about installing, configuring, administering, programming, and troubleshooting the Spatial Extender. Also provides significant descriptions of spatial data concepts and provides reference information (messages and SQL) specific to the Spatial Extender.	SC27-0701 db2sbx70	db2sb
<i>SQL Getting Started</i>	Introduces SQL concepts and provides examples for many constructs and tasks.	SC09-2973 db2y0x70	db2y0
<i>SQL Reference, Volume 1 and Volume 2</i>	Describes SQL syntax, semantics, and the rules of the language. This book also includes information about release-to-release incompatibilities, product limits, and catalog views. You can order both volumes of the <i>SQL Reference</i> in the English language in North America with the form number SBOF-8933.	Volume 1 SC09-2974 db2s1x70 Volume 2 SC09-2975 db2s2x70	db2s0
<i>System Monitor Guide and Reference</i>	Describes how to collect different kinds of information about databases and the database manager. This book explains how to use the information to understand database activity, improve performance, and determine the cause of problems.	SC09-2956 db2f0x70	db2f0
<i>Text Extender Administration and Programming</i>	Provides general information about DB2 extenders and information on the administration and configuring of the text extender and on programming using the text extenders. It includes reference information, diagnostic information (with messages) and samples.	SC26-9930 desu9x70	desu9

Table 3. DB2 Information (continued)

Name	Description	Form Number PDF File Name	HTML Directory
<i>Troubleshooting Guide</i>	Helps you determine the source of errors, recover from problems, and use diagnostic tools in consultation with DB2 Customer Service.	GC09-2850 db2p0x70	db2p0
<i>What's New</i>	Describes the new features, functions, and enhancements in DB2 Universal Database, Version 7.	SC09-2976 db2q0x70	db2q0
DB2 Installation and Configuration Information			
<i>DB2 Connect Enterprise Edition for OS/2 and Windows Quick Beginnings</i>	Provides planning, migration, installation, and configuration information for DB2 Connect Enterprise Edition on the OS/2 and Windows 32-bit operating systems. This book also contains installation and setup information for many supported clients.	GC09-2953 db2c6x70	db2c6
<i>DB2 Connect Enterprise Edition for UNIX Quick Beginnings</i>	Provides planning, migration, installation, configuration, and task information for DB2 Connect Enterprise Edition on UNIX-based platforms. This book also contains installation and setup information for many supported clients.	GC09-2952 db2cyx70	db2cy
<i>DB2 Connect Personal Edition Quick Beginnings</i>	Provides planning, migration, installation, configuration, and task information for DB2 Connect Personal Edition on the OS/2 and Windows 32-bit operating systems. This book also contains installation and setup information for all supported clients.	GC09-2967 db2c1x70	db2c1
<i>DB2 Connect Personal Edition Quick Beginnings for Linux</i>	Provides planning, installation, migration, and configuration information for DB2 Connect Personal Edition on all supported Linux distributions.	GC09-2962 db2c4x70	db2c4
<i>DB2 Data Links Manager Quick Beginnings</i>	Provides planning, installation, configuration, and task information for DB2 Data Links Manager for AIX and Windows 32-bit operating systems.	GC09-2966 db2z6x70	db2z6

Table 3. DB2 Information (continued)

Name	Description	Form Number PDF File Name	HTML Directory
<i>DB2 Enterprise - Extended Edition for UNIX Quick Beginnings</i>	Provides planning, installation, and configuration information for DB2 Enterprise - Extended Edition on UNIX-based platforms. This book also contains installation and setup information for many supported clients.	GC09-2964 db2v3x70	db2v3
<i>DB2 Enterprise - Extended Edition for Windows Quick Beginnings</i>	Provides planning, installation, and configuration information for DB2 Enterprise - Extended Edition for Windows 32-bit operating systems. This book also contains installation and setup information for many supported clients.	GC09-2963 db2v6x70	db2v6
<i>DB2 for OS/2 Quick Beginnings</i>	Provides planning, installation, migration, and configuration information for DB2 Universal Database on the OS/2 operating system. This book also contains installation and setup information for many supported clients.	GC09-2968 db2i2x70	db2i2
<i>DB2 for UNIX Quick Beginnings</i>	Provides planning, installation, migration, and configuration information for DB2 Universal Database on UNIX-based platforms. This book also contains installation and setup information for many supported clients.	GC09-2970 db2ixx70	db2ix
<i>DB2 for Windows Quick Beginnings</i>	Provides planning, installation, migration, and configuration information for DB2 Universal Database on Windows 32-bit operating systems. This book also contains installation and setup information for many supported clients.	GC09-2971 db2i6x70	db2i6
<i>DB2 Personal Edition Quick Beginnings</i>	Provides planning, installation, migration, and configuration information for DB2 Universal Database Personal Edition on the OS/2 and Windows 32-bit operating systems.	GC09-2969 db2i1x70	db2i1
<i>DB2 Personal Edition Quick Beginnings for Linux</i>	Provides planning, installation, migration, and configuration information for DB2 Universal Database Personal Edition on all supported Linux distributions.	GC09-2972 db2i4x70	db2i4

Table 3. DB2 Information (continued)

Name	Description	Form Number PDF File Name	HTML Directory
<i>DB2 Query Patroller Installation Guide</i>	Provides installation information about DB2 Query Patroller.	GC09-2959 db2iwx70	db2iw
<i>DB2 Warehouse Manager Installation Guide</i>	Provides installation information for warehouse agents, warehouse transformers, and the Information Catalog Manager.	GC26-9998 db2idx70	db2id
Cross-Platform Sample Programs in HTML			
Sample programs in HTML	Provides the sample programs in HTML format for the programming languages on all platforms supported by DB2. The sample programs are provided for informational purposes only. Not all samples are available in all programming languages. The HTML samples are only available when the DB2 Application Development Client is installed. For more information on the programs, refer to the <i>Application Building Guide</i> .	No form number	db2hs
Release Notes			
<i>DB2 Connect Release Notes</i>	Provides late-breaking information that could not be included in the DB2 Connect books.	See note #2.	db2cr
<i>DB2 Installation Notes</i>	Provides late-breaking installation-specific information that could not be included in the DB2 books.	Available on product CD-ROM only.	
<i>DB2 Release Notes</i>	Provides late-breaking information about all DB2 products and features that could not be included in the DB2 books.	See note #2.	db2ir

Notes:

1. The character *x* in the sixth position of the file name indicates the language version of a book. For example, the file name db2d0e70 identifies the English version of the *Administration Guide* and the file name db2d0f70 identifies the French version of the same book. The following letters are used in the sixth position of the file name to indicate the language version:

Language	Identifier
Brazilian Portuguese	b

Bulgarian	u
Czech	x
Danish	d
Dutch	q
English	e
Finnish	y
French	f
German	g
Greek	a
Hungarian	h
Italian	i
Japanese	j
Korean	k
Norwegian	n
Polish	p
Portuguese	v
Russian	r
Simp. Chinese	c
Slovenian	l
Spanish	z
Swedish	s
Trad. Chinese	t
Turkish	m

2. Late breaking information that could not be included in the DB2 books is available in the Release Notes in HTML format and as an ASCII file. The HTML version is available from the Information Center and on the product CD-ROMs. To view the ASCII file:
 - On UNIX-based platforms, see the `Release.Notes` file. This file is located in the `DB2DIR/Readme/%L` directory, where `%L` represents the locale name and `DB2DIR` represents:
 - `/usr/lpp/db2_07_01` on AIX
 - `/opt/IBMDB2/V7.1` on HP-UX, PTX, Solaris, and Silicon Graphics IRIX
 - `/usr/IBMDB2/V7.1` on Linux.
 - On other platforms, see the `RELEASE.TXT` file. This file is located in the directory where the product is installed. On OS/2 platforms, you can also double-click the **IBM DB2** folder and then double-click the **Release Notes** icon.

Printing the PDF Books

If you prefer to have printed copies of the books, you can print the PDF files found on the DB2 publications CD-ROM. Using the Adobe Acrobat Reader, you can print either the entire book or a specific range of pages. For the file name of each book in the library, see Table 3 on page 278.

You can obtain the latest version of the Adobe Acrobat Reader from the Adobe Web site at <http://www.adobe.com>.

The PDF files are included on the DB2 publications CD-ROM with a file extension of PDF. To access the PDF files:

1. Insert the DB2 publications CD-ROM. On UNIX-based platforms, mount the DB2 publications CD-ROM. Refer to your *Quick Beginnings* book for the mounting procedures.
2. Start the Acrobat Reader.
3. Open the desired PDF file from one of the following locations:
 - On OS/2 and Windows platforms:
x:\doc\language directory, where *x* represents the CD-ROM drive and *language* represent the two-character country code that represents your language (for example, EN for English).
 - On UNIX-based platforms:
/cdrom/doc/%L directory on the CD-ROM, where */cdrom* represents the mount point of the CD-ROM and *%L* represents the name of the desired locale.

You can also copy the PDF files from the CD-ROM to a local or network drive and read them from there.

Ordering the Printed Books

You can order the printed DB2 books either individually or as a set (in North America only) by using a sold bill of forms (SBOF) number. To order books, contact your IBM authorized dealer or marketing representative, or phone 1-800-879-2755 in the United States or 1-800-IBM-4YOU in Canada. You can also order the books from the Publications Web page at <http://www.elink.ibm.com/pbl/pbl>.

Two sets of books are available. SBOF-8935 provides reference and usage information for the DB2 Warehouse Manager. SBOF-8931 provides reference and usage information for all other DB2 Universal Database products and features. The contents of each SBOF are listed in the following table:

Table 4. Ordering the printed books

SBOF Number	Books Included	
SBOF-8931	<ul style="list-style-type: none"> • Administration Guide: Planning • Administration Guide: Implementation • Administration Guide: Performance • Administrative API Reference • Application Building Guide • Application Development Guide • CLI Guide and Reference • Command Reference • Data Movement Utilities Guide and Reference • Data Warehouse Center Administration Guide • Data Warehouse Center Application Integration Guide • DB2 Connect User's Guide • Installation and Configuration Supplement • Image, Audio, and Video Extenders Administration and Programming • Message Reference, Volumes 1 and 2 	<ul style="list-style-type: none"> • OLAP Integration Server Administration Guide • OLAP Integration Server Metaoutline User's Guide • OLAP Integration Server Model User's Guide • OLAP Integration Server User's Guide • OLAP Setup and User's Guide • OLAP Spreadsheet Add-in User's Guide for Excel • OLAP Spreadsheet Add-in User's Guide for Lotus 1-2-3 • Replication Guide and Reference • Spatial Extender Administration and Programming Guide • SQL Getting Started • SQL Reference, Volumes 1 and 2 • System Monitor Guide and Reference • Text Extender Administration and Programming • Troubleshooting Guide • What's New
SBOF-8935	<ul style="list-style-type: none"> • Information Catalog Manager Administration Guide • Information Catalog Manager User's Guide • Information Catalog Manager Programming Guide and Reference 	<ul style="list-style-type: none"> • Query Patroller Administration Guide • Query Patroller User's Guide

DB2 Online Documentation

Accessing Online Help

Online help is available with all DB2 components. The following table describes the various types of help.

Type of Help	Contents	How to Access...
<i>Command Help</i>	Explains the syntax of commands in the command line processor.	<p>From the command line processor in interactive mode, enter:</p> <p style="padding-left: 40px;"><i>? command</i></p> <p>where <i>command</i> represents a keyword or the entire command.</p> <p>For example, <i>? catalog</i> displays help for all the CATALOG commands, while <i>? catalog database</i> displays help for the CATALOG DATABASE command.</p>
<i>Client Configuration Assistant Help</i>	Explains the tasks you can perform in a window or notebook. The help includes overview and prerequisite information you need to know, and it describes how to use the window or notebook controls.	From a window or notebook, click the Help push button or press the F1 key.
<i>Command Center Help</i>		
<i>Control Center Help</i>		
<i>Data Warehouse Center Help</i>		
<i>Event Analyzer Help</i>		
<i>Information Catalog Manager Help</i>		
<i>Satellite Administration Center Help</i>		
<i>Script Center Help</i>	<p>From the command line processor in interactive mode, enter:</p> <p style="padding-left: 40px;"><i>? XXXnnnnn</i></p> <p>where <i>XXXnnnnn</i> represents a valid message identifier.</p> <p>For example, <i>? SQL30081</i> displays help about the SQL30081 message.</p> <p>To view message help one screen at a time, enter:</p> <p style="padding-left: 40px;"><i>? XXXnnnnn more</i></p> <p>To save message help in a file, enter:</p> <p style="padding-left: 40px;"><i>? XXXnnnnn > filename.ext</i></p> <p>where <i>filename.ext</i> represents the file where you want to save the message help.</p>	

Type of Help	Contents	How to Access...
<i>SQL Help</i>	Explains the syntax of SQL statements.	<p>From the command line processor in interactive mode, enter:</p> <pre>help <i>statement</i></pre> <p>where <i>statement</i> represents an SQL statement.</p> <p>For example, help SELECT displays help about the SELECT statement.</p> <p>Note: SQL help is not available on UNIX-based platforms.</p>
<i>SQLSTATE Help</i>	Explains SQL states and class codes.	<p>From the command line processor in interactive mode, enter:</p> <pre>? <i>sqlstate</i> or ? <i>class code</i></pre> <p>where <i>sqlstate</i> represents a valid five-digit SQL state and <i>class code</i> represents the 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>

Viewing Information Online

The books 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 view the online books or sample programs with any browser that conforms to HTML Version 3.2 specifications.

To view online books or sample programs:

- If you are running DB2 administration tools, use the Information Center.
- From a browser, click **File** —> **Open Page**. The page you open contains descriptions of and links to DB2 information:
 - On UNIX-based platforms, open the following page:

```
INSTHOME/sql1lib/doc/%L/html/index.htm
```

where %L represents the locale name.

- On other platforms, open the following page:

```
sql1lib\doc\html\index.htm
```

The path is located on the drive where DB2 is installed.

If you have not installed the Information Center, you can open the page by double-clicking the **DB2 Information** icon. Depending on the system you are using, the icon is in the main product folder or the Windows Start menu.

Installing the Netscape Browser

If you do not already have a Web browser installed, you can install Netscape from the Netscape CD-ROM found in the product boxes. For detailed instructions on how to install it, perform the following:

1. Insert the Netscape CD-ROM.
2. On UNIX-based platforms only, mount the CD-ROM. Refer to your *Quick Beginnings* book for the mounting procedures.
3. For installation instructions, refer to the `CDNAVnn.txt` file, where *nn* represents your two character language identifier. The file is located at the root directory of the CD-ROM.

Accessing Information with the Information Center

The Information Center provides quick access to DB2 product information. The Information Center is available on all platforms on which the DB2 administration tools are available.

You can open the Information Center by double-clicking the Information Center icon. Depending on the system you are using, the icon is in the Information folder in the main product folder or the Windows **Start** menu.

You can also access the Information Center by using the toolbar and the **Help** menu on the DB2 Windows platform.

The Information Center provides six types of information. Click the appropriate tab to look at the topics provided for that type.

Tasks	Key tasks you can perform using DB2.
Reference	DB2 reference information, such as keywords, commands, and APIs.
Books	DB2 books.
Troubleshooting	Categories of error messages and their recovery actions.
Sample Programs	Sample programs that come with the DB2 Application Development Client. If you did not install the DB2 Application Development Client, this tab is not displayed.
Web	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 a find feature, so you can look for a specific topic without browsing the lists.

For a full text search, follow the hypertext link in the Information Center to the **Search DB2 Online Information** search form.

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 using one of the following methods:

On Windows

Click **Start** and select **Programs** → **IBM DB2** → **Information** → **Start HTML Search Server**.

On OS/2

Double-click the **DB2 for OS/2** folder, and then double-click the **Start HTML Search Server** icon.

Refer to the release notes if you experience any other problems when searching the HTML information.

Note: The Search function is not available in the Linux, PTX, and Silicon Graphics IRIX environments.

Using DB2 Wizards

Wizards help you complete specific administration tasks by taking you through each task one step at a time. Wizards are available through the Control Center and the Client Configuration Assistant. The following table lists the wizards and describes their purpose.

Note: The Create Database, Create Index, Configure Multisite Update, and Performance Configuration wizards are available for the partitioned database environment.

Wizard	Helps You to...	How to Access...
<i>Add Database</i>	Catalog a database on a client workstation.	From the Client Configuration Assistant, click Add .
<i>Backup Database</i>	Determine, create, and schedule a backup plan.	From the Control Center, right-click the database you want to back up and select Backup → Database Using Wizard .

Wizard	Helps You to...	How to Access...
<i>Configure Multisite Update</i>	Configure a multisite update, a distributed transaction, or a two-phase commit.	From the Control Center, right-click the Databases folder and select Multisite Update .
<i>Create Database</i>	Create a database, and perform some basic configuration tasks.	From the Control Center, right-click the Databases folder and select Create → Database Using Wizard .
<i>Create Table</i>	Select basic data types, and create a primary key for the table.	From the Control Center, right-click the Tables icon and select Create → Table Using Wizard .
<i>Create Table Space</i>	Create a new table space.	From the Control Center, right-click the Table Spaces icon and select Create → Table Space Using Wizard .
<i>Create Index</i>	Advise which indexes to create and drop for all your queries.	From the Control Center, right-click the Index icon and select Create → Index Using Wizard .
<i>Performance Configuration</i>	Tune the performance of a database by updating configuration parameters to match your business requirements.	From the Control Center, right-click the database you want to tune and select Configure Performance Using Wizard . For the partitioned database environment, from the Database Partitions view, right-click the first database partition you want to tune and select Configure Performance Using Wizard .
<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, right-click the database you want to restore and select Restore → Database Using Wizard .

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, perform the following steps:

1. Copy all files and subdirectories from `\sql11ib\doc\html` on your local system to a Web server. Each book has its own subdirectory that contains 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, refer to the NetQuestion Appendix in the *Installation and Configuration Supplement*.
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. When you are able to view the book files, you can bookmark commonly viewed topics. You will probably want to bookmark the following pages:
 - List of books
 - Tables of contents of frequently used books
 - Frequently referenced articles, such as the ALTER TABLE topic
 - The Search form

For information about how you can serve the DB2 Universal Database online documentation files from a central machine, refer to the NetQuestion Appendix in the *Installation and Configuration Supplement*.

Searching Information Online

To find information in the HTML files, use one of the following methods:

- Click **Search** in the top frame. Use the search form to find a specific topic. This function is not available in the Linux, PTX, or Silicon Graphics IRIX environments.
- Click **Index** in the top frame. Use the index to find a specific topic in the book.
- Display the table of contents or index of the help or 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 “Accessing Information with the Information Center” on page 291 for details.

Appendix D. Notices

IBM may not offer the products, services, or features discussed in this document in all countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described 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:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:
INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make

improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

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
Office of the Lab Director
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.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information 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.

COPYRIGHT LICENSE:

This information may contain sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Each copy or any portion of these sample programs or any derivative work must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

Trademarks

The following terms, which may be denoted by an asterisk(*), are trademarks of International Business Machines Corporation in the United States, other countries, or both.

ACF/VTAM	IBM
AISPO	IMS
AIX	IMS/ESA
AIX/6000	LAN DistanceMVS
AIXwindows	MVS/ESA
AnyNet	MVS/XA
APPN	Net.Data
AS/400	OS/2
BookManager	OS/390
CICS	OS/400
C Set++	PowerPC
C/370	QBIC
DATABASE 2	QMF
DataHub	RACF
DataJoiner	RISC System/6000
DataPropagator	RS/6000
DataRefresher	S/370
DB2	SP
DB2 Connect	SQL/DS
DB2 Extenders	SQL/400
DB2 OLAP Server	System/370
DB2 Universal Database	System/390
Distributed Relational Database Architecture	SystemView
DRDA	VisualAge
eNetwork	VM/ESA
Extended Services	VSE/ESA
FFST	VTAM
First Failure Support Technology	WebExplorer
	WIN-OS/2

The following terms are trademarks or registered trademarks of other companies:

Microsoft, Windows, and Windows NT are trademarks or registered trademarks of Microsoft Corporation.

Java or all Java-based trademarks and logos, and Solaris are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Tivoli and NetView are trademarks of Tivoli Systems Inc. in the United States, other countries, or both.

UNIX is a registered trademark in the United States, other countries or both and is licensed exclusively through X/Open Company Limited.

Other company, product, or service names, which may be denoted by a double asterisk(**) may be trademarks or service marks of others.

Index

A

abends, responding to 6
access plan, viewing with Control Center 14
active logs 242
adapter number (NetBIOS) 56
add database wizard 292, 293
adm subdirectory, tools 15
administration of databases 13
administrative tools (DB2)
 description 14
 logging 171
 tracing 183
administrative tools (Windows NT) 230
administrator authority 27
agent 153
 hangs 127
agent pool 153
AIX commands 213
alert
 overview 173
 SNA 186
 SNA, routing (OS/2) 224
 SNMP 173
alias 43
APPC protocol 59
application errors and warnings 78
application requesters (AR) 61
application servers (AS) 61
applications 99
 CLI, ODBC and JDBC 75
 compiling 73
 db2ocat tool 80
 determining status 7
 for DB2 Universal Database Enterprise - Extended Edition for the Solaris Operating Environment and DB2 Universal Database Enterprise - Extended Edition for Windows NT 98
 linking 73
 running 74
 slow or hanging because of locking 37
 SQLJ 79
 stored procedure 82

 applications 99 (*continued*)
 suspended (DB2 Universal Database Enterprise - Extended Edition for the Solaris Operating Environment and DB2 Universal Database Enterprise - Extended Edition for Windows NT) 96
 suspended (TCP/IP) 51
 suspended or looping 74
 troubleshooting 71
 architectural integrity, checking 16
 archive logging 242
 archived logs
 offline 242
 online 242
 AS/400, connections 61
 authentication
 host connections 66
 AutoLoader (DB2 Universal Database Enterprise - Extended Edition for the Solaris Operating Environment and DB2 Universal Database Enterprise - Extended Edition for Windows NT) 93

B

backup
 plan 14
 troubleshooting 28
 using Control Center 14
backup database wizard 292
bad state 103
bin subdirectory, tools 15
binary data, loading (DB2 Universal Database Enterprise - Extended Edition for the Solaris Operating Environment and DB2 Universal Database Enterprise - Extended Edition for Windows NT) 95
BINARY WITH PRECISION column definition 129
books 277, 287

C

call stack 170
CCSIDs 130
changing TCP/IP port entries 126
circular logging 242
CLI/ODBC/JDBC trace 186

client

APPC connections 59
 host 67
 host connections 61
 IPX/SPX connections 53
 named pipe connections 58
 NetBIOS connections 55
 TCP/IP connections 49
 troubleshooting 4, 43
Client Configuration Assistant 45
code page conversions 73
collecting data for DB2 customer support 11
column is no longer visible in the grid 136
command line processor, unable to use 35
commands, AIX
 errpt 213
 lsattr 213
 lsp 213
commands, DB2
 db2_all (DB2 Universal Database Enterprise - Extended Edition for the Solaris Operating Environment and DB2 Universal Database Enterprise - Extended Edition for Windows NT;) 215
 db2_call_stack 170
 db2atld (DB2 Universal Database Enterprise - Extended Edition for the Solaris Operating Environment and DB2 Universal Database Enterprise - Extended Edition for Windows NT) 93
 db2bfd 15
 db2cat 16
 db2ckmig 20
 db2dart 16
 db2drdat 185
 db2evmon 18
 db2expln 39
 db2fexmt 39
 db2flsn 16, 115
 db2ipxad 16
 db2level 17
 db2look 17

- commands, DB2 (*continued*)
 - db2recr 17
 - db2set 44
 - db2snmpd 174
 - db2sql92 17
 - db2start 88, 89
 - db2tbst 17
 - db2trc 177
 - db2untag 17
 - ddcstrc (DB2 Connect) 184
 - dynexpln 39
 - list applications 7
 - load 33
 - monitoring 18
 - netstat (DB2 Universal Database Enterprise - Extended Edition for the Solaris Operating Environment and DB2 Universal Database Enterprise - Extended Edition for Windows NT;) 216
 - rah (DB2 Universal Database Enterprise - Extended Edition for the Solaris Operating Environment and DB2 Universal Database Enterprise - Extended Edition for Windows NT;) 215
 - register 53
 - reorg 38
 - reorgchk 38
 - runstats 38
 - troubleshooting 15
 - commands, OS/2
 - cmtrace (SNA trace) 186
 - epw 220
 - epwinst 220
 - epwrout 224
 - pstat 225
 - syslog 218
 - commands, UNIX-based
 - cmtrace (SNA trace) 186
 - correct syntax 35
 - dbx 210
 - df 214
 - iostat 7, 214
 - ipcrm 8
 - ipcs 8
 - kill 8
 - netstat 215
 - ps 212
 - pstack (Solaris) 214
 - commands, UNIX-based (*continued*)
 - spmon (DB2 Universal Database Enterprise - Extended Edition for the Solaris Operating Environment and DB2 Universal Database Enterprise - Extended Edition for Windows NT;) 215
 - truss (SVR4) 214
 - vmstat 214
 - xmperf 213
 - communicating with IBM 11
 - communication problems 126
 - configuration parameters
 - DB2_RR_TO_RS 38
 - diaglevel 158
 - diagpath 158
 - dlchktime 38
 - locklist 37
 - locktimeout 37
 - logfilsiz 21
 - logprimary 21
 - logsecond 21
 - maxagents 153
 - maxlocks 37
 - nname (NetBIOS) 56
 - notifylevel 158
 - num_poolagents 153
 - sheapthres 40
 - sortheap 40
 - configure multisite update wizard 292
 - configuring clients 44
 - connecting to a Sybase source fails 120
 - connection
 - analyzing events 15
 - changing number for DRDA 65
 - testing on the server 46
 - troubleshooting 43
 - unsuccessful 47
 - Control Center
 - online information 236
 - OS/2 online help not working 237
 - overview 14
 - search utility 236
 - UNIX online help not working 237
 - coordinator agent 153
 - core files
 - DB2 169
 - UNIX 210
 - corruption, checking for 16
 - create database wizard 293
 - create table wizard 293
 - customer support, contacting 11
- ## D
- data
 - architectural integrity, checking 16
 - backups 28
 - binary, loading (DB2 Universal Database Enterprise - Extended Edition for the Solaris Operating Environment and DB2 Universal Database Enterprise - Extended Edition for Windows NT) 95
 - format (DB2 Universal Database Enterprise - Extended Edition for the Solaris Operating Environment and DB2 Universal Database Enterprise - Extended Edition for Windows NT) 94
 - importing 34
 - indexing (DB2 Universal Database Enterprise - Extended Edition for the Solaris Operating Environment and DB2 Universal Database Enterprise - Extended Edition for Windows NT) 96
 - loading 33
 - splitting and loading (DB2 Universal Database Enterprise - Extended Edition for the Solaris Operating Environment and DB2 Universal Database Enterprise - Extended Edition for Windows NT) 93
 - Data Warehouse Center
 - files have disappeared 123
 - hangs 128
 - services, cannot install 122
 - will not start 124
 - database
 - administration
 - troubleshooting 13
 - alias 43
 - consistency 20
 - creating (DB2 Universal Database Enterprise - Extended Edition for the Solaris Operating Environment and DB2 Universal Database Enterprise - Extended Edition for Windows NT) 91

- database (*continued*)
 - design 13
 - directories 248
 - files 249
 - non-recoverable 241
 - objects, displaying with Control Center 14
 - production, mimicking 17
 - recoverable 241
 - recovery (DB2 Universal Database Enterprise - Extended Edition for the Solaris Operating Environment and DB2 Universal Database Enterprise - Extended Edition for Windows NT) 96
 - remote 44
 - system catalog statistics, mimicking 17
- database directory 43
- database logs 242
- database manager
 - monitoring 18
 - monitoring with Control Center 15
- database partition 87
- Database System Monitor
 - Event Monitor 234
 - overview 233
 - Snapshot Monitor 233
- DataJoiner Version 2 CREATE SERVER MAPPING SQL statements fail 117
- DB2
 - administrative tools 14
 - DB2 Universal Database Enterprise - Extended Edition for the Solaris Operating Environment and DB2 Universal Database Enterprise - Extended Edition for Windows NT 87
 - host connections 61
 - information, finding 9
 - maintaining 10
 - SQL0973 error 134
- db2_call_stack 102
- DB2 Connect
 - tracing 184
 - troubleshooting 61
- DB2 customer support, contacting 11
- DB2 Data Links Manager
 - data access token 106
 - Data Links File Manager 106
- DB2 Data Links Manager (*continued*)
 - Data Links Filesystem Filter 106
 - export operation 110
 - import operation 110
 - load considerations 111
 - migration considerations 112
 - overview 105
 - performing a backup 109
 - recovery 107
 - troubleshooting 105
 - Windows NT 112
- DB2 facilities 147
 - alert 150
 - diagnostics 147
 - logging 147
 - process identification 149
 - system logging 149
 - trace 148
- DB2 family
 - unable to connect to database 117
- DB2 for AS/400
 - cannot import tables 128
 - unable to connect 119
- DB2 Index Advisor 239
- DB2 library
 - books 277
 - Information Center 291
 - language identifier for books 285
 - late-breaking information 286
 - online help 288
 - ordering printed books 287
 - printing PDF books 286
 - searching online information 294
 - setting up document server 293
 - structure of 277
 - viewing online information 290
 - wizards 292
- DB2 Log 126
- DB2 Process Model
 - differences between Intel and UNIX 154
 - displaying threads (OS/2) 225
 - overview 151
 - ps command (UNIX) 212
 - viewing 155
- DB2 Query Patroller
 - overview 85
 - troubleshooting 85
- DB2 Satellite Edition
 - troubleshooting 141
- DB2 Trace Facility 101
- DB2 Universal Database Enterprise - Extended Edition, installation 88
- DB2 Universal Database Enterprise - Extended Edition for the Solaris Operating Environment and DB2 Universal Database Enterprise - Extended Edition for Windows NT
 - commands 215
 - determining failing nodes 88
 - developing applications 98
 - monitoring applications 75
 - processes 154
 - severe error 99
 - starting 88
 - troubleshooting 87
- DB2 Universal Database Server , troubleshooting tools 15
- DB2 wizards 15
- DB2COMM registry value 47
- db2dart tool 16
- db2diag.log
 - additional information in syslog (UNIX) 207
 - diagnosing if listeners are up 153
 - dump files 169
 - example with DB2 trace 181
 - examples 163
 - header 160
 - inconsistent data 21
 - interpreting 160
 - NetBIOS entries 57
 - pruning 160
 - relationship to Windows NT event viewer 231
 - SQLCA structures 162
 - tracing if insufficient 177
 - trap files 170
 - troubleshooting client connections 47
- DB2MEMDBG registry variable 102
- DBA4745N 237
- deadlocks 36
- Desktop Management Interface (DMI) service layer 173
- diaglevel, setting 158
- diagnostic tools
 - OS/2 217
 - UNIX 207
 - Windows 95 and Windows 98 229
 - Windows NT 229
- diagpath, setting 158

- directories
 - changing with Control Center 14
 - directory
 - database 248
 - Distributed Relational Database Architecture (DRDA)
 - tracing 185
 - troubleshooting 61
 - Dr. Watson (Windows) 229
 - dump files
 - core file (UNIX) 210
 - OS/2 alerts 225
 - traces 178
 - dump utility, standalone (OS/2) 217
- E**
- environment variables 44
 - error code 163
 - error log (OS/2) 218
 - error record, FFST for OS/2 error log 223
 - Event Analyzer 15
 - Event Monitor
 - matching records with their application 235
 - Output Records 235
 - reading an event monitor trace 235
 - event snapshots 18
 - event viewer (Windows NT) 231
 - exceptions
 - responding to 6
 - trap file 170
 - explain facility 39
 - export message files 172
- F**
- FFDC 157
 - FFST/2 implementation (OS/2) 219
 - file
 - database 249
 - files
 - autoexec.bat (OS/2) 54
 - config.sys 54
 - core (UNIX) 210
 - db2alert.log 173
 - db2cc.log 171
 - db2cli.ini 44, 76
 - db2diag.log 160
 - dbttrc.trc 184
 - dump 169
 - net.cfg (OS/2) 54
 - nwcalls.dll (Win-OS/2) 55
 - odbc.ini 44
 - files (*continued*)
 - services (TCP/IP) 50
 - size limit 34
 - syslog (UNIX) 207
 - system (Solaris) 215
 - TLI*.DLL (Novell IPX/SPX) 53
 - tli_spx.dll (Win-OS/2) 55
 - trap 170
 - firewall 151
 - first failure data capture (FFDC)
 - alerts 173
 - configuration parameters 158
 - db2diag.log 160
 - dump files 169
 - first failure data capture 157
 - interpreting hexadecimal codes 163
 - introduction 11
 - overview 157
 - SQLCA structures 162
 - stack traceback 170
 - tracing if insufficient 177
 - trap files 170
 - First Failure Data Capture (FFDC) 101
 - fix pack 20
 - fix packs, applying 10
 - FixPaks 20
 - format of data (DB2 Universal Database Enterprise - Extended Edition for the Solaris Operating Environment and DB2 Universal Database Enterprise - Extended Edition for Windows NT) 94
 - forums 9
 - free space, identifying (UNIX) 214
 - ftp sites 10
- G**
- general protection faults (GPF), responding to 6
 - GET SNAPSHOT 99
 - graphical user interface (GUI)
 - administrative tools 14
- H**
- hangs, responding to 7
 - hardware errors
 - identifying (AIX) 213
 - identifying (HP-UX) 214
 - hexadecimal values in db2diag.log 163
 - hierarchy when shutting down 103
 - history of problems, maintaining 3
- host
 - troubleshooting connections to 61
- HPS switch, checking (DB2 Universal Database Enterprise - Extended Edition for the Solaris Operating Environment and DB2 Universal Database Enterprise - Extended Edition for Windows NT;) 215
- HTML
 - sample programs 285
- hung applications 74
- I**
- I/O activities, monitoring (UNIX) 214
 - IBM, contacting 11
 - idle agent 153
 - import message files 172
 - importing data 34
 - importing source tables 129
 - index
 - architectural integrity, checking 16
 - failing in DB2 Universal Database Enterprise - Extended Edition for the Solaris Operating Environment and DB2 Universal Database Enterprise - Extended Edition for Windows NT 96
 - inconsistent, re-creating 17
 - Index DB2 wizard 39
 - Index SmartGuide 239
 - index wizard 293
 - indoubt transactions (DB2 Universal Database Enterprise - Extended Edition for the Solaris Operating Environment and DB2 Universal Database Enterprise - Extended Edition for Windows NT) 98
 - Information Center 291
 - overview 14
 - Troubleshooting Page 238
 - information on DB2, finding 9
 - initialization
 - fails
 - 3043 disk or network error 123
 - 9159: Win32 function RegOpenKeyEx failed message 124
 - add install path to Windows NT service path 123
 - unable to connect 124

- installing
 - Netscape browser 291
 - Integer 129
 - Internet Engineering Task Force (IETF) 173
 - InterProcess Communications (IPC) 97
 - IPX/SPX protocol 53
 - isolation levels (CS, RS) 38
- J**
- JDBCTRACE, keyword 196
 - JDBCTRACEFLUSH, keyword 197
 - JDBCTRACEPATHNAME, keyword 198
 - job scheduling, using Control Center 14
- K**
- keepalive settings (TCP/IP) 51
 - kernel
 - debugging, OS/2 217
 - updating, (UNIX) 28
- L**
- LAN, determining availability 48
 - Lana number (Windows 95) 56
 - language identifier
 - books 285
 - late-breaking information 286
 - LIST APPLICATIONS 99
 - listeners
 - definition 153
 - testing 46
 - load message files 172
 - LOAD utility 93
 - loading data, troubleshooting
 - tips 33
 - location of problem, determining 4
 - locking 36
 - log space 21
 - logging
 - archive 242
 - circular 242
 - logging, all platforms 157
 - logon error 125
 - logs
 - active 242
 - database 242
 - increasing space for importing data 34
 - online archived logs 242
 - system log (OS/2) 218
 - loops
 - in applications 74
 - responding to 7
- Lotus Approach 75
- M**
- maintaining DB2 10
 - Management Information Base (MIB) 173
 - media management with Control Center 14
 - memory, identifying (AIX) 213
 - memory corruption 102
 - memory leakage 102
 - message files, import, export and load 172
 - messages, responding to 5
 - Microsoft Access 75
 - error 134
 - Microsoft Query 75
 - migration
 - troubleshooting 19
 - misc subdirectory, tools 15
 - mobile 141
 - monitor switches 18
 - monitoring database manager with Control Center 15
 - MVS, connections 61
- N**
- named pipes 58
 - NAMEFMT not a valid command 121
 - NetBIOS connections lost 30
 - NetBIOS protocol 55
 - NetQuestion
 - overview 143
 - troubleshooting 143
 - Netscape browser
 - installing 291
 - network traffic, identifying (UNIX) 215
 - newsgroups 9
 - next key locking 38
 - node directory 44
 - nodes (DB2 Universal Database Enterprise - Extended Edition for the Solaris Operating Environment and DB2 Universal Database Enterprise - Extended Edition for Windows NT)
 - database partition 92
 - database partition size 93
 - definition 87
 - failing 88
 - hangs 90
 - no response 90
 - slow 90
 - won't start 90
- non-recoverable databases 241
- O**
- ODBC applications
 - configuring driver 75
 - messages 78
 - mixed case problems 77
 - optimizing the driver 76
 - troubleshooting tips 75
 - offline archived logs 242
 - online archived logs 242
 - online help 9, 288
 - online information
 - searching 294
 - viewing 290
 - optimizer, viewing access plans 14
 - origins of problems, determining 4
 - OS/2 commands 217
 - OS/390, connections 61
- P**
- packed descriptors for tables, viewing 16
 - page rate, monitoring (UNIX) 214
 - paging space, identifying (AIX) 213
 - passwords 27
 - patches, applying 10
 - PDF 286
 - performance
 - data, displaying (UNIX) 213
 - improving 38
 - monitoring with Control Center 15
 - using the database monitor 18
 - performance configuration wizard 293
 - Performance Monitor
 - Windows NT 231
 - pervasive computing 141
 - point-in-time monitoring 18
 - point of recovery 243
 - prepare of SQL statement 129
 - preserving CCSIDs in a CREATE TABLE statement 130
 - printing PDF books 286
 - problem description, determining 4
 - problem determination 3
 - process status command 212
 - promoting a step that uses a DB2 for OS/390 warehouse target
 - fails 132
 - PTFs, applying 10
- R**
- rebinding 35
 - recoverable databases 241

- recovery
 - DB2 server 28
 - DB2 Universal Database
 - Enterprise - Extended Edition for the Solaris Operating Environment and DB2
 - Universal Database Enterprise - Extended Edition for Windows NT 96
 - operating system 7
 - point of 243
 - recovery concepts 241
 - redbooks 9
 - registry values 44
 - release notes 286
 - remote debugging (OS/2) 217
 - replication
 - DB2 DataPropagator 114
 - overview 113
 - troubleshooting 113
 - resources, increasing (IPX/SPX) 54
 - responding to DB2 users 4
 - RESTART DATABASE 97
 - restore wizard 293
 - restoring data 30
 - restoring Version 1 database image (DB2 for OS/2) 17
 - return code 7356 129
 - return codes
 - byte-reversal 163
 - example 166
 - list 253
 - roadmaps
 - alerts 147
 - diagnostics 147
 - logging 147
 - process identification 147
 - tracing 147
 - ROLLFORWARD PENDING 31
- S**
- SAM tool (HP-UX) 214
 - sample programs
 - cross-platform 285
 - HTML 285
 - satellite control database 141
 - Satellite Edition 141
 - schema names, considerations
 - during migration 20
 - screen captures (Windows) 231
 - searching
 - online information 292, 294
 - segmentation violation
 - logs 170
 - responding to 6
 - server
 - administrative tools 14
 - client connections 47
 - collecting data for DB2 Customer Support 40
 - disk alignment 34
 - failure 20
 - raw devices 34
 - recovery 33
 - restoring 30
 - SQL0298N 32
 - SQL0902C 28, 31
 - SQL1024N 32
 - SQL1032N 26
 - SQL1277N 31, 33
 - SQL1403N 27
 - SQL2419N 30
 - SQL4908N 29
 - Terminal Server 26
 - timing out (NetBIOS) 57
 - troubleshooting 4, 13
 - services file (TCP/IP) 50
 - setting up document server 293
 - Setup Communications tool 45
 - shutdown hierarchy 103
 - signal 170
 - Simple Network Management Protocol (SNMP) 173
 - SmartGuides
 - wizards 292
 - Snapshot Monitor 233
 - SNMP agent 173
 - SNMP subagent, starting 174
 - software fixes, applying 10
 - Solaris OS commands (useful for debugging) 213
 - splitting data (DB2 Universal Database Enterprise - Extended Edition for the Solaris Operating Environment and DB2 Universal Database Enterprise - Extended Edition for Windows NT) 93
 - SQL
 - prepare of statement fails 129
 - SQL Advise Facility 239
 - SQL error code 271
 - SQL messages
 - definition 5
 - responding to 5
 - SQL0204N 77
 - SQL0283N 246
 - SQL0290N 246
 - SQL0294N 17
 - SQL0805N 35
 - SQL0818N 35
 - SQL messages (*continued*)
 - SQL0911N 36
 - SQL0964C 21
 - SQL1003N 75
 - SQL1004C 167
 - SQL1015N 100
 - SQL1016N 28
 - SQL1018N 28
 - SQL1034C 97, 100
 - SQL1042C 28, 97
 - SQL1061W 98
 - SQL1109N 55
 - SQL1224N 100
 - SQL1402N 66
 - SQL1403N 49, 66
 - SQL2216N 246
 - SQL30073N 67
 - SQL30081N (IPX/SPX) 53
 - SQL30081N (NetBIOS) 55
 - SQL30081N (TCP/IP) 49
 - SQL30082N 66
 - SQL30082N (NetBIOS) 58
 - SQL3805N 33
 - SQL5043N 47, 51
 - SQL0973 error 134
 - SQL1042C 20
 - SQL4400N 237
 - SQLCA (SQL communication area) 271
 - DB2 Universal Database
 - Enterprise - Extended Edition for the Solaris Operating Environment and DB2 Universal Database Enterprise - Extended Edition for Windows NT nodes 88
 - interpreting 162
 - SQLSTATE 5
 - stack traceback
 - displaying (Solaris) 214
 - for suspended system 9
 - for UNIX-based systems 170
 - starting a trace 179
 - step running VWPRCPY fails 137
 - stored procedure 82
 - developing 71
 - fenced 154
 - Stored Procedure Builder 238
 - subagent 153
 - suspended applications 74
 - suspensions, responding to 7
 - Sybase Open Client Context
 - fails 120
 - symptom record, FFST for OS/2 223

- syntax diagrams, how to read x
- syslog (UNIX) 207
- system calls, tracing (UNIX) 214
- system catalog statistics,
 - mimicking 17
- system configuration, using Control Center 14
- system file (Solaris) 215
- system log (SYSLOG)
 - OS/2 218
 - UNIX 207
- system resources, increasing (IPX/SPX) 54
- Systems Network Architecture (SNA)
 - overview 61
 - routing alerts (OS/2) 224
 - tracing 186
- SystemView agent 173

T

- table
 - dropped table PIT recovery 243
 - recovery 241
 - system catalog tables,
 - mimicking 17
 - viewing packed descriptors 16
- table space
 - ARCHIVE LOGGING 245
 - Changing OFFLINE to ONLINE
 - table space state 246
 - container, using again with
 - db2untag 17
 - DROP TABLESPACE 247
 - OFFLINE 245
 - recovery 241
 - RESTART ... DROP
 - PENDING 248
 - RESTART ... DROP PENDING
 - Table Space State 247
 - RESTART ... DROP PENDING
 - TABLESPACE 246
 - state, getting description of with
 - db2tbst 17
- target table not being journaled 131
- TCP/IP problems 126
- TCP/IP protocol 49
- test tool to mimic production
 - databases 17
- threads 151
- tools
 - client connectivity 44
 - database administration 14
 - db2ocat 80
- trace
 - CLI/ODBC/JDBC 186

- TRACE, keyword 199
- TRACECOMM, keyword 200
- TRACEFILENAME, keyword 200
- TRACEFLUSH, keyword 201
- TRACEPATHNAME, keyword 202
- TRACEPIDTID, keyword 204
- traces
 - administration tools (DB2) 183
 - DB2 177
 - DB2 Connect 184
 - db2trc 177
 - DRDA 185
 - dumping (DB2) 178
 - examples (DB2) 180
 - SNA 186
 - turning off (DB2) 179
 - verifying (DB2) 180
- TRACETIMESTAMP, keyword 203
- trap 170
- troubleshooting 117
 - abends, responding to 6
 - APPC protocol 59
 - applications 71
 - clients 43
 - contacting IBM 11
 - database administration 13
 - DB2 Universal Database
 - Enterprise - Extended Edition for the Solaris Operating Environment and DB2 Universal Database Enterprise - Extended Edition for Windows NT 87
 - db2diag.log 157
 - getting started 3
 - history of problems,
 - maintaining 3
 - host connections 61
 - information, finding 9
 - IPX/SPX protocol 53
 - loops, responding to 7
 - maintaining products 10
 - messages, responding to 5
 - named pipes 58
 - NetBIOS protocol 55
 - origins and symptoms of
 - problems, determining 4
 - problem description,
 - determining 4
 - return codes 253
 - servers 13
 - suspensions, responding to 7
 - TCP/IP protocol 49
 - where to find a problem 4
- Troubleshooting Page 238

U

- UNIX-based commands 213
- updates to DB2, applying 10
- user-defined functions (UDFs)
 - developing 71
 - fenced 154
- user forums 9
- user name
 - authentication on host 66
 - case sensitivity 66
 - not valid on Windows 95 and Windows 98 49
- user names 27
- using a previous release of an AS/400 system 129
- utilities, unable to use 35

V

- viewing
 - online information 290
- Visual Explain 14
- VM, connections 61
- VSE, connections 61
- VWPFFTP data wasn't transferred 137

W

- warehouse logger 126
- WarpCenter (OS/2 Warp Version 4) 217
- watchdog process 155
- Windows 2000
 - cannot install Data Warehouse Center services 122
- Windows NT
 - cannot install Data Warehouse Center services 122
- wizards 15
 - add database 292, 293
 - backup database 292
 - completing tasks 292
 - configure multisite update 292
 - create database 293
 - create table 293
 - create table space 293
 - index 293
 - performance configuration 293
 - restore database 293
- Work in Progress
 - incorrect start or end date or time 134
- World Wide Web 9

Contacting IBM

If you have a technical problem, please review and carry out the actions suggested by the *Troubleshooting Guide* before contacting DB2 Customer Support. This guide suggests information that you can gather to help DB2 Customer Support 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.

If you live in the U.S.A., then you can call one of the following numbers:

- 1-800-237-5511 for customer support
- 1-888-426-4343 to learn about available service options

Product Information

If you live in the U.S.A., then you can call one of the following numbers:

- 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.

<http://www.ibm.com/software/data/>

The DB2 World Wide Web pages provide current DB2 information about news, product descriptions, education schedules, and more.

<http://www.ibm.com/software/data/db2/library/>

The DB2 Product and Service Technical Library provides access to frequently asked questions, fixes, books, and up-to-date DB2 technical information.

Note: This information may be in English only.

<http://www.elink.ibm.com/pbl/pbl/>

The International Publications ordering Web site provides information on how to order books.

<http://www.ibm.com/education/certify/>

The Professional Certification Program from the IBM Web site provides certification test information for a variety of IBM products, including DB2.

ftp.software.ibm.com

Log on as anonymous. In the directory /ps/products/db2, you can find demos, fixes, information, and tools relating to DB2 and many other products.

comp.databases.ibm-db2, bit.listserv.db2-l

These Internet newsgroups are available for users to discuss their experiences with DB2 products.

On CompuServe: GO IBMDB2

Enter this command to access the IBM DB2 Family forums. All DB2 products are supported through these forums.

For information on how to contact IBM outside of the United States, refer to Appendix A of the *IBM Software Support Handbook*. To access this document, go to the following Web page: <http://www.ibm.com/support/>, and then select the IBM Software Support Handbook link near the bottom of the page.

Note: In some countries, IBM-authorized dealers should contact their dealer support structure instead of the IBM Support Center.



Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

GC09-2850-01

