IBM DB2 Universal Database

# Troubleshooting Guide

*Version 6*

S10J-8169-01

IBM DB2 Universal Database

# Troubleshooting Guide

*Version 6*

S10J-8169-01

Before using this information and the product it supports, be sure to read the general information under "Appendix E. Notices" on page 249.



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.

# Contents


**Welcome to DB2** . . . . . vii
Who Should Read This Book? . . . . . vii
Conventions . . . . . vii
How to Read the Syntax Diagrams . . . viii

**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 SQL Codes . . . . . 5
Responding to Abends . . . . . 6
Responding to Suspensions and Loops 6
Using All Available Information . . . . . 8
Updating DB2 Products . . . . . 9
Using Logged Data and Taking Traces . . 10
Contacting IBM for Support . . . . . 10

**Chapter 2. Troubleshooting the DB2 Universal Database Server** . . . . . 11
DB2 Administrative Tools . . . . . 12
Graphical Tools . . . . . 12
Miscellaneous Troubleshooting Tools 13
The Database System Monitor . . . . 15
Troubleshooting Tips . . . . . 17
Problems Migrating to Version 6 . . . 17
Server Failures . . . . . 18
SQL1403N received when authenticating usernames or groups . . . . . 19
Administration Difficulties . . . . . 20
Backing Up and Restoring Data . . . . 21
Data Not Loading . . . . . 25
Difficulties Importing Data . . . . . 26
Problems Using Commands, Utilities, or the Command Line Processor . . . . 26
Tables or Rows Locked . . . . . 27
Improving Database Performance . . . 29
Still Have Problems? . . . . . 31

**Chapter 3. Troubleshooting on the Client** 33
Overview of Client Connectivity . . . . 33
Client Connectivity Tools . . . . . 34
Troubleshooting Tips . . . . . 35
Determining the Scope of a Client Problem . . . . . 36
Initial Connection after Installation Fails 38
Clients Suddenly Experiencing Problems 38
User Name Not Valid on Windows 95 and Windows 98 . . . . . 39
TCP/IP Problems . . . . . 40
IPX/SPX Problems . . . . . 43
NetBIOS Problems . . . . . 46
Problems When Using Named Pipes 48
APPC Problems . . . . . 49
Still Have Problems? . . . . . 49

**Chapter 4. Troubleshooting Host Communications** . . . . . 51
Troubleshooting Host Connections . . . . 52
Troubleshooting Tips for DB2 Connect . . 53
Initial Questions . . . . . 53
Problems Making an Initial Connection after Installation . . . . . 54
Problems Encountered after Initial Connection . . . . . 54
DB2 Connect Connection Using SNA Does Not Work . . . . . 55
Changing the Number of Connections 56
Authentication Problems . . . . . 56
Windows Clients Will Not Connect. . . 58
Troubleshooting Tips for Host Connections to DB2 Universal Database . . . . . 58
Still Have Problems? . . . . . 59

**Chapter 5. Troubleshooting Applications** 61
Troubleshooting Tips . . . . . 61
Developing Applications . . . . . 62
Compiling and Linking Applications 63
Running Applications . . . . . 64
Using CLI/ODBC/JDBC Applications 65
Using SQLJ Applications . . . . . 69
Using the db2ocat Tool . . . . . 69
Stored Procedures . . . . . 72
Still Have Problems? . . . . . 72

**Chapter 6. Troubleshooting DB2 Query Patroller . . . . . 75**
Still Have Problems? . . . . . 76

**Chapter 7. Troubleshooting DB2 Universal Database Enterprise - Extended Edition . . . . . 77**
Troubleshooting Tips . . . . . 77
DB2 Universal Database Enterprise - Extended Edition Will Not Install . . . 78
Severe Error Occurs on System . . . . 78
Determining Failing Node. . . . . 79
Database Manager Will Not Start or Hangs . . . . . 79
Cannot Create Database . . . . . 81
Commands Cannot Be Issued . . . . 82
Problems Backing Up On Database Partitions . . . . . 82
Problems with LOAD and with Splitting and Loading Data . . . . . 82
Problems with Indexing Data . . . . 85
Problems during Recovery . . . . . 86
Developing Applications in DB2 Enterprise - Extended Edition . . . . 87
Still Have Problems? . . . . . 87

**Chapter 8. Troubleshooting DB2 Data Links Manager . . . . . 89**
DB2 Data Links Manager Considerations 89
DB2 Data Links Manager . . . . . 90
Data Access Token . . . . . 90
Recovery Data Links Manager Environment . . . . . 91
Backup Data Links Manager Environment . . . . . 93
Export Operation. . . . . 94
Import Operation . . . . . 94
Load Considerations . . . . . 95
DB2 Data Links Manager for Windows NT . . . . . 96
Still Have Problems? . . . . . 96

**Chapter 9. Troubleshooting Replication 97**
DpropR . . . . . 97
Using the Log Header Control File with db2flsn . . . . . 98
Example of using the db2flsn command and the db2diag.log file . . . . . 98
DpropR Enhancements. . . . . 99
Still Have Problems? . . . . . 100

**Chapter 10. Troubleshooting DB2 Satellite Edition . . . . . 101**
Still Have Problems? . . . . . 102

**Chapter 11. Troubleshooting NetQuestion 103**
Still Have Problems? . . . . . 103

**Part 2. Advanced DB2 Troubleshooting . . . . . 105**

**Chapter 12. Logs and Traces Roadmap 107**

**Chapter 13. The DB2 Process Model . . 111**
Description of the DB2 Process Model. . . 111
Differences between Intel and UNIX 114
Viewing System Threads and Processes 115

**Chapter 14. Logged Information . . . . 117**
First Failure Data Capture. . . . . 117
Setting Configuration Parameters Affecting FFDC . . . . . 118
Interpreting the db2diag.log file. . . . . 120
Interpreting db2diag.log file Entry Headers. . . . . 120
Interpreting an SQLCA Structure . . . 122
Interpreting Hexadecimal Codes . . . 123
Some Examples for Interpreting the db2diag.log file . . . . . 123
Dump Files . . . . . 128
Trap Files . . . . . 129
Gathering Stack Traceback Information on UNIX-Based Systems . . . . . 129
Using the Administrative Tools Log . . . 130
Administrative Tools Log File Sample 131
Using Message Files. . . . . 131
Alerts . . . . . 132
SNMP Alerts . . . . . 132

**Chapter 15. Traces . . . . . 135**
Using the DB2 Trace Facility (db2trc) . . . 135
Starting a DB2 Trace . . . . . 137
Verifying a DB2 Trace . . . . . 138
Examples of Using db2trc . . . . . 139
Taking Other Traces. . . . . 141
Administration Tools Trace . . . . . 142
Taking a Graphical Interface Trace . . . 143
Taking a DB2 Connect Trace (ddcstrc) 143
Taking a DRDA Trace (db2drdat) . . . 143
Taking an SNA Trace . . . . . 145

Using the DB2 CLI/ODBC/JDBC Trace Facility . . . . . 145

**Chapter 16. Diagnostic Tools for UNIX-Based Systems . . . . . 161**
Using the System Error Log (syslog) . . . 161
Setting Up the System Error Log . . . 162
Understanding the System Error Log 163
Using the Core File . . . . . 164
Accessing Core File Information. . . . 164
Viewing Process Status Using the ps Command . . . . . 166
Other Commands . . . . . 167
AIX Troubleshooting Commands . . . 167
UNIX-Based Troubleshooting Commands 168
Performance Monitoring Tools . . . . 168
Commands for DB2 Enterprise - Extended Edition. . . . . 169

**Chapter 17. Diagnostic Tools for OS/2 171**
Using the OS/2 Logging Facility . . . . 172
Preliminary Considerations . . . . . 172
Using OS/2 Logging Facility Commands 172
Using the FFST for OS/2 Utility. . . . . 174
Adding the FFST for OS/2 Program Group to the Desktop Manager Window 174
Configuring and Enabling the FFST for OS/2 Utility . . . . . 174
Accessing FFST for OS/2 Error Records 177
Routing Generic Alerts. . . . . 178
Using the pstat Command . . . . . 179

**Chapter 18. Diagnostic Tools for Windows 32-bit Operating Systems . . . 183**
Diagnostic Tools for Windows 95 and Windows 98 . . . . . 183
Diagnostic Tools for Windows NT . . . . 183
Using Administrative Tools Provided by Windows NT . . . . . 184
Taking Screen Captures . . . . . 185
Using DB2-Supplied Tools. . . . . 186

**Chapter 19. GUI and Monitoring Tools 187**
Database System Monitor . . . . . 187
Using the Snapshot Monitor . . . . . 187
Using the Event Monitor . . . . . 188
Control Center . . . . . 190
Using Search Utility for DB2 Online Information . . . . . 190
Troubleshooting Tips . . . . . 191
Troubleshooting Page . . . . . 191
Stored Procedure Builder . . . . . 192
DB2 Index Advisor . . . . . 192

**Chapter 20. Table and Table Space Recovery . . . . . 195**
Dropped Table PIT Recovery. . . . . 195
Table Space OFFLINE . . . . . 196
Archive Logging . . . . . 197
Table Space Usability . . . . . 197
RESTART ... DROP PENDING TABLESPACE. . . . . 198
RESTART ... DROP PENDING Table Space State. . . . . 198
Table Space Usability . . . . . 199
RESTART ... DROP PENDING . . . . 199

**Part 3. Appendixes . . . . . 201**

**Appendix A. How the DB2 Library Is Structured. . . . . 203**
Completing Tasks with SmartGuides . . . 203
Accessing Online Help . . . . . 204
DB2 Information – Hardcopy and Online 206
Viewing Online Information . . . . . 213
Accessing Information with the Information Center . . . . . 214
Setting Up a Document Server . . . . . 215
Searching Online Information . . . . . 216
Printing the PostScript Books. . . . . 216
Ordering the Printed Books . . . . . 217

**Appendix B. Using the Problem Determination Tools Folder. . . . . 219**
Accessing Internet Information for DB2 for OS/2. . . . . 219
Accessing Internet Information for DB2 for Windows NT . . . . . 219
Updating DB2 Products . . . . . 220
Setting Diagnostic Levels . . . . . 220
Collecting Diagnostic Information . . . . 221
Sending or Receiving Information . . . . 222
Tracing . . . . . 223

**Appendix C. SQL Communications (SQLCA) . . . . . 225**
Viewing the SQLCA Interactively . . . . 225
SQLCA Field Descriptions. . . . . 225
Order of Error Reporting . . . . . 229

DB2 Enterprise - Extended Edition Usage of the SQLCA. . . . . 229

**Appendix D. DB2 Internal Return Codes 231**

**Appendix E. Notices . . . . . 249**

Trademarks . . . . . 250
Trademarks of Other Companies . . . . 250

**Index . . . . . 253**

**Contacting IBM . . . . . 261**

# 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 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 (refer to the *Administration Guide, Design and 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 A. How the DB2 Library Is Structured" on page 203.

## 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 to indicate 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.

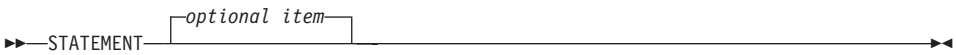
The ▸— symbol indicates that a statement is continued from the previous line.

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.

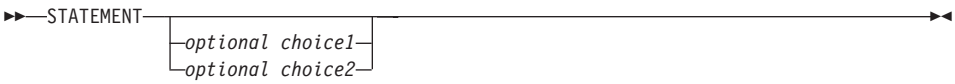

If an optional item appears above the main path, that item has no effect on the execution of the statement and is used only for readability.



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 will appear above the main path and the remaining choices will be 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 11
- "Chapter 3. Troubleshooting on the Client" on page 33
- "Chapter 4. Troubleshooting Host Communications" on page 51
- "Chapter 5. Troubleshooting Applications" on page 61
- "Chapter 6. Troubleshooting DB2 Query Patroller" on page 75
- "Chapter 7. Troubleshooting DB2 Universal Database Enterprise - Extended Edition" on page 77
- "Chapter 8. Troubleshooting DB2 Data Links Manager" on page 89
- "Chapter 9. Troubleshooting Replication" on page 97
- "Chapter 10. Troubleshooting DB2 Satellite Edition" on page 101
- "Chapter 11. Troubleshooting NetQuestion" on page 103.

# 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 8
- "Updating DB2 Products" on page 9
- "Using Logged Data and Taking Traces" on page 10
- "Contacting IBM for Support" on page 10.


## 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 complete description of any messages that are received, including SQL codes, associated reason codes, or system error codes.
  - Any SQL state received. SQL states are useful for diagnosing problems, because they are consistent across all platforms. For a list of SQL states, 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 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, 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 Service.

## 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 SQL code on a client, then investigate the SQL code on that client. (See "Responding to Unexpected Messages or SQL Codes" on page 5 for information.)

Often the SQL code alone is enough information to determine the source and cause of the problem. If the SQL code does not give enough information to determine the source of a problem, examine the db2diag.log file at the machine 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. If you know the date and time when the problem occurred, you can go directly to the corresponding db2diag.log entries. For information on this important file, see "First Failure Data Capture" on page 117. When viewing the file, keep in mind that the most recent conditions are always at the end.

## Responding to Unexpected Messages or SQL Codes

When you receive an unexpected message or SQL code, 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 8-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 SQL state received. SQL states are useful for diagnosing problems, because they are consistent across all platforms. For a list of SQL states, refer to 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 core files (for UNIX-based systems), event logs (for Windows NT), or SYSLOG files (for OS/2). For information, see "Part 2. Advanced DB2 Troubleshooting" on page 105.
   - 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.
   - 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 SQL code, SQL state, or message number. Read and follow the suggested actions.
3. Use the SQL code 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 Service:
   - The DB2 diagnostic log (db2diag.log), and any trap or dump files it mentions. See "First Failure Data Capture" on page 117.
   - The SQLCA structure from the db2diag.log or syslog files, or as captured by an application. See "Interpreting an SQLCA Structure" on page 122.
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 are at the same service level, especially if a Fix Pak has recently been installed. See "Updating DB2 Products" on page 9.
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 Service:
   - Any logged information, in particular:
     - The db2diag.log file and any trap or dump files it mentions. See "First Failure Data Capture" on page 117.
     - For UNIX, the syslog and core files. See "Chapter 16. Diagnostic Tools for UNIX-Based Systems" on page 161.
     - For OS/2, the SYSLOG file. See "Using the syslog Command" on page 173.
     - For Windows operating systems, Dr. Watson logs, and, for Windows NT, the event log. See "Chapter 18. Diagnostic Tools for Windows 32-bit Operating Systems" on page 183.
   - If the problem can be reproduced, a trace on the client and server may be helpful. Follow the steps in "Example of Tracing to a File" on page 139.

   See "Part 2. Advanced DB2 Troubleshooting" on page 105 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. 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.
   - 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 waiting for a lock 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 judgement to determine whether or not the applications are suspended or behaving as expected.
- 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`

  Work through the following steps **as a last resort**, only if the above steps did not stop the DB2 instance:
  a. Abruptly kill the DB2 instance with `db2stop -kill`
  b. Use the **kill** command to terminate any DB2 agents that cannot be stopped
  c. Use the **kill** command to terminate DB2 itself (db2sysc)
  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 sqllib/misc directory)
  - Use the following command to identify those resources that belong to the DB2 instance:

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

    Then use **ipcrm** to remove them.

2. 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 167)

- For the Windows NT operating system, use its performance monitor (see "Diagnostic Tools for Windows NT" on page 183)
- For the OS/2 system, use tools such as SPM/2 (see "Chapter 17. Diagnostic Tools for OS/2" on page 171)

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

3. If the problem persists, try to collect the following additional information before contacting DB2 Customer Service:
   - Any information logged by DB2. See "First Failure Data Capture" on page 117.
   - 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 139.
     - 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 AIX, issue **kill -36** against the db2sysc process.
       - For HP-UX, issue **kill -29** against the db2sysc process.
       - For Solaris Operating Environment, issue **kill -21** against the db2sysc process.

       For information, see "Gathering Stack Traceback Information on UNIX-Based Systems" on page 129.
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 A. How the DB2 Library Is Structured" on page 203.
- Online help is available:
  - For help with the DB2 graphical tools, use the **Help** menu.
  - For a description of SQL codes and other messages, type `db2 "? message"` from the command line, where *message* is the SQL code or other message number.
  - For a description of DB2 commands, type `db2 "? command"`, where *command* is the name of the command.

- Up-to-date bulletins and technical documentation are available from the DB2 Product and Service Technical Library on the World Wide Web, at **http://www.software.ibm.com/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.software.ibm.com/data/db2/udb/ad.
- Newsgroups and forums are provided on the Internet or from online service providers. For example, use the Internet newsgroup comp.databases.ibm-db2, or the DB2 Family Forum on CompuServe. For more information, 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 provides product updates 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 platform has its own Fix Pak numbering scheme, and its own schedule for creating fix packs.

There is an Internet site available for you to 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 World Wide Web page at http://www.software.ibm.com/data/db2/db2tech/indexsvc.html
- Use FTP to go to the anonymous server ftp.software.ibm.com. Then go to the directory /ps/products/db2/fixes.
- Use the Problem Determination folder on your desktop or the Troubleshooting tab from within the Information Center. See “Appendix B. Using the Problem Determination Tools Folder” on page 219.

**Note:** Once you download a Fix Pak, please ensure you fully review the associated README file before applying the Fix Pak. The README file contains any special instructions regarding the Fix Pak. Release Notes also accompany each Fix Pak, and may contain corrected

documentation as well as supplemental information regarding 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, refer to the *Quick Beginnings* for the appropriate platform.

## 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 often provide enough diagnostic information to solve a problem, without having to reproduce problems with traces.

For information on the logging, tracing, and alert facilities available to you, see "Part 2. Advanced DB2 Troubleshooting" on page 105.

## Contacting IBM for Support

Before contacting DB2 Customer Service, 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 World Wide Web site at http://www.software.ibm.com/data/db2/library/.
- 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 105.
- Output from **db2level**.

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

For more information on IBM contacts, see "Contacting IBM" on page 261.

# 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.

For more information on administering databases, refer to:

- 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.

It is beyond the scope of this book to cover all aspects of problem determination on a DB2 server. Refer to the information in the DB2 library to find out more about operations required of DB2 administrators. Refer to the Quick Beginnings books for information on migrating a database from a previous version. 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/recovery plan. If

you encounter serious problems, you may need a backup to restore your database and perhaps roll it forward.

[ ] 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 29.)

[ ] 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 file" on page 120.)

## DB2 Administrative Tools

The following sections outline some of the various tools available to help you perform database administrative tasks and DB2 troubleshooting. See "Improving Database Performance" on page 29 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 on 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 SmartGuides 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

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 it, click the left mouse button on a database object in the Control Center, and select **Explain SQL** from the pop-up menu. See also "Improving Database Performance" on page 29.

  **Note:** With IBM DB2 Universal DatabaseVersion 6, 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 it, use the left mouse button on a database object, and select **Snapshot monitoring** from the pop-up menu to see the monitoring choices.
- Record information over the duration of an event, such as a connection. To invoke it, click the left mouse button on a database object, and select **Monitor events** from a pop-up menu.

For DB2 Universal Database Enterprise – Extended Edition identify database nodes that are down, run user exits, or display alerts.

## Miscellaneous Troubleshooting Tools

There are standalone tools provided for troubleshooting. You can find these tools in the bin, misc, and adm subdirectories of the sqllib directory. In UNIX-based systems, the sqllib directory is a subdirectory of the instance owner's home directory.

These tools provide syntax help, which you can access by typing the command and following it with 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

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* for details).

You must run this tool on the DB2 server where the database resides. 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.)

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 fil `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 *must* 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 Pak level, that is currently installed.

**db2look** Lets you mimic a production database on your test database. Use the `-m` 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 `-e` option to generate the DDL for one or more tables from the database catalogs.

For introductory information on **db2look**, type `db2look ?`. For detailed information, type `db2look -h`.

For information on using system catalog statistics, see the *Administration Guide*.

**db2recr** Re-creates indexes that were marked as inconsistent during a database restart or use of the **db2dart** tool.

**db2resdb** Required by DB2 for OS/2 to restore a Version 1 database image to a target drive, and then migrate it to the current level.

**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. There are several parameters associated with this command. Refer to the *Command Reference* manual for additional information.

**db2tbst** Provides a textual description for a table space state.

**db2untag** 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.

Use this tool if a DROP TABLESPACE command does not work. Typically, the SQL0294N message is received.

**Attention:** Use this tool in consultation with DB2 Customer Service, 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.

## 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 **sqlmmon()** API. (You must have SYSADM, SYSCTRL, or SYSMAINT 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. Create its definition 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 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**. Once 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, refer to the *System Monitor Guide and Reference*. For a scenario of how to use them from the Control Center, refer to the *Administration Guide, Design and Implementation*.

## Troubleshooting Tips

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


- "Problems Migrating to Version 6"
- "Server Failures" on page 18
- "SQL1403N received when authenticating usernames or groups" on page 19
- "Administration Difficulties" on page 20
- "Backing Up and Restoring Data" on page 21
- "Difficulties Importing Data" on page 26
- "Problems Using Commands, Utilities, or the Command Line Processor" on page 26
- "Tables or Rows Locked" on page 27
- "Improving Database Performance" on page 29.


For additional troubleshooting information on this topic, see:


- "Chapter 7. Troubleshooting DB2 Universal Database Enterprise - Extended Edition" on page 77 for information on administering DB2 Enterprise - Extended Edition systems
- "Troubleshooting Tips for Host Connections to DB2 Universal Database" on page 58 for information on connecting to DB2 servers from host systems


**Important:** This section represents a small sampling of the information available from DB2 Customer Service. For a complete and up-to-date source of DB2 information, use the DB2 Product and Service Technical Library at http://www.software.ibm.com/data/db2/library/.

### Problems Migrating to Version 6

If you experience problems migrating to Version 6, ensure that:

[ ] You followed the instructions in the *Quick Beginnings* guide.

[ ] You checked the list of incompatibilities from previous versions. (See the *Administration Guide*.)

[ ] The database is consistent.

To verify that databases are consistent so that they can be migrated, use the **db2ckmig** command. (Refer to 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 **db2 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.

## 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 file" on page 120.

[ ] *DB2COMM* 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 37.

[ ] There is enough log space. (If not, the SQL0984C message is typically received.)

To increase log space:

- Increase the number of primary and 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/or deactivate the database.

You must ensure that the file system or disk space that contains the logs is big enough to hold all of the logs. To calculate the minimum required space for initializing the logs, use the following formula:

```
(logprimary + logsecondary) * logfilsiz * 4096
```

[ ] You are using the latest version of DB2, with appropriate fix packs installed. For information, see "Updating DB2 Products" on page 9.

[ ] The index files 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 13 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:

- There is a DIA3700C error in the db2diag.log file.
- The server goes down when particular data is accessed. 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 Service.

For DB2 Universal Database Enterprise - Extended Edition for the Solaris Operating Environment and DB2 Universal Database Enterprise - Extended Edition for Windows NT, see also "Database Manager Will Not Start or Hangs" on page 79.

## SQL1403N received when authenticating usernames or groups

**Symptom**

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

**Possible Cause**

The authorization restrictions on the Windows NT Master 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, refer to 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**, then **Programs**, then **Administrative Tools**, and then **User Manager**.

  To install DB2, you must have administrator authority on the local machine with a user name that is eight characters or less (note that you cannot use the user name "Administrator"). 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/group of the db2sysc process attempting to read a file must be the same as the id/group of the DB2 instance.

[ ] For UNIX-based environments other than AIX, you update and rebuild the kernel configuration parameters. (If not, the SQL1016N or SQL1018N messages will typically occur when you try to create an instance or use the command line processor.) Refer to 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 refer to the *Data Movement Utilities Guide and Reference* and use the Backup Database SmartGuide 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 13 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 Service.

### 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 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. 
   - If starting a new table space roll forward, put the table spaces into ROLLFORWARD PENDING state by restoring them.
   - If 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 error message SQL2419N is received when taking an online backup to disk.

**Possible Cause**

Although the disk was not full, the backup file was greater than the 2 GB 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 ADSM

**Symptom**

NetBIOS connections from DB2 clients are lost when backing up using ADSTAR Distributed Storage Manager (ADSM).

**Possible Cause**

As part of ADSM'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 all connections to the database made through NetBIOS to be dropped.

**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 ADSM 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. Refer to the *Administration Guide*.

[ ] 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 rollforward must proceed until at least the point where the online backup ended. Otherwise, the database is not accessible.)

### RESTORE using an API fails with SQL0902C return code 22

**Symptom**

On AIX, the error message SQL0902C 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**

Use the `ipcrm` AIX command to terminate all back-end processes and then drop the existing database. 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 ROLLFOWARD 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 FixPak, 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 error message SQL1277N 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` for additional information.

**Action**

Remove untag the containers in use and then perform the redirected restore using the Control Center. If you prefer to use the command line processor, you must use the `SET TABLESPACE CONTAINERS` API and then re-issue 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.

## Data Not Loading

### SQL3508N returned during a LOAD operation

**Symptom**

The error message SQL3508N 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.
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 will 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, refer to 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, Windows NT, and Version 3.2 of AIX have a 2 GB file size limit.

(Refer to "Problems with LOAD and with Splitting and Loading Data" on page 82 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 18.)

[ ] 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.

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

## Tables or Rows Locked

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, refer to 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 trace of each deadlock as it occurs.
- 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 15 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 122).

In the case of deadlocks, create an event monitor for deadlocks to gather more information. 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 *dlchktime* 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 ANSII and SQL 92 standard CS, and is the DB2 default.

Examine explain 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 flag on.

The DB2_RR_TO_RS flag 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 flag if you require ANSII and SQL standard CS,

For information on locking, configuration parameters, and explain information, refer to 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 SmartGuide 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 SmartGuide, click with the right mouse button on the database's icon 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 slow, you should ensure that the database statistics are up to date. The **runstats** command updates catalog statistics, thereby improving the potential for query performance. It is accessible from the command line or from the Control Center. Rebind all static applications after running it.

- The Index SmartGuide (or SQL Advise Facility) can be used with a specific query to create an index that may improve performance. Refer to 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.
  - **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 had not been captured at bind time.
  - **dynexpln** converts a dynamic statement into a static statement and is then interpreted 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 textual 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 could 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. A difference among Explain tools is that the Explain information in Visual Explainis provided in a graphical format. Besides this, however, the level of detail provided in either 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**, refer to 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 29). 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 28.
- 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 168.

For more information on performance:

- Search for the keyword "performance" in the DB2 Product and Service Technical Library at http://www.software.ibm.com/data/db2/library/.
- Refer to 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 Service 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
- db2diag.log and any DB2 trap, dump, or core files (see "First Failure Data Capture" on page 117)
- For UNIX systems, any system core files (see "Using the Core File" on page 164)
- Output from **db2dart** if you suspect inconsistency in the data (see "Miscellaneous Troubleshooting Tools" on page 13).
- 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, refer to 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 8, which discusses how to use the online help available for clients
- "Chapter 4. Troubleshooting Host Communications" on page 51, 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 communication protocol (for example, TCP/IP, APPC, 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 (and drive on Intel systems) 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, refer to 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.
- 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.
- The db2cli.ini and odbc.ini files are ASCII files that store values for DB2 CLI configuration. For more information, refer to the *CLI Guide and Reference*.

For details on client connectivity, refer to 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 its object from the Control Center and select **Setup Communications** from the pop-up menu.

**Note:** The communication 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, refer to the *Quick Beginnings* guide for your platform.

## Troubleshooting Tips

This section describes how to deal with some frequently encountered problems faced by users when trying to connect clients to DB2 servers. It addresses the following topics:


- "Determining the Scope of a Client Problem" on page 36
- "Initial Connection after Installation Fails" on page 38
- "Clients Suddenly Experiencing Problems" on page 38
- "User Name Not Valid on Windows 95 and Windows 98" on page 39
- "TCP/IP Problems" on page 40
- "IPX/SPX Problems" on page 43
- "NetBIOS Problems" on page 46
- "Problems When Using Named Pipes" on page 48
- "APPC Problems" on page 49.


For related information, see:


- "Chapter 2. Troubleshooting the DB2 Universal Database Server" on page 11 if you suspect the problem is with the DB2 server
- "Chapter 4. Troubleshooting Host Communications" on page 51 for troubleshooting host communications
- "Chapter 5. Troubleshooting Applications" on page 61 for troubleshooting applications running on clients, including ODBC applications.


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

**Important:** This section represents a small sampling of the information available from DB2 Customer Service. For a complete and up-to-date source of DB2 information, use the DB2 Product and Service Technical Library at http://www.software.ibm.com/data/db2/library/.

## 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. That is, 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. That is, try to establish a remote connection on the server, using the protocol, matching the clients'. Then catalog a database on the server with a node directory entry. Ensure that the node directory entry references the server machine. Attempt a connection from the server to itself.

   **Note:** For information on cataloging databases, refer to the *Administration Guide, Design and 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 communication protocol or 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 117 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:

Step 1. Set the DIAGLEVEL to 4 on the server:

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

Step 2. Disconnect all applications connected to the server:

```
db2 force application all
```

Step 3. Restart the server:

```
db2stop
db2start
```

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

For each protocol specified in *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 13. The DB2 Process Model" on page 111.)

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 41
- "SQL5043N received on server for IPX/SPX" on page 45

- "SQL5043N received on server for NetBIOS" on page 47

Step 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 install logs for unattended installations on Intel platforms:

- For OS/2, Windows 95, Windows 98, and Windows NT, specify error log and history files with the /l1 and /l2 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 is 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 there recent changes made to the client that could be affecting it? (For example, is another product or Fix Pak installed?)

[ ] Are resource limits, on the client, exceeded (for example, memory)?

**Are multiple clients experiencing difficulties?**

[ ] Is the LAN available? For example, can you use the **ping** command to servers, or issue the **net use** command to drives or devices on the network?

[ ] 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 37 for details.

[ ] Are resource limits, on the server, exceeded? (for example, memory)?

## 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 and 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 allows 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, 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, refer to the *DB2 for Windows NT Quick Beginnings* guide.

## TCP/IP Problems

This section outlines some common troubleshooting tips, 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. (Refer to 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 37, for more information.

[ ] The TCP/IP stack is functional on both the client and the server.

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 *svcename* field in the node directory maps to the same port number as the *svcename* in the server's database manager configuration.

  Try cataloging the node by specifying the *svcename* 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. Refer to 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 *svcename* configuration parameter defined is not correct.)
- The TCP/IP services file is not correct. (For example, the *svcename* configuration parameter in the database manager configuration is not defined in the file.)

**Action**

View the db2diag.log file 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 **nettune** command (for details, type `man nettune`).
- 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 World Wide 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, refer to 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 sqllib directory.

### SQL30081N on OS/2

**Symptom**

For OS/2 , the SQL30081N message is received with the function ″t_open″ 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 where you boot from.

[ ] 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 and Win-OS/2 environment, and the DLLs in the OS/2 NETWARE directory are OS/2 DLLs. In some cases, Windows and OS/2 NetWare DLLs have the same name, but DOS and Windows cannot load or run OS/2 DLLs.

### 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 where you boot from. 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 Win-OS/2 client

**Symptom**

The SQL1109N message is received when trying to connect from a Win-OS2 client.

**Possible Cause**

There are two versions of the NWCALLS.DLL and 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 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 NWCALLS.DLL.

### 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.
- Database manager configuration is not correct. (For example, the *fileserver*, *objectname*, or *ipx_socket* values 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 will 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 change database or node directories, you must perform **db2 terminate** 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 47, 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 server's database manager configuration parameter *nname*.

  If these three names are not the same, you must recatalog the node entry at the client.

[ ] 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 DB2's 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.

### 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 check the DB2 Product and Service Technical Library at http://www.software.ibm.com/data/db2/library/ for some suggested fixes.

### 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 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 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?

Refer to 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 Service 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
- db2diag.log and any trap or dump files, for both the client and server. See "First Failure Data Capture" on page 117, for more information.
- Output from the following commands:
  - LIST DB DIRECTORY
  - LIST NODE DIRECTORY

  - GET DBM CONFIG
  - **db2level**.
- ODBC/CLI traces for ODBC applications (refer to the *CLI Guide and Reference* manual).

Your DB2 Customer Service 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, refer to the following documents:
  - *DB2 Connect Enterprise Edition for OS/2 and Windows NT 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 49.
- 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 40.

This chapter provides the following troubleshooting sections:

- "Troubleshooting Host Connections"
- "Troubleshooting Tips for DB2 Connect" on page 53
- "Troubleshooting Tips for Host Connections to DB2 Universal Database" on page 58.

**Important:** These sections represent a small sampling of the information available from DB2 Customer Service. For a complete and up-to-date source of DB2 information, use the DB2 Product and Service Technical Library at http://www.software.ibm.com/data/db2/library/.

## Troubleshooting Host Connections

When solving problems with host communications, you must first isolate the communications layer to see if it is working. Use the following list of questions as a starting point.

**Can you establish a session with a host?**

[ ] Can you establish LU sessions with DB2 servers?

[ ] Are the communication products installed and operational? (An SQL30081N message indicates that lower level communication is failing.)

[ ] Have there been any changes made to your communication network that have invalidated the SNA profiles or the DB2 directory entries?

**Were the host and workstation communications configured correctly?**

[ ] If SNA is used and an initial connection is not made, are all of the LU (logical unit) and PU (physical unit) names complete and correct?

[ ] For APPC communication, is there a link and a session established? (See "APPC Problems" on page 49.)

[ ] For TCP/IP communication, are the host name and server name correct? (See "TCP/IP Problems" on page 40.)

[ ] For a DB2 Connect Enterprise Edition gateway, does the host configuration identify the application requester to the gateway? The host database management system has system catalog entries that will define the requestor in terms of location, network protocol, and security.

[ ] For a DB2 Connect Enterprise Edition gateway, does the configuration define the client population to the server and the server host to the client.

[ ] For DB2 Connect, does the client workstation configuration define the name of the workstation and the communication protocol? (The registry value *DB2COMM* should include APPC, TCPIP, or both, depending on which protocols you are using.)

Administration of host databases can be done using the Control Center interface. Check the help that is part of the Control Center for information on the requirements for this type of administration and for possible problems you may encounter.

If you determine that a link and session are established and you still experience problems, see the other chapters in this guide to troubleshoot your DB2 clients and servers.

## Troubleshooting Tips for DB2 Connect

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


- "Initial Questions"
- "Problems Making an Initial Connection after Installation" on page 54
- "Problems Encountered after Initial Connection" on page 54
- "DB2 Connect Connection Using SNA Does Not Work" on page 55
- "Changing the Number of Connections" on page 56
- "Authentication Problems" on page 56.


### Initial Questions

The client/server environment involves multiple software, hardware, and communications products, including DB2 Connect. When a problem occurs in host communications, troubleshooting is best approached by asking the following questions:

**Can you establish a session with a host?**

[ ] See "Troubleshooting Host Connections" on page 52.

[ ] For SNA sessions, see "DB2 Connect Connection Using SNA Does Not Work" on page 55.

**If you can communicate with the host, is the problem with DB2 Connect?**

[ ] Can you get a connection from a standalone client to your DB2 Connect Enterprise Edition gateway? If not, see Chapter 3. Troubleshooting on the Client.

## Problems Making an Initial Connection after Installation

Initial connection means the first attempt to connect to a host server after installation. If the initial connection fails, review the following questions to ensure that the DB2 Connect installation steps were followed:

**Did the installation processing complete successfully?**

[ ] Did you configure APPC on DB2 Connect clients as described in *Installation and Configuration Supplement*?

[ ] Were all the prerequisite software products installed?

For example, if it did not already exist on the operating system, was the communications software completely installed without any error conditions?

[ ] Was DB2 Connect Enterprise Edition installed to enable support for remote clients?

**Was a DB2 instance created?**

[ ] To check that an instance was created, get a list of instances with the **db2ilist** command. If there are no instances, or if you need to make another one, use the **db2icrt** command. For more information, refer to *DB2 Connect Enterprise Edition for OS/2 and Windows NT Quick Beginnings*.

**Were the host and workstation communications configured correctly?**

[ ] See "Troubleshooting Host Connections" on page 52 and, for SNA sessions, "DB2 Connect Connection Using SNA Does Not Work" on page 55.

**Do you have the level of authority required by the host database management system to use the host database?**

[ ] Did you sign on with the correct authority to access tables? Refer to the *DB2 Connect Enterprise Edition for OS/2 and Windows NT Quick Beginnings*, for more information.

## 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 52 and, for SNA sessions, "DB2 Connect Connection Using SNA Does Not Work".

[ ] If you suspect a problem with your communication protocol, see "Chapter 3. Troubleshooting on the Client" on page 33.

**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, refer to the *Message Reference*.

[ ] Check the db2diag.log file on the server, particularly for SQLCA information. For more details, refer to "First Failure Data Capture" on page 117.

## DB2 Connect Connection Using SNA Does Not Work

Use the following checklist as a starting point to narrow down the scope of an SNA problem when using DB2 Connect.

**Is the SNA session active?**

[ ] Is the SNA software started on the gateway?

[ ] Is the SNA software started on the host? (For example, is the DDF facility started on DB2 for MVS/ESA?)

**Is SNA information correctly configured?**

[ ] Is APPC configured properly? See "APPC Problems" on page 49, for more information.

[ ] If DB2 Connect Enterprise Edition gateways are used, check the

communications configuration (for example, SNA Server for AIX, Communication Server (formerly Communications Manager) for OS/2, IBM Communication Server for Windows NT, or Microsoft SNA Server for Windows NT).

[ ] Is the host correctly configured? For example, is VTAM configured on MVS and VM systems? For more information, refer to *DB2 Connect Enterprise Edition for OS/2 and Windows NT Quick Beginnings*.

**Is the DB2 Connect information correct?**

[ ] Are the node, system database, and DCS directories configured?

[ ] Are the DB2INSTANCE and PATH variables correct?

**Is the logon information correct?** (If not, the SQL30082N message is usually received.)

[ ] Has the DRDA AS password expired?

[ ] Have related CDB tables been updated correctly on DB2 for MVS/ESA?

## 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 will only allow 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, refer to the *Administration Guide, Design and 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, note that 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, refer to the MVS Server worksheet in the *DB2 Connect Enterprise Edition for OS/2 and Windows NT Quick Beginnings*.

### SQL1402N and SQL30082N when connecting to DB2 for MVS/ESA

**Symptom**

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

**Possible Cause**

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 Will Not Connect

**Symptom**

The message SQL30073N 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 because either:

- 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, refer to the section on configuring national language support in the *DB2 Connect Enterprise Edition for OS/2 and Windows NT 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.

## 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 Service 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 117.
- ODBC/CLI traces for ODBC applications. (Refer to 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 15. Traces" on page 135.

# Chapter 5. Troubleshooting Applications

Applications are the means by which most users will 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:

- For updated information on developing applications for DB2, refer to the Application Development with DB2 Universal Database website at http://www.software.ibm.com/data/db2/udb/ad. This web site includes the latest information on developing applications for DB2 Universal Database using the embedded SQL, Java, or CLI/ODBC APIs.
- *Application Development Guide* and http://www.software.ibm.com/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).
- *Administrative API Reference* provides information on the use of application programming interfaces (APIs) to execute DB2 administrative functions.
- *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.
- *Administration Guide, Design and Implementation* provides information on authorization and *Administration Guide, Performance* has information on tuning application performance.
- *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 how to deal with some frequently encountered problems faced 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 Service. DB2 Customer Service

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 63
- "Running Applications" on page 64
- "Using CLI/ODBC/JDBC Applications" on page 65
- "Using SQLJ Applications" on page 69
- "Using the db2ocat Tool" on page 69.


In addition, see "Developing Applications in DB2 Enterprise - Extended Edition" on page 87.

**Important:** For a complete and up-to-date source of DB2 information, use the DB2 Product and Service Technical Library at http://www.software.ibm.com/data/db2/library/.

## 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 27.

[ ] Your application is bound before you issue any static SQL statements. For information on the `sqlabndx` bind API, refer to 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 SQL code or the SQL state element of the SQLCA data structure following each SQL statement. If it encounters a negative SQL code, it should take appropriate actions. For example, it should capture the SQL code and the SQLCA structure for later analysis.

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

- `util.c` for C
- `checkerr.cbl` for COBOL
- `util.f` for FORTRAN

Negative SQL codes that result in a terminated connection are logged in the DB2 diagnostic log (db2diag.log). 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. Refer to 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, Design and Implementation*.

## Compiling and Linking Applications

At compile time, ensure that:

[ ] Your compiler environment is correct and all include files and environment variables are current. Refer to your compiler's documentation for details.

[ ] Your compiler options are correct. Refer to 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, refer to the *Application Building Guide* book for your platform.

## 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.software.ibm.com/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 27, for more information.

[ ] If you are using CLI/ODBC/JDBC applications, review the considerations in the section "Using CLI/ODBC/JDBC Applications" on page 65.

### 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 6 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 27 and refer to 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_all ";db2 UPDATE MONITOR SWITCHES USING STATEMENT ON"
```

The semi-colon(;) and `db2_all` 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_all ";db2 GET SNAPSHOT FOR ALL APPLICATIONS ON database"
```

For more information on **db2_all**, see "Commands for DB2 Enterprise - Extended Edition" on page 169.

## 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 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 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. Select the **Properties** push button. The Database Properties window appears.
3. Select the **Settings** push button. The CLI/ODBC Settings window appears.
4. Select the **Advanced** push button. 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.

[ ] 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 could 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 could be seen 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 give even though:

1. The column “column1” exists, and shows up in the Control Center

   or

   the SQL statement: `SELECT tabname, colname FROM SYSCAT.COLUMNS WHERE tabname='table2'` returns the appropriate information about the “column1” column.
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"` would also return 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 upper case.

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 quotes 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 one 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 parameter "Error popup" 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, refer to "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 on the `Properties` push button. The Database Properties window opens.
4. Click on the `Settings` push button. The CLI/ODBC Settings window opens.
5. Click on the `Advanced` push button.
6. Click on the `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, refer to the *Application Development Guide* and the DB2/Lotus Approach web page at http://www.software.ibm.com/data/db2/db2lotus. Select 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.

### Using db2ocat for Performance Improvement

**Step 1: Create an ODBC optimized catalog(s) on the host**

- To create an ODBC optimized catalog, db2ocat utility should be run by a DBA on a Windows 95, Windows 98, or Windows NT workstation. db2ocat provides a SmartGuide that takes the DBA 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.
- This step needs to be repeated for each application that would benefit from an ODBC optimized catalog.

**Step 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, DB2 CLI, and JDBC applications.

- The parameter, CLISCHEMA, is set in the DB2CLI.INI file in the subdirectory where DB2 products are installed (typically \sqllib). 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.
- The CLISCHEMA parameter can be set 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.

**Step 3: Keep the ODBC optimized catalog(s) current.**

- ODBC optimized catalogs, created with the db2ocat utility, produce a separate copy of the data extracted form 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.
- db2ocat provides a manual procedure, requiring that when catalog changes occur, a Database Administrator start the db2ocat utility on a Windows 95, Windows 98, or Windows NT workstation and press the **Refresh** button 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.
- This DataPropagator option allows Database Administrators 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 PAI calls during the "information gathering" stage after the database is first connected.
2. CLISHEMA keywords in db2cli.ini file should be set to an ODBC catalog schema. The CLISCHEMA option indicates an alternative schema, tables, and index set to be searched instead of the SYSIBM (or SYSTEM, QSYS2) schemas when the DB2 CLI and ODBC Catalog function calls are issued to obtain catalog information.
3. If you specify CLISCHEMA='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 separate 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. Refer to the compiler documentation for information on using the supplied debugger.

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

## Still Have Problems?

If you still have problems with your applications:

- Refer to the DB2 Product and Service Technical Library at http://www.software.ibm.com/data/db2/library/ for more information.
- Contact the vendor of your application or compiler if you suspect a defect or limitation.
- Contact DB2 Customer Service 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 117.

- An CLI/ODBC/JDBC trace for CLI/ODBC/JDBC applications. (Refer to the *CLI Guide and Reference* manual.)
- Output from **db2level**.

# 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 services 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 periodically scan 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 using commands entered 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, refer to the *DB2 Query Patroller Administration Guide* manual.

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

## Still Have Problems?

If you still have problems with DB2 Query Patroller:

- Refer to the DB2 Product and Service Technical Library at http://www.software.ibm.com/data/db2/library/ for more information.
- Contact the vendor of your application or compiler if you suspect a defect or limitation.
- Contact DB2 Customer Service 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. See "First Failure Data Capture" on page 117, 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, see "Commands for DB2 Enterprise - Extended Edition" on page 169 for some useful UNIX-based commands. You may also refer to the following publications

- *DB2 Enterprise - Extended Edition for UNIX Quick Beginnings*
- *DB2 Enterprise - Extended Edition for Windows NT Quick Beginnings*
- The *Administration Guide, Design and Implementation* 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

For a detailed explanation of DB2 Enterprise - Extended Edition, refer to the *Administration Guide, Design and Implementation*.

## Troubleshooting Tips

This section describes how to deal with some frequently encountered problems faced by users working with DB2 Enterprise - Extended Edition. It addresses the following topics:


- "DB2 Universal Database Enterprise - Extended Edition Will Not Install" on page 78
- "Severe Error Occurs on System" on page 78
- "Determining Failing Node" on page 79
- "Database Manager Will Not Start or Hangs" on page 79
- "Cannot Create Database" on page 81
- "Commands Cannot Be Issued" on page 82
- "Problems with LOAD and with Splitting and Loading Data" on page 82
- "Problems with Indexing Data" on page 85
- "Problems during Recovery" on page 86

- "Developing Applications in DB2 Enterprise - Extended Edition" on page 87
.

**Important:** This section represents a small sampling of the information available from DB2 Customer Service. For a complete and up-to-date source of DB2 information, refer to the DB2 Product and Service Technical Library at http://www.software.ibm.com/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.

## 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 node 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 the node by running **db2 RESTART DATABASE** for all databases on that node. Refer to the *Command Reference* for more information.

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

  – 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 **db2 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 file for additional details. For information, see "Determining Failing Node" and "Interpreting the db2diag.log file" on page 120.

**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 file, if they were not able to obtain an NFS lock on the db2diag.log file file.

## Determining Failing Node

To find out which node failed, check the db2diag.log file 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.)

For more information on the db2diag.log file file and SQLCA structures, see "Interpreting the db2diag.log file" on page 120.

## Database Manager Will Not Start or Hangs

### db2start fails

**Symptom**

The **db2start** command fails.

**Action**

Ensure that:

**[ ]** The product was successfully installed.

[ ] Your user ID has SYSADM, SYSCTRL, or SYSMAINT authority. Refer to the *Administration Guide, Design and Implementation* for details.

[ ] 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 on 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.)

[ ] If you are using the high-performance switch (HPS) on RS/6000 SP systems, ensure that it is up on all nodes. See the description for **spmon** in "Commands for DB2 Enterprise - Extended Edition" on page 169.

[ ] The db2nodes.cfg file, located in the `sqllib` 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.

[ ] FCM communication is properly enabled. You must create an entry for each DB2 logical port in the /etc/services file. Refer to the *DB2 Enterprise - Extended Edition for Windows NT Quick Beginnings* or the *DB2 Enterprise - Extended Edition for UNIX Quick Beginnings* for details.

[ ] 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 `sqllib` 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.

[ ] You have checked the contents of the `db2diag.log` file for possible reasons for the `db2start` failure.

### db2start does not respond

**Symptom**

The **db2start** command invokes no response.

**Action**

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.

### 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. We recommend that you 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.

## 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. Refer to the *Administration Guide, Design and Implementation* 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. Following 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.

## 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, refer to the *Data Movement Utilities Guide and Reference*.

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

Refer to 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, refer to 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. Refer to 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. Refer to 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, refer to the *Administration Guide, Design and Implementation*.

## Problems during Recovery

### Connection 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 file of the coordinator node to see if crash recovery is completed. Entries in the db2diag.log file will indicate when recovery is started, and when it is finished.

The connection will finish once recovery is complete on all nodes.

### 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 file of the coordinator node to see if all nodes are started.

Resolve the indoubt transactions as soon as possible. For information, refer to the discussion on heuristics in the *Administration Guide, Design and Implementation*.

### 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. See "Interpreting an SQLCA Structure" on page 122 and "Appendix C. SQL Communications (SQLCA)" on page 225, for more information.

  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, refer to the *Administration Guide, Design and Implementation*.
- 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. See "DB2 Enterprise - Extended Edition Usage of the SQLCA" on page 229 for more information about using the SQLCA in a multinode environment.

## Still Have Problems?

If you still have problems with your DB2 Enterprise - Extended Edition system, contact DB2 Customer Service 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.
- db2diag.log file, and any trap or dump files, for each node. See "First Failure Data Capture" on page 117, for more information.
- Any SYSLOG files. See "Using the System Error Log (syslog)" on page 161, for more information.
- Any core files. See "Using the Core File" on page 164, for more information.
- The contents of the `db2nodes.cfg` file.

- Output from **db2level**.

# 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 90
- "Data Access Token" on page 90
- "Recovery Data Links Manager Environment" on page 91
- "Backup Data Links Manager Environment" on page 93
- "Export Operation" on page 94
- "Import Operation" on page 94
- "Load Considerations" on page 95
- "DB2 Data Links Manager for Windows NT" on page 96.


## 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 Pak 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, refer to *DB2 Data Links Manager for Windows NT Quick Beginnings* and *DB2 Data Links Manager for AIX 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 refer to 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:<br>• Restore the database(s) and roll forward to end of logs.<br>• 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:<br>• Restore the database(s) and roll forward to end of logs.<br>• Restore database DLFM_DB and roll forward to end of log. |
| Recover from a disk crash on the DLFM Server. | DB2 applicatons 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 applicatons interacting with the DLFM can hang. | The affected DB2 Data Links Managers shuld 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:<br>• Restore the DLFS file system.<br>On the DB2 Database Server:<br>• Peform 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 for Windows NT Quick Beginnings* and *DB2 Data Links Manager for AIX 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 for Windows NT Quick Beginnings* and *DB2 Data Links Manager for AIX 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 inserted as DATALINK values.
- Define the prefix name (or names) to the DLFM on the file servers. For more information, refer to the description on registering databases with the

DLFM in the *DB2 Data Links Manager for Windows NT Quick Beginnings* and *DB2 Data Links Manager for AIX Quick Beginnings*.

- If intending to refer to 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:

- Refer to the DB2 Product and Service Technical Library at http://www.software.ibm.com/data/db2/library/ for more information.
- See "Appendix A. How the DB2 Library Is Structured" on page 203 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 Service 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 117.
  - 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.

IBM Replication provides the following facilities when troubleshooting problems:

- Error messages and SQL states for the DB2 Control Center, the Capture program, and the Apply program
- The Apply program trial table, log file and trace file
- The Capture program trace table, log file and trace file
- The IBM Replication control tables and files

When testing or troubleshooting replication problems, we recommend that you 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
- During the running of the Capture and Apply programs

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.

## DpropR

DpropR 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

input_LSN is a 12 byte string that represents the internal (6–byte) hexadecimal value with leading zeros.

The log header control file sqlogctl.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 UDB Version 5.2

For more information see ″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
```

## DpropR Enhancements

DpropR 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 still can be copied but the target column will be defined as DECIMAL.

DpropR 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 base 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 base 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**

Synchronization anomalies may occur, because the LOB column value 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:

- Refer to the DB2 Product and Service Technical Library at http://www.software.ibm.com/data/db2/library/ for more information.
- Contact the vendor of your application or compiler if you suspect a defect or limitation.
- Contact DB2 Customer Service 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 117.
  - Output from **db2level**.

# Chapter 10. 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 will have a model office which is a special member of the test satellites in a group. Typically, you will 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 97 for information on troubleshooting replication problems.

## Still Have Problems?

If you still have problems with the satellite environment:

- Refer to the DB2 Product and Service Technical Library at http://www.software.ibm.com/data/db2/library/ for more information.
- Contact the vendor of your application or compiler if you suspect a defect or limitation.
- Contact DB2 Customer Service 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 117.
  - Output from **db2level**.

# Chapter 11. 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, there is nothing special you have to do 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, refer to the *Installation and Configuration Supplement*.

## Still Have Problems?

If you still have problems with NetQuestion:

- Refer to the DB2 Product and Service Technical Library at http://www.software.ibm.com/data/db2/library/ for more information.
- See "Appendix A. How the DB2 Library Is Structured" on page 203 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 Service 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 117.
  - Output from **db2level**.

# Part 2. Advanced DB2 Troubleshooting

This part addresses the following topics:


- "Chapter 12. Logs and Traces Roadmap" on page 107
- "Chapter 13. The DB2 Process Model" on page 111
- "Chapter 14. Logged Information" on page 117
- "Chapter 15. Traces" on page 135
- "Chapter 16. Diagnostic Tools for UNIX-Based Systems" on page 161
- "Chapter 17. Diagnostic Tools for OS/2" on page 171
- "Chapter 18. Diagnostic Tools for Windows 32-bit Operating Systems" on page 183
- "Chapter 19. GUI and Monitoring Tools" on page 187
- "Chapter 20. Table and Table Space Recovery" on page 195.

# Chapter 12. Logs and Traces Roadmap

When you contact DB2 Customer Service, 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 | • For servers.<br>• Analyzes database objects for architectural integrity. | "Miscellaneous Troubleshooting Tools" on page 13. |

| DB2 Logging | | | |
|---|---|---|---|
| Facility | Platform | Notes | Refer to... |
| db2diag.log file | All | • For clients and servers.<br>• Logs errors and (depending on the diagnostic level) warnings and informational messages. | "First Failure Data Capture" on page 117. |
| Alert file | All | • For clients and servers.<br>• Contains severe DB2 errors. | "Alerts" on page 132. |
| Dump files | All | • For clients and servers.<br>• Provides additional information for some errors. | "Dump Files" on page 128. |
| Trap files | All | • For clients and servers.<br>• Files that are created by DB2 in the event of a segmentation violation or trap. | "Trap Files" on page 129. |

| | | | |
|---|---|---|---|
| DB2 message files | All | • For servers.<br>• Logs activities during operations such as bind, load, import and export. | "Using Message Files" on page 131. |
| DB2 Performance monitor | Windows NT, Windows 95, OS/2 | • For clients and servers.<br>• Logs deadlocks and other performance information. | "The Database System Monitor" on page 15, and the *System Monitor Guide and Reference*. |
| Administrative tools log | Windows NT, Windows 95, OS/2 | • For clients and servers using the Control Center and other DB2 administrative tools.<br>• Logs information for the administrative tools. | "Using the Administrative Tools Log" on page 130. |

| **Traces** | | | |
|---|---|---|---|
| Facility | Platform | Notes | Refer to... |
| **db2trc** | All | • For clients and servers.<br>• Traces DB2 events. | "Using the DB2 Trace Facility (db2trc)" on page 135. |
| Administrative tools trace | Windows NT, OS/2 | • For clients and servers using the Control Center and other DB2 administrative tools.<br>• Traces problems with the administrative tools. | "Administration Tools Trace" on page 142. |
| **db2drdat** | All | • Traces DRDA requesters and servers. | "Taking a DRDA Trace (db2drdat)" on page 143 and *DB2 Connect User's Guide*. |
| **ddcstrc** | All | • Traces DB2 Connect Enterprise Edition gateways. | *DB2 Connect User's Guide*. |
| CLI trace | All | • Traces CLI, JDBC, or ODBC applications. | Refer to the *CLI Guide and Reference* manual. |
| SNA trace | All | • Traces communications via SNA. | "Taking an SNA Trace" on page 145. |

| Process Identification | | | |
|---|---|---|---|
| Facility | Platform | Notes | Refer to... |
| **ps** | UNIX | • For clients and servers.<br>• Shows current processes (along with process names for AIX). | "Chapter 13. The DB2 Process Model" on page 111 and "Viewing Process Status Using the ps Command" on page 166. |
| **pstat** | OS/2 | • For clients and servers.<br>• Shows processes and their threads. | "Chapter 13. The DB2 Process Model" on page 111 and "Using the pstat Command" on page 179. |
| Windows NT Performance monitor | Windows NT | • For clients and servers.<br>• Shows processes and their threads. | "Chapter 13. The DB2 Process Model" on page 111 and "Using Administrative Tools Provided by Windows NT" on page 184. |

| System Logging | | | |
|---|---|---|---|
| Facility | Platform | Notes | Refer to... |
| System error log (syslog) | UNIX and OS/2 | • Logs alert conditions.<br>• For use by DB2 Customer Service. | "Using the System Error Log (syslog)" on page 161 and "Using the OS/2 Logging Facility" on page 172. |
| Core file | UNIX | • For UNIX-based systems.<br>• Logs severe errors.<br>• For use by DB2 Customer Service. | "Using the Core File" on page 164. |
| UNIX commands | UNIX | • Let you monitor and troubleshoot UNIX-based systems. | "Other Commands" on page 167. |
| Event logger | Windows NT | • Logs system errors, including severe DB2 errors for Windows NT systems.<br>• Accessible to remote system administrators. | "Diagnostic Tools for Windows NT" on page 183. |

| | | | |
|---|---|---|---|
| SPM/2 | OS/2 | • Monitors the performance of OS/2. | "Chapter 17. Diagnostic Tools for OS/2" on page 171. |
| Administrative tools | Windows NT Version 4.0 | • Provide various administrative tools to diagnose and monitor Windows NT systems. | "Diagnostic Tools for Windows NT" on page 183. |
| Dr. Watson | All Windows systems | • Logs system errors during a General Protection Fault (GPF). | "Diagnostic Tools for Windows NT" on page 183. |
| Native OS/2 logging | OS/2 | • Logs error information on OS/2 programs. | "Using the OS/2 Logging Facility" on page 172. |

| **Alerts** | | | |
|---|---|---|---|
| Facility | Platform | Notes | Refer to... |
| DB2 alert file | All | • For clients and servers.<br>• Contains severe errors. | "Alerts" on page 132. |
| SNA alerts | OS/2 | • Sends information over an SNA network in the event of a severe error.<br>• Can be generated by DB2 servers. | "Using the FFST for OS/2 Utility" on page 174. |
| SNMP subagent | Windows NT, AIX, OS/2 | • Sends information over an SNMP network in the event of a severe error.<br>• Can be set up for DB2 servers. | "SNMP Alerts" on page 132. |

# Chapter 13. 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.

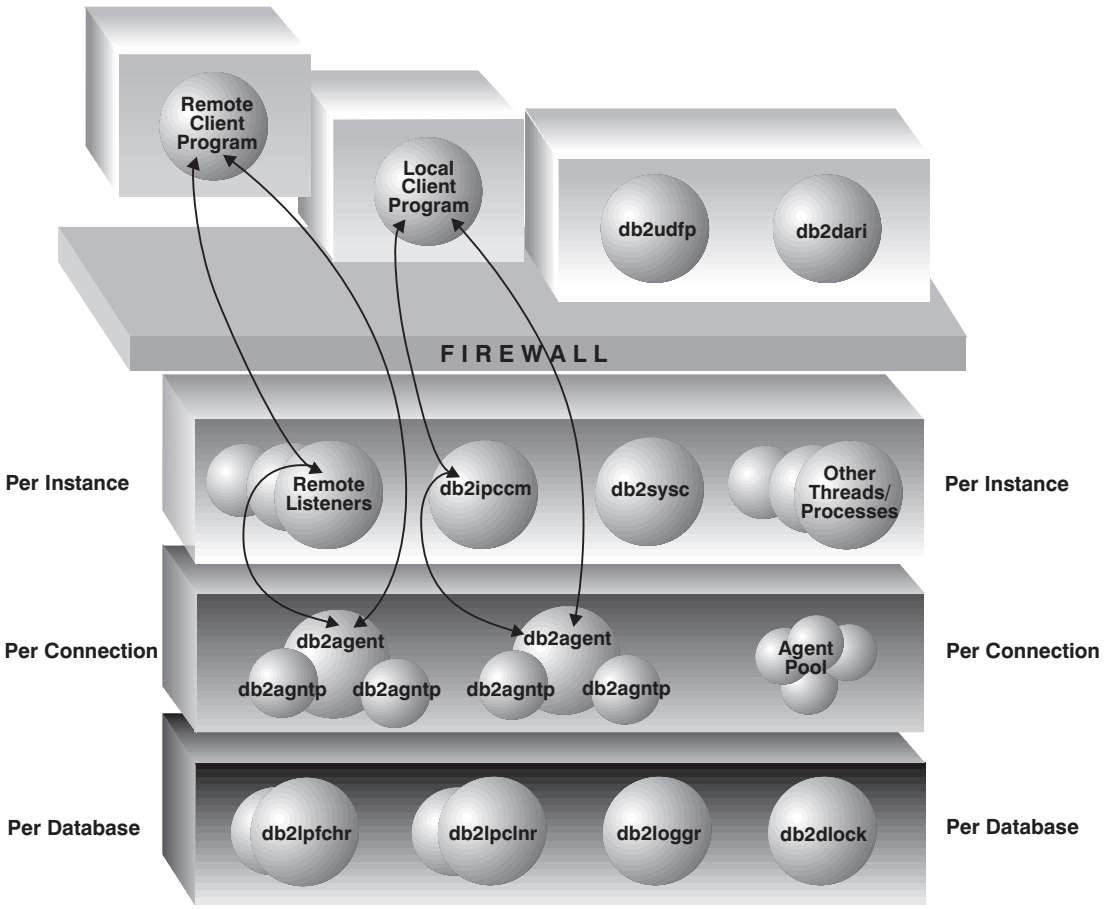



*Figure 1. Process Model for DB2 Systems*

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
- **db2tcpcm**, 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 file at the server. For more information, see "Using the db2diag.log file to diagnose server communication problems" on page 37.

**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
- **db2pclnr**, 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 file to diagnose if any parts of the database manager are failing. See “First Failure Data Capture” on page 117, 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 166.
- On OS/2: the **pstat** command. For more information, see “Using the pstat Command” on page 179.
- On Windows NT Version 4.0: the Task Manager. For more information, see “Using Administrative Tools Provided by Windows NT” on page 184.
- 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 186.

# Chapter 14. 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 file**
When an error occurs, the db2diag.log file is updated with information about the error. For more details, see "Interpreting the db2diag.log file" on page 120. For information on the DIAGLEVEL setting, which determines how much information is logged in the file, see "Setting Configuration Parameters Affecting FFDC" on page 118.

**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 132 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 Service. See "Dump Files" on page 128, 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 129, 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, refer to 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 sattelite 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/sqllib/db2dump, where $HOME is the home directory of the instance owner.

To set these configuration parameters, use the UPDATE DATABASE MANAGER CONFIGURATION command from the command line processor. (Alternatively, use the Problem Determination folder for the OS/2 platform; see "Appendix B. Using the Problem Determination Tools Folder" on page 219 for information.)

**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, Performance*, the *Command Reference*, and the *Administrative API Reference* manuals.

## Interpreting the db2diag.log file

The db2diag.log file 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 file includes the following:

- The location reporting an error. Application identifiers allow you to match up entries pertaining to an application on the db2diag.log files 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 file:

- 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 file **grows continuously**. When it gets too large, back it up and then erase the file. A new db2diag.log file is generated automatically the next time it is required by the system.

### Interpreting db2diag.log file Entry Headers

The following example shows the header information for a sample db2diag.log file entry.

```
1997-03-16-11.53.18.001160 1    Instance:payroll 2   Node:000 3
PID:44829(db2agent (SAMPLE)) 4     Appid:*LOCAL.payroll.970317140834 5
lock_manager 6          sqlplrq 7    Probe:111 8   Database:SAMPLE 9
DIA9999E 10 An internal return code occurred. Report the following:
"0xFFFFE10E". 11
```

**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 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 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.

**6** The DB2 component that is writing the message. For messages written by user applications using the `db2AdminMsgWrite` API, the component will read “User Application”.

**7** 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 ″sqlplrq″ 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

**8** Identification of the internal error that was reported.

**9** The database on which the error occurred.

**10** Diagnostic message indicating that an internal error occurred.

**11** Hexadecimal representation of an internal return code. For more information, see "Interpreting Hexadecimal Codes" on page 123.

### Interpreting an SQLCA Structure

For severe errors, an SQLCA structure is dumped into the db2diag.log file. For details on each of the SQLCA fields, see "Appendix C. SQL Communications (SQLCA)" on page 225.

The following diagram provides an example of a db2diag.log file 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          sqlrerlg    Probe:17   Database:SAMPLE
DIA9999E An internal return code occurred. Report the following : "0xFFFFE101".
Data Title :SQLCA pid(14358) 1
sqlcaid : SQLCA    sqlcabc: 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 file 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 D. DB2 Internal Return Codes" on page 231 for a list of return codes.

## Some Examples for Interpreting the db2diag.log file

The following examples illustrate how you can use the db2diag.log file 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 file on the client machine.

Assume that the db2diag.log file on the client machine contains no entries. Look at the db2diag.log file on the DB2 server machine to see if an error has occurred on the server.

Assume that the db2diag.log file 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 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 Service.

**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 file on the client machine.

Assume the db2diag.log file on the client machine contains no entries. Look at the db2diag.log file on the DB2 server machine to see if an error has occurred on the server.

Assume the db2diag.log file 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 123.

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 D. DB2 Internal Return Codes" on page 231, 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 file 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 refer to 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 file 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 file 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 D. DB2 Internal Return Codes" on page 231, we see that the disk is full.

**2** The sqlpgif1 function repeatedly encounters the error. This function is a logging function (because the fourth character is "p". See "Interpreting the db2diag.log file" on page 120, 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

   `(logprimary * (logfilsiz * 4096) + 8192`

   to

   `((logprimary + logsecond) * (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 (refer to the *Administration Guide, Performance*).

When the db2diag.log file 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 file and the DB2 Trace" on page 140.

## 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 Service representatives. They should be collected and sent along with the db2diag.log file.

When a dump file is created or appended to, an entry is made in the db2diag.log file indicating the time and the type of data written. These db2diag.log file 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          sqlrerlg    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 file.

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 Service. 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 Service 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 DBTTRACE_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 Service may require more information than is in this log and request a DB2 tools trace. See "Administration Tools Trace" on page 142, 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, refer to 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 file) and to the alert log (db2alert.log). Like the db2diag.log file, 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 file, 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 161.
- For OS/2, FFST/2 and SNA alerts. See "Using the FFST for OS/2 Utility" on page 174.
- For Windows NT, the event log. See "Diagnostic Tools for Windows NT" on page 183.

### 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**.

To turn on DPI internal debugging and tracing, use the -d parameter with the **db2snmpd** command. DPI tracing will be turned on at level 2, enabling the display of a hexadecimal dump of incoming and outgoing DPI packets.

# Chapter 15. 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 Service.

## 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 Service to take a trace if the db2diag.log file 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:

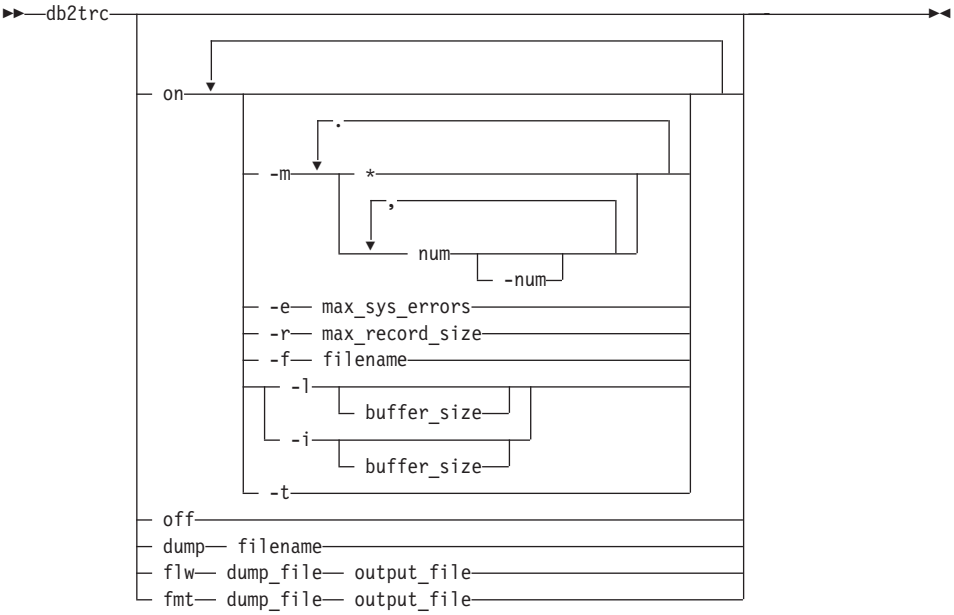
- Typically you will use the trace facility only when directed by DB2 Customer Service or by your technical support representative. Trace information should be sent to IBM for analysis **only when it is explicitly requested**.
- Only use this facility when error conditions are predictable and reproducible.
- 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 137, 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. In most cases, only the dump file should be sent to DB2 Customer Service.

## 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 Service:

**-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 Service 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 139 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 file 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 file on the client machine. For more information, see "Interpreting the db2diag.log file" on page 120.

Assume that the db2diag.log file on the client machine contains no entries. Look at the db2diag.log file on the DB2 server machine to see if an error has occurred on the server.

Assume that the db2diag.log file 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 D. DB2 Internal Return Codes" on page 231, 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 file 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 = SQLO_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" on page 142
- "Taking a Graphical Interface Trace" on page 143
- "Taking a DB2 Connect Trace (ddcstrc)" on page 143
- "Taking a DRDA Trace (db2drdat)" on page 143

- "Taking an SNA Trace" on page 145
- "Using the DB2 CLI/ODBC/JDBC Trace Facility" on page 145.

## Administration Tools Trace

Occasionally, DB2 Customer Service 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. See "Taking a Graphical Interface Trace" on page 143, for more information.

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 130, 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 DBTTRACE=Y
   ```
3. You can specify where the trace file will be stored with the following command:
   ```
   set DBTTRACE_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 DBTTRACE_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 DBTTRACE_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 Graphical Interface Trace

If you have problems with graphical interfaces on the Windows 95, Windows NT, or OS/2 operating systems, you can enable an ICLUI trace. This trace helps you diagnose problems with the user interface (for example, with windows or dialog boxes not appearing or not refreshing).

To invoke an ICLUI trace for the Control Center:

1. Ensure the Control Center is closed.
2. Set ICLUI tracing on to collect standard error output:

   ```
   set ICLUI_TRACE=ON
   set ICLUI_TRACETO=STDERR
   ```

3. Invoke the Control Center with the following command:

   ```
   db2cc 2 > db2cc.out
   ```

   In this example, the ICLUI trace will be stored in a file called db2cc.out. (The ″2″ is a filepointer to standard log (STDLOG), which is being redirected to db2cc.out.)
4. Perform the procedure that caused the problem with the user interface.
5. Capture the db2cc.out file.

## Taking a DB2 Connect Trace (ddcstrc)

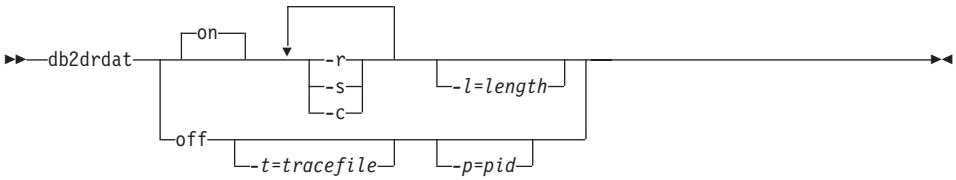
The DB2 Connect trace (ddcstrc) is useful for diagnosing problems with the DB2 Connect product. For more information, refer to the *DB2 Connect User's Guide*.

## 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 SYSMAINT 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 135.
- 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 132,

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.software.ibm.com/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 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. The CLI/ODBC/JDBC configuration keywords used by the trace facility are:

- "TRACE" on page 154
- "TRACEFILENAME" on page 156
- "TRACEPATHNAME" on page 158
- "TRACEFLUSH" on page 157
- "TRACECOMM" on page 155

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)
4. TraceFlush=(0 or 1) - optional
5. TraceComm=(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 154 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 and "TRACEFLUSH" on page 157 for more information.

To have information about each network request included in the trace file, set **TRACECOMM** to 1. See "TRACECOMM" on page 155 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** 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 (eg. 65397.0, 65397.1, 65397.2 etc.). 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.
- 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 153 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 ″<——″.

- 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 ouput (<——) 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,
          cbValueMax=4, pcbValue=&20714e04 )
64      ---> Time elapsed - +5.069000E-003 seconds

65  SQLBindCol( )
66      <--- SQL_SUCCESS   Time elapsed - +2.660000E-003 seconds

67  SQLBindCol( hStmt=1:1, iCol=2, fCType=SQL_C_CHAR, rgbValue=&20714e88,
          cbValueMax=21, pcbValue=&20714e54 )
68      ---> Time elapsed - +2.492000E-003 seconds

69  SQLBindCol( )
70      <--- SQL_SUCCESS   Time elapsed - +2.795000E-003 seconds

71  SQLBindCol( hStmt=1:1, iCol=3, fCType=SQL_C_CHAR, rgbValue=&20714ee8,
          cbValueMax=21, pcbValue=&20714eb4 )
72      ---> Time elapsed - +2.490000E-003 seconds

73  SQLBindCol( )
74      <--- SQL_SUCCESS   Time elapsed - +2.749000E-003 seconds

75  SQLFetch( hStmt=1:1 )
76      ---> Time elapsed - +2.660000E-003 seconds

77  SQLFetch( )
78      <--- SQL_SUCCESS   Time elapsed - +9.200000E-003 seconds
79  ( iCol=1, fCType=SQL_C_LONG, rgbValue=10, pcbValue=4 )
80  ( iCol=2, fCType=SQL_C_CHAR, rgbValue="-3                  ",
        pcbValue=20 )
81  ( iCol=3, fCType=SQL_C_CHAR, rgbValue="1997-05-23", pcbValue=10 )

82  SQLFetch( hStmt=1:1 )
83      ---> Time elapsed - +4.942000E-003 seconds

84  SQLFetch( )
85      <--- SQL_SUCCESS   Time elapsed - +7.860000E-003 seconds
86  ( iCol=1, fCType=SQL_C_LONG, rgbValue=10, pcbValue=4 )
87  ( iCol=2, fCType=SQL_C_CHAR, rgbValue="-3                  ",
        pcbValue=20 )
88  ( iCol=3, fCType=SQL_C_CHAR, rgbValue="1997-05-23", pcbValue=10 )

89  SQLFetch( hStmt=1:1 )
90      ---> Time elapsed - +4.872000E-003 seconds

91  SQLFetch( )
92      <--- SQL_SUCCESS   Time elapsed - +7.669000E-003 seconds
93  ( iCol=1, fCType=SQL_C_LONG, rgbValue=10, pcbValue=4 )
94  ( iCol=2, fCType=SQL_C_CHAR, rgbValue="-3                  ",
        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 158 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 (eg. 65397.0, 65397.1, 65397.2 etc.).

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:

- "TRACE"
- "TRACEFILENAME" on page 156
- "TRACEPATHNAME" on page 158
- "TRACEFLUSH" on page 157
- "TRACECOMM" on page 155

**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 156, "TRACEFLUSH" on page 157, "TRACEPATHNAME" on page 158

**Usage Notes:**

When this option is on (1), CLI/ODBC/JDBC trace records are appended to the file indicated by the configuration parameter or to files in the subdirectory indicated by the 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 154, "TRACEFILENAME" on page 156, "TRACEPATHNAME" on page 158, "TRACEFLUSH" on page 157

**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/JDBCtrace 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 154, "TRACEFLUSH" on page 157, "TRACEPATHNAME" on page 158

**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.

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 154, "TRACEFILENAME" on page 156, "TRACEPATHNAME" on page 158

**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/JDBCoption 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 154, "TRACEFILENAME" on page 156, "TRACEFLUSH" on page 157

**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.

# Chapter 16. 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 Service 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 164, 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 166, 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 167.

## 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 file. The exception is when a severe error occurs where the DB2 database manager was not able to write to the db2diag.log file. 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 file and in the alert log (db2alert.log). For more information, see "First Failure Data Capture" on page 117.

### 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 sqlobeep 2  reports: 3
Jun 18 15:02:53  4  bluj  5  DB2[46827 6 ]: extra symptom string provided: 7  RIDS/sqlesysc_
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 Service.

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 128 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 *) pcAdress[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 13. The DB2 Process Model” on page 111.)

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 ″svtdbm″ use: `ps -fu svtdbm`

The following sample shows typical output from this command. (Note that there would be additional processes shown for DB2 Enterprise - Extended Edition.)

```
svtdbm 1   5112 2 27894 3   0 10:48:17 4      -  0:00 db2ipccm 5
svtdbm   6135 27894   0 10:48:17      -  0:00 db2gds
svtdbm  14329 27894   0 10:48:17      -  0:00 db2resyn
svtdbm  15356 27894   0 10:48:17      -  0:00 db2snacm 6
svtdbm  18682 27894   0 10:48:17      -  0:00 db2tcpcm 6
svtdbm  27894 31989   0 10:48:17      -  0:00 db2sysc 7
svtdbm  33275 27894   0 10:48:17      -  0:00 db2tcpim 6
svtdbm  39939 18682   7 10:48:19      -  0:00 db2agent
svtdbm  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 13. The DB2 Process Model” on page 111 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. Refer to 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 `sqllib` directory and the *Administration Guide, Design and 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>0.0.0.0.0.0
```

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 129, for more information.

# Chapter 17. 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 Service 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 172, 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 174 and "Taking an SNA Trace" on page 145.

**pstat** The **pstat** command lists status information on which processes are running. For more information, see "Using the pstat Command" on page 179.

**SPM/2** The System Performance Monitor for OS/2 (SPM/2) is a separate product that helps you monitor memory, memory utilization, CPU times, threads and processes, and input/output.

**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 Service.

**Kernel debugging for advanced problem diagnosis**
In severe situations, DB2 Customer Service 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 174.

### 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, refer to 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, refer to 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 172.
- 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 informaiton 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:

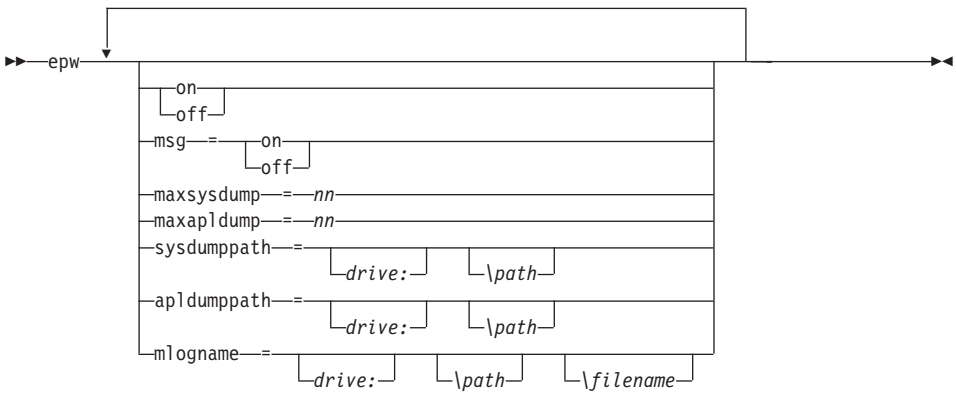
- Enable and disable FFST for OS/2
- Enable and disable the FFST for OS/2 message display
- Select the maximum number of dump data sets of a given type (either system dumps or application dumps) that may exist at one time in the current subdirectory for that type of dump
- Select the subdirectory in 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 172.
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 `OS2SYS`*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 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 `OS2APLxx.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 172. 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 Service to determine exactly which portion of DB2 caused the error to be logged.

  This string is generated in the following format: `PIDS/`*`ccccccccc`* `LVLS/lll RIDS/`*`ssssssss`* `PCSS/`*`nnnn`* `PRCS/`*`pppppppp`* `PRCS/`*`scscscsc`* `MS/`*`mnmnmnmn`*
- 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 **epwrout** command.

Its syntax is as follows:

```
epwrout [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=EPWROUT -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.

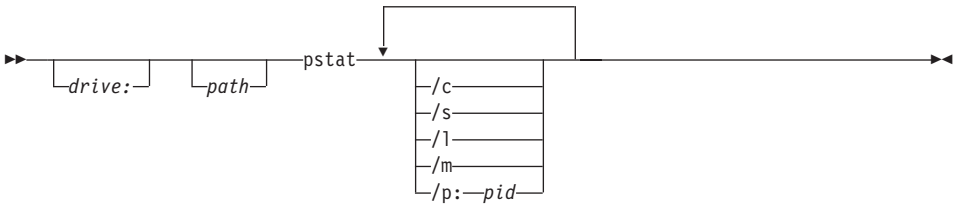
This command helps you determine:

- 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 13. The DB2 Process Model” on page 111.

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
          Parent
Process   Process  Session  Process                    Thread
  ID        ID       ID      Name                       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 18. 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 Service 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 186.

**DB2 traces**
DB2 traces provide some information on reproducible problems. For more information, see "Chapter 15. Traces" on page 135.

**ODBC/CLI traces**
CLI traces help to identify problems in CLI and ODBC applications. For more information, refer to 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 186.

**ODBC/CLI traces**
CLI traces help to identify problems in CLI and ODBC applications. For more information, refer to 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 132.

## 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".

- 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 refer to the db2diag.log file. Therefore it is recommended that you save both files before you contact DB2 Customer Service.

## 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 15.
- The db2diag.log file identifies serious problems. For more information, see "First Failure Data Capture" on page 117.

# Chapter 19. 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 190
- "Troubleshooting Page" on page 191
- "Stored Procedure Builder" on page 192
- "DB2 Index Advisor" on page 192


## 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, refer to 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, refer to the *System Monitor Guide and Reference.* For a listing of snapshot request types and logical data groupings, refer to "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()`, refer to "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, refer to "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, refer to ″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, refer to ″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, refer to "Control Center Installation and Configuration." You can find this document in the file `x:\sqllib\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 `sqllib` 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 141 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 messsage 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 messsage 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.

## 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. Refer to 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 `db2advis`.
- 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.

Refer to the "SQL Advise Facility" section in the "SQL Explain Facility" chapter of the *Administration Guide, Performance* manual for more information on how to use the Index Advisor.

# Chapter 20. Table and Table Space Recovery

This chapter discusses how to troubleshoot table and table space problems. It addresses the following topics:


- "Dropped Table PIT Recovery"
- "Table Space OFFLINE" on page 196
- "Table Space Usability" on page 197.


## Dropped Table PIT Recovery

This feature allows you to recover dropped tables to the table space. Prepare every table for Point-in-Time 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 ore 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.

# Part 3. Appendixes

# Appendix A. How the DB2 Library Is Structured

The DB2 Universal Database library consists of SmartGuides, online help, books and sample programs in HTML format. This section describes the information that is provided, and how to access it.

To access product information online, you can use the Information Center. You can view task information, DB2 books, troubleshooting information, sample programs, and DB2 information on the Web. See "Accessing Information with the Information Center" on page 214 for details.

## Completing Tasks with SmartGuides

SmartGuides help you complete some administration tasks by taking you through each task one step at a time. SmartGuides are available through the Control Center and the Client Configuration Assistant. The following table lists the SmartGuides.

**Note:** Create Database, Index, and Configure Multisite Update SmartGuide are available for the partitioned database environment.

| SmartGuide | Helps You to... | How to Access... |
|---|---|---|
| *Add Database* | Catalog a database on a client workstation. | From the Client Configuration Assistant, click **Add**. |
| *Back up Database* | Determine, create, and schedule a backup plan. | From the Control Center, click with the right mouse button on the database you want to back up and select **Backup->Database using SmartGuide**. |
| *Configure Multisite Update SmartGuide* | Perform a multi-site update, a distributed transaction, or a two-phase commit. | From the Control Center, click with the right mouse button on the **Database** icon and select **Multisite Update**. |
| *Create Database* | Create a database, and perform some basic configuration tasks. | From the Control Center, click with the right mouse button on the **Databases** icon and select **Create->Database using SmartGuide**. |

| SmartGuide | Helps You to... | How to Access... |
| --- | --- | --- |
| *Create Table* | Select basic data types, and create a primary key for the table. | From the Control Center, click with the right mouse button on the **Tables** icon and select **Create->Table using SmartGuide**. |
| *Create Table Space* | Create a new table space. | From the Control Center, click with the right mouse button on the **Table spaces** icon and select **Create->Table space using SmartGuide**. |
| *Index* | Advise which indexes to create and drop for all your queries. | From the Control Center, click with the right mouse button on the **Index** icon and select **Create->Index using SmartGuide**. |
| *Performance Configuration* | Tune the performance of a database by updating configuration parameters to match your business requirements. | From the Control Center, click with the right mouse button on the database you want to tune and select **Configure using SmartGuide**. |
| *Restore Database* | Recover a database after a failure. It helps you understand which backup to use, and which logs to replay. | From the Control Center, click with the right mouse button on the database you want to restore and select **Restore->Database using SmartGuide**. |

## Accessing Online Help

Online help is available with all DB2 components. The following table describes the various types of help. You can also access DB2 information through the Information Center. For information see "Accessing Information with the Information Center" on page 214.

| Type of Help | Contents | How to Access... |
| --- | --- | --- |
| *Command Help* | Explains the syntax of commands in the command line processor. | From the command line processor in interactive mode, enter:<br>? *command*<br>where *command* is a keyword or the entire command.<br>For example, `? catalog` displays help for all the CATALOG commands, while `? catalog database` displays help for the CATALOG DATABASE command. |

| Type of Help | Contents | How to Access... |
| --- | --- | --- |
| *Control Center Help*<br><br>*Client Configuration Assistant Help*<br><br>*Event Analyzer Help*<br><br>*Command Center Help* | Explains the tasks you can perform in a window or notebook. The help includes prerequisite information you need to know, and describes how to use the window or notebook controls. | From a window or notebook, click the **Help** push button or press the F1 key. |
| *Message Help* | Describes the cause of a message, and any action you should take. | From the command line processor in interactive mode, enter:<br><br>`?` *XXXnnnnn*<br><br>where *XXXnnnnn* is a valid message identifier.<br><br>For example, `? SQL30081` displays help about the SQL30081 message.<br><br>To view message help one screen at a time, enter:<br><br>`?` *XXXnnnnn* `| more`<br><br>To save message help in a file, enter:<br><br>`?` *XXXnnnnn* `>` *filename.ext*<br><br>where *filename.ext* is the file where you want to save the message help. |
| *SQL Help* | Explains the syntax of SQL statements. | From the command line processor in interactive mode, enter:<br><br>`help` *statement*<br><br>where *statement* is an SQL statement.<br><br>For example, **help** SELECT displays help about the SELECT statement.<br>**Note:** SQL help is not available on UNIX-based platforms. |
| *SQLSTATE Help* | Explains SQL states and class codes. | From the command line processor in interactive mode, enter:<br><br>**?** *sqlstate* or **?** *class-code*<br><br>where *sqlstate* is a valid five-digit SQL state and *class-code* is the first two digits of the SQL state.<br><br>For example, `? 08003` displays help for the 08003 SQL state, while `? 08` displays help for the 08 class code. |

## DB2 Information – Hardcopy and Online

The table in this section lists the DB2 books. They are divided into two groups:

**Cross-platform books**
These books contain the common DB2 information for all platforms.

**Platform-specific books**
These books are for DB2 on a specific platform. For example, there are separate *Quick Beginnings* books for DB2 on OS/2, on Windows NT, and on the UNIX-based platforms.

**Cross-platform sample programs in HTML**
These samples are the HTML version of the sample programs that are installed with the SDK. They are for informational purposes and do not replace the actual programs.

Most books are available in HTML and PostScript format, or you can choose to order a hardcopy from IBM. The exceptions are noted in the table.

On OS/2 and Windows platforms, HTML documentation files can be installed under the doc\html subdirectory. Depending on the language of your system, some files may be in that language, and the remainder are in English.

On UNIX platforms, you can install multiple language versions of the HTML documentation files under the doc/%L/html subdirectories. Any documentation that is not available in a national language is shown in English.

You can obtain DB2 books and access information in a variety of different ways:

**View** See "Viewing Online Information" on page 213.

**Search** See "Searching Online Information" on page 216.

**Print** See "Printing the PostScript Books" on page 216.

**Order** See "Ordering the Printed Books" on page 217.

| Name | Description | Form Number<br>File Name for Online Book | HTML Directory |
|---|---|---|---|
| | **Cross-Platform Books** | | |

| Name | Description | Form Number<br><br>File Name for Online Book | HTML Directory |
|---|---|---|---|
| *Administration Guide* | *Administration Guide, Design and Implementation* contains information required to design, implement, and maintain a database. It also describes database access using the Control Center(whether local or in a client/server environment), auditing, database recovery, distributed database support, and high availability.<br><br>*Administration Guide, Performance* contains information that focuses on the database environment, such as application performance evaluation and tuning.<br><br>You can order both volumes of the *Administration Guide* in the English language in North America using the form number SBOF-8922. | Volume 1<br>SC09-2839<br>db2d1x60<br><br>Volume 2<br>SC09-2840<br>db2d2x60 | db2d0 |
| *Administrative API Reference* | Describes the DB2 application programming interfaces (APIs) and data structures you can use to manage your databases. Explains how to call APIs from your applications. | SC09-2841<br><br>db2b0x60 | db2b0 |
| *Application Building Guide* | 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.<br><br>This book combines the *Building Applications* books for the OS/2, Windows, and UNIX-based environments. | SC09-2842<br><br>db2axx60 | db2ax |
| *APPC, CPI-C and SNA Sense Codes* | Provides general information about APPC, CPI-C, and SNA sense codes that you may encounter when using DB2 Universal Database products.<br>**Note:** Available in HTML format only. | No form number<br><br>db2apx60 | db2ap |

| Name | Description | Form Number<br><br>File Name for Online Book | HTML Directory |
|---|---|---|---|
| *Application Development Guide* | Explains how to develop applications that access DB2 databases using embedded SQL or JDBC, how to write stored procedures, user-defined types, user-defined functions, and how to use triggers. It also discusses programming techniques and performance considerations.<br><br>This book was formerly known as the *Embedded SQL Programming Guide*. | SC09-2845<br><br>db2a0x60 | db2a0 |
| *CLI Guide and Reference* | Explains how to develop applications that access DB2 databases using the DB2 Call Level Interface, a callable SQL interface that is compatible with the Microsoft ODBC specification. | SC09-2843<br><br>db2l0x60 | db2l0 |
| *Command Reference* | Explains how to use the command line processor, and describes the DB2 commands you can use to manage your database. | SC09-2844<br><br>db2n0x60 | db2n0 |
| *Data Movement Utilities Guide and Reference* | Explains how to use the Load, Import, Export, Autoloader, and Data Propogation utilities to work with the data in the database. | SC09-2858<br><br>db2dmx60 | db2dm |
| *DB2 Connect Personal Edition Quick Beginnings* | Provides planning, installing, and configuring information for DB2 Connect Personal Edition. | GC09-2830<br><br>db2c1x60 | db2c1 |
| *DB2 Connect User's Guide* | Provides concepts, programming and general usage information about the DB2 Connect products. | SC09-2838<br><br>db2c0x60 | db2c0 |
| *Connectivity Supplement* | 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, and on how to use DRDA application servers with DB2 Connect application requesters.<br>**Note:** Available in HTML and PostScript formats only. | No form number<br><br>db2h1x60 | db2h1 |
| *Glossary* | Provides a comprehensive list of all DB2 terms and definitions.<br>**Note:** Available in HTML format only. | No form number<br><br>db2t0x50 | db2t0 |

| Name | Description | Form Number<br><br>File Name for Online Book | HTML Directory |
|---|---|---|---|
| *Installation and Configuration Supplement* | Guides you through the planning, installation, and set up of platform-specific DB2 clients. This supplement contains information on binding, setting up client and server communications, DB2 GUI tools, DRDA AS, distributed installation, and the configuration of distributed requests and access methods to heterogeneous data sources. | GC09-2857<br><br>db2iyx60 | db2iy |
| *Message Reference* | Lists messages and codes issued by DB2, and describes the actions you should take. | GC09-2846<br><br>db2m0x60 | db2m0 |
| *Replication Guide and Reference* | Provides planning, configuration, administration, and usage information for the IBM Replication tools supplied with DB2. | SC26-9642<br><br>db2e0x60 | db2e0 |
| *SQL Getting Started* | Introduces SQL concepts, and provides examples for many constructs and tasks. | SC09-2856<br><br>db2y0x60 | db2y0 |
| *SQL Reference*, Volume 1 and Volume 2 | Describes SQL syntax, semantics, and the rules of the language. Also includes information about release-to-release incompatibilities, product limits, and catalog views.<br><br>You can order both volumes of the *SQL Reference* in the English language in North America with the form number SBOF-8923. | SBOF-8923<br><br>Volume 1<br>db2s1x60<br><br>Volume 2<br>db2s2x60 | db2s0 |
| *System Monitor Guide and Reference* | Describes how to collect different kinds of information about databases and the database manager. Explains how to use the information to understand database activity, improve performance, and determine the cause of problems. | SC09-2849<br><br>db2f0x60 | db2f0 |
| *Troubleshooting Guide* | Helps you determine the source of errors, recover from problems, and use diagnostic tools in consultation with DB2 Customer Service. | S10J-8169 | db2p0 |

| Name | Description | Form Number<br><br>File Name for Online Book | HTML Directory |
|---|---|---|---|
| *What's New* | Describes the new features, functions, and enhancements in DB2 Universal Database, Version 6.0, including information about Java-based tools. | SC09-2851<br><br>db2q0x60 | db2q0 |
| | **Platform-Specific Books** | | |
| *Administering Satellites Guide and Reference* | Provides planning, configuration, administration, and usage information for satellites. | GC09-2821<br><br>db2dsx60 | db2ds |
| *DB2 Personal Edition Quick Beginnings* | Provides planning, installation, migration, and configuration information for DB2 Universal Database Personal Edition on the OS/2, Windows 95, and Windows NT operating systems. | GC09-2831<br><br>db2i1x60 | db2i1 |
| *DB2 for OS/2 Quick Beginnings* | Provides planning, installation, migration, and configuration information for DB2 Universal Database on the OS/2 operating system. Also contains installing and setup information for many supported clients. | GC09-2834<br><br>db2i2x60 | db2i2 |
| *DB2 for UNIX Quick Beginnings* | Provides planning, installation, migration, and configuration information for DB2 Universal Database on UNIX-based platforms. Also contains installing and setup information for many supported clients. | GC09-2836<br><br>db2ixx60 | db2ix |
| *DB2 for Windows NT Quick Beginnings* | Provides planning, installation, migration, and configuration information for DB2 Universal Database on the Windows NT operating system. Also contains installing and setup information for many supported clients. | GC09-2835<br><br>db2i6x60 | db2i6 |
| *DB2 Enterprise - Extended Edition for UNIX Quick Beginnings* | Provides planning, installation, and configuration information for DB2 Enterprise - Extended Edition for UNIX. Also contains installing and setup information for many supported clients. | GC09-2832<br><br>db2v3x60 | db2v3 |

| Name | Description | Form Number<br><br>File Name for Online Book | HTML Directory |
|---|---|---|---|
| *DB2 Enterprise - Extended Edition for Windows NT Quick Beginnings* | Provides planning, installation, and configuration information for DB2 Enterprise - Extended Edition for Windows NT. Also contains installing and setup information for many supported clients. | GC09-2833<br><br>db2v6x60 | db2v6 |
| *DB2 Connect Enterprise Edition for OS/2 and Windows NT Quick Beginnings* | Provides planning, migration, installation, and configuration information for DB2 Connect Enterprise Edition on the OS/2 and Windows NT operating systems. Also contains installation and setup information for many supported clients.<br><br>This book was formerly part of the *DB2 Connect Enterprise Edition Quick Beginnings.* | GC09-2828<br><br>db2c6x60 | db2c6 |
| *DB2 Connect Enterprise Edition for UNIX Quick Beginnings* | Provides planning, migration, installation, configuration, and usage information for DB2 Connect Enterprise Edition in UNIX-based platforms. Also contains installation and setup information for many supported clients.<br><br>This book was formerly part of the *DB2 Connect Enterprise Edition Quick Beginnings.* | GC09-2829<br><br>db2cyx60 | db2cy |
| *DB2 Data Links Manager for AIX Quick Beginnings* | Provides planning, installation, configuration, and task information for DB2 Data Links Manager for AIX. | GC09-2837<br><br>db2z0x60 | db2z0 |
| *DB2 Data Links Manager for Windows NT Quick Beginnings* | Provides planning, installation, configuration, and task information for DB2 Data Links Manager for Windows NT. | GC09-2827<br><br>db2z6x60 | db2z6 |
| *DB2 Query Patroller Administration Guide* | Provides administration information on DB2 Query Patrol. | SC09-2859<br><br>db2dwx60 | db2dw |
| *DB2 Query Patroller Installation Guide* | Provides installation information on DB2 Query Patrol. | GC09-2860<br><br>db2iwx60 | db2iw |
| *DB2 Query Patroller User's Guide* | Describes how to use the tools and functions of the DB2 Query Patrol. | SC09-2861<br><br>db2wwx60 | db2ww |

| Name | Description | Form Number<br><br>File Name for Online Book | HTML Directory |
|---|---|---|---|
| **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 for informational purposes (not all samples are available in all languages). Only available when the SDK is installed.<br><br>See *Application Building Guide* for more information on the actual programs.<br>**Note:** Available in HTML format only. | No form number | db2hs/c<br>db2hs/cli<br>db2hs/clp<br>db2hs/cpp<br>db2hs/cobol<br>db2hs/cobol_mf<br>db2hs/fortran<br>db2hs/java<br>db2hs/rexx |

**Notes:**

1. The character in the sixth position of the file name indicates the language of a book. For example, the file name `db2d0e60` indicates that the *Administration Guide* is in English. The following letters are used in the file names to indicate the language of a book:

| Language | Identifier |
|---|---|
| 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. For late breaking information that could not be included in the DB2 books:
   - On UNIX-based platforms, see the Release.Notes file. This file is located in the DB2DIR/Readme/%L directory, where %L is the locale name and DB2DIR is:
     - /usr/lpp/db2_06_01 on AIX
     - /opt/IBMdb2/V6.1 on HP-UX, Solaris, SCO UnixWare 7, and Silicon Graphics IRIX
     - /usr/IBMdb2/V6.1 on Linux.
   - On other platforms, see the RELEASE.TXT file. This file is located in the directory where the product is installed.
   - Under Windows `Start` menu

## Viewing Online Information

The manuals included with this product are in Hypertext Markup Language (HTML) softcopy format. Softcopy format enables you to search or browse the information, and provides hypertext links to related information. It also makes it easier to share the library across your site.

You can 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 on all platforms other than SCO UnixWare 7:

- If you are running DB2 administration tools, use the Information Center. See "Accessing Information with the Information Center" on page 214 for details.
- Select the `Open Page` menu item of your Web browser. The page you open contains descriptions of and links to DB2 information:
  - On UNIX-based platforms, open the following page:

    ```
    file:/INSTHOME/sqllib/doc/%L/html/index.htm
    ```

    where *%L* is the locale name.
  - On other platforms, open the following page:

    ```
    sqllib\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 on the **DB2 Online Books** icon. Depending on the system you are using, the icon is in the main product folder or the Windows Start menu.

To view online books or sample programs on the SCO UnixWare 7:

- DB2 Universal Database for SCO UnixWare 7 uses the native SCOhelp utility to search the DB2 information. You can access SCOhelp by the following methods:
  - entering the "scohelp" command on the command line,
  - selecting the `Help` menu in the Control Panel of the CDE desktop or
  - selecting `Help` in the Root menu of the Panorama desktop

  For more information on SCOhelp, refer to the *Installation and Configuration Supplement*.

## 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.

Depending on your system, you can access the Information Center from the:

- Main product folder
- Toolbar in the Control Center
- Windows Start menu
- Help menu of the Control Center

The Information Center provides the following kinds of information. Click the appropriate tab to look at the information:

| | |
|---|---|
| **Tasks** | Lists tasks you can perform using DB2. |
| **Reference** | Lists DB2 reference information, such as keywords, commands, and APIs. |
| **Books** | Lists DB2 books. |
| **Troubleshooting** | Lists categories of error messages and their recovery actions. |
| **Sample Programs** | Lists sample programs that come with the DB2 Software Developer's Kit. If the Software Developer's Kit is not installed, this tab is not displayed. |
| **Web** | Lists DB2 information on the World Wide |

Web. To access this information, you must have a connection to the Web from your system.

When you select an item in one of the lists, the Information Center launches a viewer to display the information. The viewer might be the system help viewer, an editor, or a Web browser, depending on the kind of information you select.

The Information Center provides some search capabilities, so you can look for specific topics, and filter capabilities to limit the scope of your searches.

For a full text search, click the `Search` button of the Information Center follow the *Search DB2 Books* link in each HTML file.

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 by double-clicking its icon on the Windows or OS/2 desktop.

Refer to the release notes if you experience any other problems when searching the HTML information.

**Note:** Search function is not available in the Linux and Silicon Graphics environments.

## Setting Up a Document Server

By default, the DB2 information is installed on your local system. This means that each person who needs access to the DB2 information must install the same files. To have the DB2 information stored in a single location, use the following instructions:

1. Copy all files and subdirectories from \sqllib\doc\html on your local system to a Web server. Each book has its own subdirectory containing all the necessary HTML and GIF files that make up the book. Ensure that the directory structure remains the same.
2. Configure the Web server to look for the files in the new location. For information, see the NetQuestion Appendix in *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. Once you are able to view the book files, you should bookmark commonly viewed topics. Among those, 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 setting up a search, see the NetQuestion Appendix in *Installation and Configuration Supplement* book.

## Searching Online Information

To search for information in the HTML books, you can do the following:

- Click on **Search the DB2 Books** at the bottom of any page in the HTML books. Use the search form to find a specific topic. This function is not available in the Linux or Silicon Graphics IRIX environments.
- Click on **Index** at the bottom of any page in an HTML book. Use the index to find a specific topic in the book.
- Display the table of contents or index of the HTML book, and then use the find function of the Web browser to find a specific topic in the book.
- Use the bookmark function of the Web browser to quickly return to a specific topic.
- Use the search function of the Information Center to find specific topics. See "Accessing Information with the Information Center" on page 214 for details.

## Printing the PostScript Books

If you prefer to have printed copies of the manuals, you can decompress and print PostScript versions. For the file name of each book in the library, see the table in "DB2 Information – Hardcopy and Online" on page 206. Specify the full path name for the file you intend to print.

On OS/2 and Windows platforms:

1. Copy the compressed PostScript files to a hard drive on your system. The files have a file extension of .exe and are located in the x:\doc*language*\books\ps directory, where x: is the letter representing the CD-ROM drive and *language* is the two-character country code that represents your language (for example, EN for English).
2. Decompress the file that corresponds to the book that you want. Each compressed book is a self-extracting executable file. To decompress the

book, simply run it as you would run any other executable program. The result from this step is a printable PostScript file with a file extension of .ps.

3. Ensure that your default printer is a PostScript printer capable of printing Level 1 (or equivalent) files.
4. Enter the following command from a command line:

   ```
   print filename.ps
   ```

On UNIX-based platforms:

1. Mount the CD-ROM. Refer to your *Quick Beginnings* manual for the procedures to mount the CD-ROM.
2. Change to /cdrom/doc/%L/ps directory on the CD-ROM, where */cdrom* is the mount point of the CD-ROM and *%L* is the name of the desired locale. The manuals will be installed in the previously-mentioned directory with file names ending with .ps.Z.
3. Decompress and print the manual you require using the following command:
   - For AIX:

     ```
     zcat filename | qprt -P PSPrinter_queue
     ```

   - For HP-UX, Solaris, or SCO UnixWare 7:

     ```
     zcat filename | lp -d PSPrinter_queue
     ```

   - For Linux:

     ```
     zcat filename | lpr -P PSPrinter_queue
     ```

   - For Silicon Graphics IRIX:

     ```
     zcat < filename | lp -d PSPrinter_queue
     ```

   where *filename* is the full path name and extension of the compressed PostScript file and *PSprinter_queue* is the name of the PostScript printer queue.

   For example, to print the English version of *DB2 for UNIX Quick Beginnings* on AIX, you can use the following command:

   ```
   zcat /cdrom/doc/en/ps/db2ixe60.ps.Z | qprt -P ps1
   ```

## Ordering the Printed Books

You can order the printed DB2 manuals either as a set or individually. There are two sets of books available. The form number for the entire set of DB2 books is SBOF-8926-00. The form number for the books listed under the heading "Cross-Platform Books" is SBOF-8924-00.

**Note:** These form numbers only apply if you are ordering books that are printed in the English language in North America.

You can also order books individually by the form number listed in "DB2 Information – Hardcopy and Online" on page 206. To order printed versions, contact your IBM authorized dealer or marketing representative, or phone `1-800-879-2755` in the United States or `1-800-IBM-4YOU` in Canada.

# Appendix B. Using the Problem Determination Tools Folder

For OS/2 users, several tools are provided in the Problem Determination Tools folder (available from the DB2 folder). These tools let you perform a variety of problem determination tasks.

For Windows NT users, tools to perform a DB2 trace and to access DB2 information on the Internet are accessible from the **Start -> Programs -> DB2** menus.

**Note:** Several of these tools require that you have an Internet connection established before you use them.

## Accessing Internet Information for DB2 for OS/2

DB2 for OS/2 supplies an HTML page that provides links to DB2 information on the Internet. The page can be found in the file:

*x*:*path*\doc\html\db2itnet.htm

where:

- *x* is the disk drive on which DB2 is installed
- *path* is the name of the directory where DB2 is installed (the default is sqllib)

There is an icon to access this page in the DB2 folder. Or, you may wish to create a bookmark in your Web browser to access this page.

## Accessing Internet Information for DB2 for Windows NT

DB2 for Windows NT provides direct access to the latest DB2 information on the Internet. To access this information:

1. Establish your Internet connection.
2. Select the **Start -> Programs -> DB2 -> Problem Determination Tools** menu choice. The Problem Determination Tools folder appears.
3. Double-click on the **Support through Internet** icon.
4. Your Web browser will start, and show a page that provides access to DB2 information on the World Wide Web. (You may wish to create bookmarks of your favorite DB2 pages so that you can access them directly from your browser in the future.)

## Updating DB2 Products

Typically your IBM representative will advise you of any fix packs you need, but you should browse the Internet periodically to see if there are new fix packs available. It is good practice to run your system on the latest practical code level.

DB2 for OS/2 provides a tool to FTP to an IBM Internet site and download software fix packs for your products.

To receive fix packs using this tool:

1. Double-click on the **Problem Determination Tools** icon in the DB2 folder. The Problem Determination Tools folder appears.
2. Double-click on the **Receive Fixes** icon. The Internet address and remote directory for the FTP server are specified on the window that appears. You do not need to change these values.
3. There are different fix packs for the various national languages. Click on the language you want.
4. Click on the product for which you want to receive the Fix Pak.
5. A second panel appears showing the fix packs that are available for the product and language you have selected.
6. Ensure the **Local directory** is set for the location for the downloaded files.
7. Click on the Fix Pak you need. A list of all files associated with the Fix Pak appears. Choose the file you want to download. One of the files you can select is a readme file. It contains instructions for unpacking the files and descriptions of the problems fixed.
8. Click on **Get** to begin the transmission.

## Setting Diagnostic Levels

You can change the amount of diagnostic information generated by DB2 events, and specify where the information is stored on your workstation. For more information, see "First Failure Data Capture" on page 117.

To set diagnostic levels using this tool:

1. Double-click on the **Problem Determination Tools** icon in the DB2 folder. The Problem Determination Tools folder appears.
2. Double-click on the **Set Diagnostic Levels** icon.
3. For **Select a Diagnostic Level (DIAGLEVEL)**, click on the radio button that corresponds to the level of information you want recorded.
4. For **Diagnostic Path (DIAGPATH)** enter the fully-qualified path name of the location to store the diagnostic information.

5. Click on **OK** to set the level and path.
6. Click on **Exit** to close the panel.

## Collecting Diagnostic Information

DB2 for OS/2 provides automatic collection of the files required by DB2 Customer Service to diagnose DB2 problems.

You must have a problem record open with DB2 Customer Service to use this tool. Ensure that you have the problem record number available.

To collect diagnostic information using this tool:

1. Go to the server or client on which you suspect the problem to occur.
2. Double-click on the **Problem Determination Tools** icon in the DB2 folder. The Problem Determination Tools folder appears.
3. Double-click on the **Collect Diagnostic Information** icon.
4. If you have more than one instance, select the instance you want to work with and click on **OK**.
5. Enter your problem record number. This is the number that is assigned to your problem when you open a problem with IBM.
6. Click on **OK** to proceed.
7. A panel with a default list of files appears. Click on **Add** to include more files to the list.
8. Click on **Pack** to collect and compress the files.
9. To transmit the file that you created, click on the **Send** push button. Alternatively, you can use the **Send or Receive Information** icon to send the file at a later time.

Files are collected and packed into one file. The name of this file is based on the problem record number.

The tool attempts to collect all items on the following list, which represents the diagnostic information most often needed by DB2 Customer Service. The list cannot be changed.

- All dump files contained under the instance directory. These files, when present, contain DB2 diagnostic information.
- All ndf files under the CMLIB directory. These files, when present, contain APPC configuration information.
- All trap files under the DB2 instance directory. These files, when present, contain system exception information such as access violations.
- CONFIG.SYS file. This file contains statements that set up the system configuration each time you restart the operating system.

- Database directory information.
- Database manager configuration information.
- The db2diag.log file. This file is used by DB2 to log diagnostic information.
- DCS directory information, if DB2 Connect is installed on this workstation.
- LANTRAN.LOG. This file contains diagnostic information when this workstation has communication problems with the LAN.
- LOG0001.DAT. This file is the general system log file for this workstation. It is located under your boot drive.
- Node directory information. This file contains information about the DB2 license of this workstation.
- PROTOCOL.INI. This file provides LAN protocol information about the workstation.
- PSTAT. This collects the process status information on this workstation.
- The services file. This file provides TCP/IP connection information.
- STARTUP.CMD. This file shows which programs are automatically started when you start the operating system.
- SYSLEVEL. This file shows all the software including version and level numbers installed on this workstation.

## Sending or Receiving Information

DB2 for OS/2 provides a tool to assist you in sending diagnostic information to IBM or receiving information from IBM.

To send information using this tool:

1. Double-click on the **Problem Determination Tools** icon in the DB2 folder. The Problem Determination Tools folder appears.
2. Double-click on the **Send/Receive Problem Information** icon. The Internet address and remote directory for the IBM FTP server are specified on the window that appears. You do not need to change these values.
3. Click on **Send**. A window appears where you can enter the name of the file you want to send to IBM. It is recommended that you use the **Collect Diagnostic Information** tool to collect and compress the files to send.
4. For a list of available files, click on **Find**.
5. Find the file you want to send by selecting the drive and directory where the file is located. Click on the file you want to send and click on **OK**.
6. Click on **Send** once you have verified that the name of the file is correct.

To receive information using this tool:

1. Double-click on the **Problem Determination Tools** icon in the DB2 folder. The Problem Determination Tools folder appears.

2. Double-click on the **Send/Receive Problem Information** icon. The Internet address and remote directory for the IBM FTP server are specified on the window that appears. You do not need to change these values.
3. Click on **Get** to view a list of files available from the IBM FTP site. (DB2 Customer Service may have placed files for you to download at this location.)
4. Select a file you would like to receive and click on **Get** to initiate the file transfer.

## Tracing

DB2 Customer Service may request that you take a DB2 trace. This task is made easier on DB2 for OS/2, DB2 for Windows NT, and DB2 for Windows 95, where a graphical interface is available for collecting trace information about DB2 activities. Once the trace information is collected, you can send it to DB2 Customer Service for analysis.

To collect trace information, perform the following steps:

1. For OS/2:
   a. Double-click on the **Problem Determination Tools** icon in the DB2 folder. The Problem Determination Tools folder appears.
   b. Double-click on the **Trace** icon.
2. For Windows NT: Select the **Start -> Programs -> DB2 -> Trace** menu choice.
3. Select **Start** to activate the tracing of DB2 activities. Default trace parameters are already specified.
4. Recreate the problem you are diagnosing by running through a typical sequence of steps.
5. When you have finished recreating the problem, transfer the tracing information from memory to the file specified in **Dump to file** by selecting the **Dump** push button. Ensure that the file name for the dump file is your problem record number.
6. Stop the tracing process by selecting the **Stop** push button.
7. Select **Send** when you are ready to send the information to IBM. A panel appears with the Internet address and remote directory of IBM's FTP server. You typically do not need to change these values.

# Appendix C. 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 file.

## SQLCA Field Descriptions

*Table 1. Fields of SQLCA*

| Name[1] | Data Type | Field Values |
|---|---|---|
| sqlcaid | CHAR(8) | An "eye catcher" for storage dumps containing 'SQLCA'. |
| 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 1. Fields of SQLCA (continued)*

| Name[1] | Data Type | Field Values |
|---|---|---|
| sqlcode | INTEGER | Contains the SQL return code. For specific meanings of SQL return codes, see the message section of the *Message Reference*.<br><br>**Code** **Means**<br>**0** Successful execution (although one or more SQLWARN indicators may be set).<br>**positive** Successful execution, but with a warning condition.<br>**negative** Error condition. |
| sqlerrml | SMALLINT | Length indicator for *sqlerrmc*, in the range 0 through 70. 0 means that the value of *sqlerrmc* is not relevant. |
| sqlerrmc | VARCHAR (70) | Contains one or more tokens, separated by X'FF', that are substituted for variables in the descriptions of error conditions.<br><br>This field is also used when a successful connection is completed.<br><br>When a NOT ATOMIC compound SQL statement is issued, it may contain information on up to 7 errors.<br><br>For specific meanings of SQL return codes, see the message section of the *Message Reference*. |
| sqlerrp | CHAR(8) | 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, SQL06010 means DB2 Universal Database versions for Version 6 Release 1 Modification level 0.<br><br>If SQLCODE indicates an error condition, then this field identifies the module that returned the error.<br><br>This field is also used when a successful connection is completed. |
| sqlerrd | ARRAY | 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. |

*Table 1. Fields of SQLCA (continued)*

| Name[1] | Data Type | Field Values |
|---|---|---|
| sqlerrd(1) | INTEGER | 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] |
| sqlerrd(2) | INTEGER | 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. |
| sqlerrd(3) | INTEGER | 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. |
| sqlerrd(4) | INTEGER | 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. |

*Table 1. Fields of SQLCA (continued)*

| Name[1] | Data Type | Field Values |
|---|---|---|
| sqlerrd(5) | INTEGER | Contains the total number of rows deleted, inserted, or updated as a result of both:<br>• The enforcement of constraints after a successful delete operation<br>• The processing of triggered SQL statements from activated triggers.<br>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. |
| sqlerrd(6) | INTEGER | 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. |
| sqlwarn | Array | 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. |
| sqlwarn0 | CHAR(1) | Blank if all other indicators are blank; contains W if at least one other indicator is not blank. |
| sqlwarn1 | CHAR(1) | 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. |
| sqlwarn2 | CHAR(1) | Contains W if null values were eliminated from the argument of a function. [b] |
| sqlwarn3 | CHAR(1) | Contains W if the number of columns is not equal to the number of host variables. |
| sqlwarn4 | CHAR(1) | Contains W if a prepared UPDATE or DELETE statement does not include a WHERE clause. |
| sqlwarn5 | CHAR(1) | Reserved for future use. |
| sqlwarn6 | CHAR(1) | Contains W if the result of a date calculation was adjusted to avoid an impossible date. |
| sqlwarn7 | CHAR(1) | Reserved for future use. |
| sqlwarn8 | CHAR(1) | Contains W if a character that could not be converted was replaced with a substitution character. |

*Table 1. Fields of SQLCA (continued)*

| Name[1] | Data Type | Field Values |
|---|---|---|
| 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 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 D. 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 file.

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 Service.

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 123.

*Table 2. DB2 Internal Return Codes*

| Internal Return Code | SQL State | Reason Code | Description |
|---|---|---|---|
| 81A7 | | | Invalid alternate |
| 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 |

*Table 2. DB2 Internal Return Codes (continued)*

| Internal Return Code | SQL State | Reason Code | Description |
|---|---|---|---|
| 813A | -295 | | Container names are too long |
| 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 |
| 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 2. 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 2. 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 2. 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 2. 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 2. 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 2. 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 2. 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 |
| D085 | | 21 | Authentication failed due to no user licenses available |
| D107 | -960 | | No more file tokens |

*Table 2. DB2 Internal Return Codes (continued)*

| Internal Return Code | SQL State | Reason Code | Description |
|---|---|---|---|
| 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 |
| 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 |

*Table 2. DB2 Internal Return Codes (continued)*

| Internal Return Code | SQL State | Reason Code | Description |
|---|---|---|---|
| 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 |
| 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 |

*Table 2. DB2 Internal Return Codes (continued)*

| Internal Return Code | SQL State | Reason Code | Description |
|---|---|---|---|
| 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 |
| 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 |

*Table 2. DB2 Internal Return Codes (continued)*

| Internal Return Code | SQL State | Reason Code | Description |
|---|---|---|---|
| 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 |
| 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 |

*Table 2. DB2 Internal Return Codes (continued)*

| Internal Return Code | SQL State | Reason Code | Description |
|---|---|---|---|
| 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 |
| 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 |

*Table 2. DB2 Internal Return Codes (continued)*

| Internal Return Code | SQL State | Reason Code | Description |
|---|---|---|---|
| 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 |
| 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 |

*Table 2. DB2 Internal Return Codes (continued)*

| Internal Return Code | SQL State | Reason Code | Description |
|---|---|---|---|
| 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 |
| 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 |

*Table 2. DB2 Internal Return Codes (continued)*

| Internal Return Code | SQL State | Reason Code | Description |
|---|---|---|---|
| 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 | | | PMAP is of a 1-node nodegroup |

# Appendix E. Notices

Any reference to an IBM licensed program in this publication is not intended to state or imply that only IBM's licensed program may be used. Any functionally equivalent product, program or service that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program, or service. Evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, is the user's responsibility.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the

IBM Director of Licensing
IBM Corporation, North Castle Drive
Armonk, NY 10504-1785
U.S.A.

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.

This publication may contain examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

## Trademarks

The following terms are trademarks or registered trademarks of the IBM Corporation in the United States and/or other countries:

ACF/VTAM
ADSTAR
AISPO
AIX
AIXwindows
AnyNet
APPN
AS/400
CICS
C Set++
C/370
DATABASE 2
DataHub
DataJoiner
DataPropagator
DataRefresher
DB2
DB2 Connect
DB2 Universal Database
Distributed Relational Database Architecture
DRDA
Extended Services
FFST
First Failure Support Technology
IBM
IMS
LAN Distance
MVS/ESA
MVS/XA
OS/400
OS/390
OS/2
PowerPC
QMF
RACF
RISC System/6000
SP
SQL/DS
SQL/400
S/370
System/370
System/390
SystemView
VisualAge
VM/ESA
VSE/ESA
VTAM
WIN-OS/2

## Trademarks of Other Companies

The following terms are trademarks or registered trademarks of the companies listed:

C-bus is a trademark of Corollary, Inc. in the United States, other countries, or both.

HP-UX is a trademark of Hewlett-Packard.

Java, HotJava, Solaris, Solstice, and Sun are trademarks of Sun Microsystems, Inc.

Microsoft, Windows, Windows NT, Visual Basic, and the Windows logo are trademarks or registered trademarks of Microsoft Corporation in the United States, other countries, or both.

PC Direct is a trademark of Ziff Communications Company in the United States, other countries, or both and is used by IBM Corporation under license.

ActionMedia, LANDesk, MMX, Pentium, and ProShare are trademarks of Intel Corporation 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 13
adapter number (NetBIOS) 46
adm subdirectory, tools 13
Administering Satellites Guide and Reference 210
Administration Guide 206
administration of databases 11
Administrative API Reference 207
administrative tools (DB2)
  description 12
  logging 130
  tracing 142
administrative tools (Windows NT) 184
administrator authority 20
agent 113
agent pool 113
AIX commands 167
alert
  overview 132
  SNA 145
  SNA, routing (OS/2) 178
alias 33
APPC, CPI-C and SNA Sense Codes 207
APPC protocol 49
Application Building Guide 207
Application Development Guide 207
application errors and warnings 68
application requesters (AR) 51
application servers (AS) 51
applications
  CLI, ODBC and JDBC 65
  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 87
  slow or hanging because of locking 28
applications *(continued)*
  suspended (DB2 Universal Database Enterprise - Extended Edition for the Solaris Operating Environment and DB2 Universal Database Enterprise - Extended Edition for Windows NT) 86
  suspended (TCP/IP) 41
  suspended or looping (applications) 64
  troubleshooting 61
architectural integrity, checking 14
AS/400, connections 51
authentication
  host connections 56
AutoLoader (DB2 Universal Database Enterprise - Extended Edition for the Solaris Operating Environment and DB2 Universal Database Enterprise - Extended Edition for Windows NT) 82

## B

backup
  plan 12
  troubleshooting 21
  using Control Center 12
bin subdirectory, tools 13
binary data, loading (DB2 Universal Database Enterprise - Extended Edition for the Solaris Operating Environment and DB2 Universal Database Enterprise - Extended Edition for Windows NT) 85

## C

call stack 129
CLI Guide and Reference 208
CLI/ODBC/JDBC 145
CLI trace 145
client
  APPC connections 49
  host 58
  host connections 51
  IPX/SPX connections 43
  named pipe connections 48
  NetBIOS connections 46
  TCP/IP connections 40
  troubleshooting 4, 33
Client Configuration Assistant 35
code page conversions 63
collecting data for DB2 customer service 10
command line processor, unable to use 26
Command Reference 208
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;) 169
  db2_call_stack 129
  db2atld (DB2 Universal Database Enterprise - Extended Edition for the Solaris Operating Environment and DB2 Universal Database Enterprise - Extended Edition for Windows NT) 83
  db2bfd 13
  db2cat 13
  db2ckmig 17
  db2dart 14
  db2drdat 143
  db2evmon 16
  db2expln 30
  db2fexmt 30
  db2flsn 14
  db2ipxad 14
  db2level 14
  db2look 14
  db2recr 15
  db2resdb 15
  db2set 34
  db2snmpd 133
  db2sql92 15
  db2start 79
  db2tbst 15
  db2trc 135
  db2untag 15
  ddcstrc (DB2 Connect) 143
  dynexpln 30
  list applications 7
  load 25
  monitoring 16

commands, DB2 *(continued)*
netstat (DB2 Universal Database Enterprise - Extended Edition for the Solaris Operating Environment and DB2 Universal Database Enterprise - Extended Edition for Windows NT;) 170
rah (DB2 Universal Database Enterprise - Extended Edition for the Solaris Operating Environment and DB2 Universal Database Enterprise - Extended Edition for Windows NT;) 169
register 43
reorg 29
reorgchk 29
runstats 29
troubleshooting 13
commands, OS/2
cmtrace (SNA trace) 145
epw 174
epwinst 174
epwrout 178
pstat 179
syslog 173
commands, UNIX-based
cmtrace (SNA trace) 145
correct syntax 26
dbx 164
df 168
errpt (AIX) 167
iostat 7, 169
ipcrm 7
ipcs 7
kill 7
lsattr (AIX) 167
lsps (AIX) 167
netstat 169
ps 166
pstack (Solaris) 168
spmon (DB2 Universal Database Enterprise - Extended Edition for the Solaris Operating Environment and DB2 Universal Database Enterprise - Extended Edition for Windows NT;) 170
truss (SVR4) 168
vmstat 168
xmperf 167
communicating with IBM 10
compiling 63
configuration parameters
DB2_RR_TO_RS 29
diaglevel 118
diagpath 118
dlchktime 28
locklist 28
locktimeout 28
logfilsiz 18
logprimary 18
logsecond 18
maxagents 113
maxlocks 28
nname (NetBIOS) 46
notifylevel 118
num_poolagents 113
sheapthres 30
sortheap 30
configuring clients 34
connection
analyzing events 13
changing number for DRDA 56
testing on the server 36
troubleshooting 33
unsuccessful 38
Connectivity Supplement 208
Control Center 12, 190
search utility 190
coordinator agent 113
core files (DB2) 128
core files (UNIX) 164
corruption, checking for 14
customer service, contacting 10

## D

data
architectural integrity, checking 14
backups 21
binary, loading (DB2 Universal Database Enterprise - Extended Edition for the Solaris Operating Environment and DB2 Universal Database Enterprise - Extended Edition for Windows NT) 85
format (DB2 Universal Database Enterprise - Extended Edition for the Solaris Operating Environment and DB2 Universal Database Enterprise - Extended Edition for Windows NT) 83
importing 26
data *(continued)*
indexing (DB2 Universal Database Enterprise - Extended Edition for the Solaris Operating Environment and DB2 Universal Database Enterprise - Extended Edition for Windows NT) 85
loading 25
restoring 23
splitting and loading (DB2 Universal Database Enterprise - Extended Edition for the Solaris Operating Environment and DB2 Universal Database Enterprise - Extended Edition for Windows NT) 82
Data Movement Utilities Guide and Reference 208
database
administration troubleshooting 11
alias 33
consistency 17
creating (DB2 Universal Database Enterprise - Extended Edition for the Solaris Operating Environment and DB2 Universal Database Enterprise - Extended Edition for Windows NT) 81
design 11
objects, displaying with Control Center 12
production, mimicking 14
recovery (DB2 Universal Database Enterprise - Extended Edition for the Solaris Operating Environment and DB2 Universal Database Enterprise - Extended Edition for Windows NT) 86
remote 34
system catalog statistics, mimicking 14
database directory 33
database manager
monitoring 15
monitoring with Control Center 13
database partition 77
Database System Monitor 187
Event Monitor 188
Snapshot Monitor 187
datalinks 90

datalinks, backup data links manager environment 93
datalinks, data access token 90
datalinks, general 89
datalinks, import operation 94
datalinks, load considerations 95
datalinks, migration considerations 96
datalinks, recovery data links manager environment 91
datalinks,export operation 94
DB2
  administrative tools 12
  DB2 Universal Database Enterprise - Extended Edition for the Solaris Operating Environment and DB2 Universal Database Enterprise - Extended Edition for Windows NT 77
  host connections 51
  information, finding 8
  maintaining 9
DB2 Connect
  tracing 143
  troubleshooting 51
DB2 Connect Enterprise Edition for OS/2 and Windows NT Quick Beginnings 211
DB2 Connect Enterprise Edition for UNIX Quick Beginnings 211
DB2 Connect Personal Edition Quick Beginnings 208
DB2 Connect User's Guide 208
DB2 customer service, contacting 10
DB2 Data Links Manager for AIX Quick Beginnings 211
DB2 Data Links Manager for Windows NT Quick Beginnings 211
DB2 Enterprise - Extended Edition for UNIX Quick Beginnings 210
DB2 Enterprise - Extended Edition for Windows NT Quick Beginnings 210
DB2 Index Advisor 192
DB2 library
  books 206
  Information Center 214
  language identifier for books 212
  late-breaking information 213
  online help 204
  ordering printed books 217
DB2 library *(continued)*
  printing PostScript books 216
  searching online information 216
  setting up document server 215
  SmartGuides 203
  structure of 203
  viewing online information 213
DB2 Personal Edition Quick Beginnings 210
DB2 Query Patroller Administration Guide 211
DB2 Query Patroller Installation Guide 211
DB2 Query Patroller User's Guide 211
DB2 Satellite Edition 101
DB2 Universal Database Enterprise - Extended Edition for the Solaris Operating Environment and DB2 Universal Database Enterprise - Extended Edition for Windows NT
  commands 169
  determining failing nodes 79
  developing applications 87
  monitoring applications 65
  processes 114
  severe error 78
  starting 79
  troubleshooting 77
DB2COMM registry value 37
db2dart tool 14
db2diag.log file
  additional information in syslog (UNIX) 161
  diagnosing if listeners are up 113
  dump files 128
  example with DB2 trace 140
  examples 123
  header 120
  inconsistent data 19
  interpreting 120
  NetBIOS entries 48
  pruning 120
  relationship to Windows NT event viewer 185
  SQLCA structures 122
  tracing if insufficient 135
  trap files 129
  troubleshooting client connections 37
deadlocks 27
Desktop Management Interface (DMI) service layer 132
diaglevel, setting 118
diagnostic tools
  OS/2 171
  UNIX 161
  Windows 95 and Windows 98 183
  Windows NT 183
diagpath, setting 118
directories
  changing with Control Center 12
Distributed Relational Database Architecture (DRDA)
  tracing 143
  troubleshooting 51
Dr. Watson (Windows) 183
dump files 128
  core file (UNIX) 164
  OS/2 alerts 179
  traces 137
dump utility, standalone (OS/2) 171

## E

environment variables 34
error code 123
error log (OS/2) 172
error record, FFST for OS/2 error log 177
Event Analyzer 13
Event Monitor 188
  matching records with their application 189
  Output Records 189
  reading an event monitor trace 189
event snapshots 16
event viewer (Windows NT) 185
exceptions
  responding to 6
  trap file 129
explain facility 30
export message files 131

## F

FFDC 117
FFST/2 implementation (OS/2) 174
files
  autoexec.bat (OS/2) 44
  config.sys 44
  core (UNIX) 164
  db2alert.log 132
  db2cc.log 130
  db2cli.ini 34, 66
  db2diag.log file 120
  dbttrc.trc 143

files *(continued)*
dump 128
net.cfg (OS/2) 44, 45
nwcalls.dll (Win-OS/2) 45
odbc.ini 34
services (TCP/IP) 41
size limit 25
syslog (UNIX) 161
system (Solaris) 169
TLI*.DLL (Novell IPX/SPX) 43
tli_spx.dll (Win-OS/2) 45
trap 129
firewall 111
first failure data capture (FFDC)
alerts 132
collecting on OS/2 219
configuration parameters 118
db2diag.log file 120
dump files 128
first failure data capture 117
introduction 10
overview 117
SQLCA structures 122
stack traceback 129
tracing if insufficient 135
trap files 129
fix packs, applying 9
format of data (DB2 Universal Database Enterprise - Extended Edition for the Solaris Operating Environment and DB2 Universal Database Enterprise - Extended Edition for Windows NT) 83
forums 9
free space, identifying (UNIX) 168
ftp sites 9

## G

general protection faults (GPF), responding to 6
Glossary 208
graphical user interface (GUI)
administrative tools 12
tracing 143

## H

hangs, responding to 6
hardware errors
identifying (AIX) 167
identifying (HP-UX) 168
hexadecimal values in db2diag.log file 123
history of problems, maintaining 3
host
troubleshooting connections to 51
HPS switch, checking (DB2 Universal Database Enterprise - Extended Edition for the Solaris Operating Environment and DB2 Universal Database Enterprise - Extended Edition for Windows NT;) 170
hung applications 64

## I

I/O activities, monitoring (UNIX) 169
IBM, contacting 10
ICLUI trace 143
idle agent 113
import message files 131
importing data 26
index
architectural integrity, checking 14
failing in DB2 Universal Database Enterprise - Extended Edition for the Solaris Operating Environment and DB2 Universal Database Enterprise - Extended Edition for Windows NT 85
inconsistent, re-creating 15
Index SmartGuide 192
indoubt transactions (DB2 Universal Database Enterprise - Extended Edition for the Solaris Operating Environment and DB2 Universal Database Enterprise - Extended Edition for Windows NT) 87
Information Center 12
information on DB2, finding 8
Installation and Configuration Supplement 208
Internet Engineering Task Force (IETF) 133
IPX/SPX protocol 43
isolation levels (CS, RS) 29

## J

job scheduling, using Control Center 12

## K

keepalive settings (TCP/IP) 41
kernel, updating (UNIX) 21
kernel debugging (OS/2) 171

## L

LAN, determining availability 39
Lana number (Windows 95) 47
link time 63
listeners
definition 113
testing 36
load message files 131
LOAD utility 82
loading data, troubleshooting tips 25
location of problem, determining 4
locking 27
log space 18
logging, all platforms 117
logs
increasing space for importing data 26
system log (OS/2) 172
loops
in applications 64
responding to 6
Lotus Approach 65

## M

maintaining DB2 9
Management Information Base (MIB) 133
media management with Control Center 12
memory, identifying (AIX) 167
message files, import, export and load 131
Message Reference 209
messages, responding to 5
Microsoft Access 65
Microsoft Query 65
migration
troubleshooting 17
Version 1 of DB2 for OS/2 15
misc subdirectory, tools 13
mobile 101
monitor switches 16
monitoring database manager with Control Center 13
MVS, connections 51

## N

named pipes 48
NetBIOS connections lost 23
NetBIOS protocol 46
NetQuestion 103
network traffic, identifying (UNIX) 169
newsgroups 9
next key locking 29
node directory 34

nodes (DB2 Universal Database Enterprise - Extended Edition for the Solaris Operating Environment and DB2 Universal Database Enterprise - Extended Edition for Windows NT)
  definition 77
  failing 79
  slow 81
  won't start 81

## O

ODBC application messages 68
ODBC applications
  configuring driver 66
  mixed case problems 66
  troubleshooting tips 65
online help 8
optimizer, viewing access plans 13
origins of problems, determining 4
OS/2 commands 171
OS/390, connections 51

## P

packed descriptors for tables, viewing 13
page rate, monitoring (UNIX) 168
paging space, identifying (AIX) 167
passwords 20
patches, applying 9
performance
  data, displaying (UNIX) 167
  improving 29
  monitoring with Control Center 13
  using the database monitor 15
Performance Monitor 13
performance monitor (Windows NT) 185
pervasive computing 101
point-in-time monitoring 16
problem description, determining 4
problem determination 3
Problem Determination Tools folder 219
process model
  differences between Intel and UNIX 114
  displaying threads (OS/2) 179
  overview 111
  ps command (UNIX) 166
  viewing 115
process status command 166
PTFs, applying 9

## Q

Quick Beginnings for OS/2 210
Quick Beginnings for UNIX 210
Quick Beginnings for Windows NT 210

## R

rebinding 26
recovery
  DB2 server 21
  DB2 Universal Database Enterprise - Extended Edition for the Solaris Operating Environment and DB2 Universal Database Enterprise - Extended Edition for Windows NT 86
  operating system 6
redbooks 9
registry values 34
remote debugging (OS/2) 171
replication 97
  DpropR 97
Replication Guide and Reference 209
resources, increasing (IPX/SPX) 44
responding to DB2 users 4
restoring data 23
restoring Version 1 database image (DB2 for OS/2) 15
return codes
  byte-reversal 123
  example 126
  list 231
roadmaps
  logging, tracing, and alerts 107
ROLLFORWARD PENDING 24

## S

SAM tool (HP-UX) 168
satellite control database 101
Satellite Edition 101
schema names, considerations during migration 18
screen captures (Windows) 185
segmentation violation
  logs 129
  responding to 6
server
  administrative tools 12
  client connections 37
  collecting data for DB2 Customer Service 31
  failure 18
  timing out (NetBIOS) 47
  troubleshooting 4, 11
service tools 219
services file (TCP/IP) 41
setting up document server 215
Setup Communications tool 34
signal 129
Simple Network Management Protocol (SNMP) 132
SmartGuides 12
Snapshot Monitor 187
SNMP subagent, starting 133
software fixes, applying 9
Solaris OS commands (useful for debugging) 167
splitting data (DB2 Universal Database Enterprise - Extended Edition for the Solaris Operating Environment and DB2 Universal Database Enterprise - Extended Edition for Windows NT) 82
SQL Advise Facility 192
SQL Getting Started 209
SQL messages
  definition 5
  responding to 5
  SQL0204N 66
  SQL0294N 15
  SQL0805N 26
  SQL0818N 26
  SQL0902C 21, 24
  SQL0911N 27
  SQL0984C 18
  SQL1003N 65
  SQL1015N 79
  SQL1016N 21
  SQL1018N 21
  SQL1034C 79
  SQL1042C 20
  SQL1061W 86
  SQL1109N 45
  SQL1224N 79
  SQL1277N 24, 25
  SQL1402N 57
  SQL1403N 19, 39, 57
  SQL2419N 22
  SQL30073N 58
  SQL30081N (IPX/SPX) 43
  SQL30081N (NetBIOS) 46
  SQL30081N (TCP/IP) 40
  SQL30082N 57
  SQL30082N (NetBIOS) 48
  SQL3805N 25
  SQL4908N 22
  SQL5043N 37
SQL Reference 209
SQL state 5

SQLCA (SQL communication area) 225
DB2 Universal Database Enterprise - Extended Edition for the Solaris Operating Environment and DB2 Universal Database Enterprise - Extended Edition for Windows NT nodes 79
interpreting 122
stack traceback
displaying (Solaris) 168
for suspended system 8
gathering 129
starting a trace 137
stored procedure
developing 61
fenced 114
Stored Procedure Builder 192
subagent 113
suspended applications 64
suspensions, responding to 6
symptom record, FFST for OS/2 178
syslog (UNIX) 161
system calls, tracing (UNIX) 168
system catalog statistics, mimicking 14
system configuration, using Control Center 12
system file (Solaris) 169
system log (SYSLOG)
OS/2 172
UNIX 161
System Monitor Guide and Reference 209
System Performance Monitor for OS/2 (SPM/2) 171
system resources, increasing (IPX/SPX) 44
Systems Network Architecture (SNA)
overview 51
routing alerts (OS/2) 178
tracing 145
troubleshooting 55
SystemView agent 132

## T

table
recovery 195
system catalog tables, mimicking 14
viewing packed descriptors 13
table space
container, using again with db2untag 15
recovery 195
state, getting description of with db2tbst 15
TCP/IP protocol 40
test tool to mimic production databases 14
threads 111
tools
client connectivity 34
database administration 12
datalinks 90
datalinks, backup data links manager environment 93
datalinks, data access token 90
datalinks, export operation 94
datalinks, general 89
datalinks, import operation 94
datalinks, load considerations 95
datalinks, migrations considerations 96
datalinks, recovery data links manager environment 91
trace 145
TRACE, keyword 154
trace facility, DB2 135
TRACECOMM, keyword 155
TRACEFILENAME, keyword 156
TRACEFLUSH, keyword 157
TRACEPATHNAME, keyword 158
traces
administration tools (DB2) 142
DB2 135
DB2 Connect 143
DRDA 143
dumping (DB2) 137
examples (DB2) 139
graphical interface (ICLUI) 143
SNA 145
turning off (DB2) 137
verifying (DB2) 138
trap 129
troubleshooting
abends, responding to 6
APPC protocol 49
applications 61
clients 33
contacting IBM 10
database administration 11
troubleshooting *(continued)*
DB2 Universal Database Enterprise - Extended Edition for the Solaris Operating Environment and DB2 Universal Database Enterprise - Extended Edition for Windows NT 77
db2diag.log file 117
getting started 3
history of problems, maintaining 3
host connections 51
information, finding 8
IPX/SPX protocol 43
loops, responding to 6
maintaining products 9
messages, responding to 5
named pipes 48
NetBIOS protocol 46
origins and symptoms of problems, determining 4
problem description, determining 4
return codes 231
servers 11
suspensions, responding to 6
TCP/IP protocol 40
where to find a problem 4
Troubleshooting Guide 209
Troubleshooting Page 191

## U

UNIX-based commands 167
updates to DB2, applying 9
user-defined functions (UDFs)
developing 61
fenced 114
user forums 9
user name
authentication on host 56
case sensitivity 56
not valid on Windows 95 and Windows 98 39
user names 20
utilities, unable to use 26

## V

Visual Explain 13
VM, connections 51
VSE, connections 51

## W

WarpCenter (OS/2 Warp Version 4) 171
watchdog process 115

What's New 209
World Wide Web 9

## Contacting IBM

This section lists ways you can get more information from IBM.

If you have a technical problem, please take the time to review and carry out the actions suggested by the *Troubleshooting Guide* before contacting DB2 Customer Support. Depending on the nature of your problem or concern, this guide will suggest information you can gather to help us to serve you better.

For information or to order any of the DB2 Universal Database products contact an IBM representative at a local branch office or contact any authorized IBM software remarketer.

**Telephone**

If you live in the U.S.A., call one of the following numbers:

- 1-800-237-5511 to learn about available service options.
- 1-800-IBM-CALL (1-800-426-2255) or 1-800-3IBM-OS2 (1-800-342-6672) to order products or get general information.
- 1-800-879-2755 to order publications.

For information on how to contact IBM outside of the United States, see Appendix A of the IBM Software Support Handbook. You can access this document by accessing the following page:

`http://www.ibm.com/support/`

then performing a search using the keyword "handbook".

Note that in some countries, IBM-authorized dealers should contact their dealer support structure instead of the IBM Support Center.

**World Wide Web**

http://www.software.ibm.com/data/
http://www.software.ibm.com/data/db2/library/

The DB2 World Wide Web pages provide current DB2 information about news, product descriptions, education schedules, and more. The DB2 Product and Service Technical Library provides access to frequently asked questions, fixes, books, and up-to-date DB2 technical information. (Note that this information may be in English only.)

**Anonymous FTP Sites**

ftp.software.ibm.com

Log on as anonymous. In the directory /ps/products/db2, you can find demos, fixes, information, and tools concerning DB2 and many related products.

**Internet Newsgroups**

comp.databases.ibm-db2, bit.listserv.db2-l

These newsgroups are available for users to discuss their experiences with DB2 products.

**CompuServe**

**GO IBMDB2** to access the IBM DB2 Family forums

All DB2 products are supported through these forums.

To find out about the IBM Professional Certification Program for DB2 Universal Database, go to http://www.software.ibm.com/data/db2/db2tech/db2cert.html

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

S10J-8169-01