

# **IBM DB2 Universal Database Troubleshooting Guide Version 5**

Document Number S10J-8169-00





IBM DB2 Universal Database

# Troubleshooting Guide

*Version 5*





IBM DB2 Universal Database

# Troubleshooting Guide

*Version 5*

Before using this information and the product it supports, be sure to read the general information under Appendix F, "Notices" on page 175.

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

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

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

© **Copyright International Business Machines Corporation 1993, 1997. All rights reserved.**

Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

---

## Contents

<b>Welcome to DB2</b> . . . . .	ix
Roadmap . . . . .	ix
Conventions . . . . .	x

---

### Part 1. Basic DB2 Troubleshooting . . . . . 1

<b>Chapter 1. Good Troubleshooting Practices</b> . . . . .	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 . . . . .	9
Contacting IBM for Support . . . . .	9
<b>Chapter 2. Troubleshooting the DB2 Universal Database Server</b> . . . . .	11
DB2 Administrative Tools . . . . .	12
Graphical Tools . . . . .	12
Miscellaneous Troubleshooting Tools . . . . .	13
The Database System Monitor . . . . .	14
Troubleshooting Tips . . . . .	15
Migrating to Version 5 . . . . .	16
Server Fails . . . . .	16
Can't Administer Databases . . . . .	18
Backing Up and Restoring Data . . . . .	19
Loading Data . . . . .	20
Importing Data . . . . .	20
Can't Use Commands, Utilities, or the Command Line Processor . . . . .	20
Tables or Rows Locked . . . . .	21
User Authentication with DB2 for Windows NT . . . . .	23
Improving Database Performance . . . . .	27
Still Have Problems? . . . . .	28
<b>Chapter 3. Troubleshooting on the Client</b> . . . . .	29
Overview of Client Connectivity . . . . .	29
Client Connectivity Tools . . . . .	30
Troubleshooting Tips . . . . .	30
Determining the Scope of a Client Problem . . . . .	31
Cannot Make Initial Connection After Installation . . . . .	33
Clients Suddenly Experiencing Problems . . . . .	33
User Name Not Valid on Windows 95 . . . . .	34

Using the TCP/IP Protocol . . . . .	35
Using the IPX/SPX Protocol . . . . .	37
Using the NetBIOS Protocol . . . . .	40
Using Named Pipes . . . . .	42
Using the APPC Protocol . . . . .	43
Still Have Problems? . . . . .	43
<b>Chapter 4. Troubleshooting Host Communications . . . . .</b>	<b>45</b>
Troubleshooting Host Connections . . . . .	46
Troubleshooting Tips for DB2 Connect . . . . .	47
Initial Questions . . . . .	47
Cannot Make Initial Connection After Installation . . . . .	47
Problems Encountered After Initial Connection . . . . .	48
DB2 Connect Connection Using SNA Does Not Work . . . . .	49
Changing Number of Connections . . . . .	49
Authentication Problems . . . . .	50
Windows Clients Won't Connect . . . . .	51
Troubleshooting Tips for Host Connections to DB2 Universal Database . . . . .	51
Still Have Problems? . . . . .	52
<b>Chapter 5. Troubleshooting Applications . . . . .</b>	<b>53</b>
Troubleshooting Tips . . . . .	53
Developing Applications . . . . .	54
Compiling and Linking Applications . . . . .	54
Running Applications . . . . .	55
Using CLI/ODBC Applications . . . . .	56
Still Have Problems? . . . . .	57
<b>Chapter 6. Troubleshooting DB2 Universal Database Extended Enterprise Edition . . . . .</b>	<b>59</b>
Troubleshooting Tips . . . . .	59
Extended Enterprise Edition Won't Install . . . . .	60
Severe Error Occurs on System . . . . .	60
Determining Failing Node . . . . .	61
Database Manager Won't Start or Hangs . . . . .	61
Cannot Create Database . . . . .	62
Commands Cannot Be Issued . . . . .	63
Problems with Splitting and Loading Data . . . . .	63
Problems with Indexing Data . . . . .	66
Problems During Recovery . . . . .	66
Developing Applications in DB2 Extended Enterprise Edition . . . . .	67
Still Have Problems? . . . . .	67

---

**Part 2. Advanced DB2 Troubleshooting . . . . . 69**

<b>Chapter 7. Roadmap . . . . .</b>	<b>71</b>
-------------------------------------	-----------

<b>Chapter 8. The DB2 Process Model . . . . .</b>	<b>75</b>
---	-----------



Description of the DB2 Process Model . . . . .	75
Differences Between Intel and UNIX . . . . .	78
Viewing System Threads and Processes . . . . .	78
<b>Chapter 9. Using Logged Information . . . . .</b>	<b>81</b>
Understanding First Failure Data Capture . . . . .	81
Setting DIAGLEVEL and DIAGPATH . . . . .	81
Interpreting the db2diag.log File . . . . .	83
Interpreting db2diag.log Entry Headers . . . . .	83
Interpreting an SQLCA Structure . . . . .	85
Interpreting Hexadecimal Codes . . . . .	85
Some Examples for Interpreting the db2diag.log File . . . . .	86
Dump Files . . . . .	90
Trap Files . . . . .	91
Gathering Stack Traceback Information on UNIX-Based Systems . . . . .	91
Using the Administrative Tools Log . . . . .	92
Administrative Tools Log File Sample . . . . .	93
Using Message Files . . . . .	93
Alerts . . . . .	94
SNMP Alerts . . . . .	94
<b>Chapter 10. Taking Traces . . . . .</b>	<b>97</b>
Using the DB2 Trace Facility (db2trc) . . . . .	97
Starting a DB2 Trace . . . . .	99
Verifying a DB2 Trace . . . . .	100
Examples of Using db2trc . . . . .	100
Taking Other Traces . . . . .	103
Administration Tools Trace . . . . .	103
Taking a Graphical Interface Trace . . . . .	104
Taking a CLI/ODBC/JDBC Trace . . . . .	105
Taking a DB2 Connect Trace (ddcstrc) . . . . .	106
Taking a DRDA Trace (db2drdat) . . . . .	106
Taking an SNA Trace . . . . .	107
<b>Chapter 11. Diagnostic Tools for UNIX-Based Systems . . . . .</b>	<b>109</b>
Using the System Error Log (syslog) . . . . .	109
Setting Up the System Error Log . . . . .	110
Understanding the System Error Log . . . . .	111
Using the Core File . . . . .	112
Accessing Core File Information . . . . .	112
Viewing Process Status Using the ps Command . . . . .	113
Other Commands . . . . .	114
AIX Troubleshooting Commands . . . . .	114
UNIX-Based Troubleshooting Commands . . . . .	115
Performance Monitoring Tools . . . . .	115
Commands for DB2 Extended Enterprise Edition . . . . .	116
<b>Chapter 12. Diagnostic Tools for OS/2 . . . . .</b>	<b>119</b>

Using the OS/2 Logging Facility . . . . .	120
Preliminary Considerations . . . . .	120
Using OS/2 Logging Facility Commands . . . . .	120
Using the FFST for OS/2 Utility . . . . .	121
Adding the FFST for OS/2 Program Group to the Desktop Manager Window . . . . .	121
Configuring and Enabling the FFST for OS/2 Utility . . . . .	122
Accessing FFST for OS/2 Error Records . . . . .	125
Routing Generic Alerts . . . . .	125
Using the pstat Command . . . . .	126
<b>Chapter 13. Diagnostic Tools for Windows Operating Systems . . . . .</b>	<b>129</b>
Diagnostic Tools for Windows Version 3.1 and Windows 95 . . . . .	129
Diagnostic Tools for Windows NT . . . . .	129
Using Administrative Tools Provided by Windows NT . . . . .	130
Taking Screen Captures . . . . .	131
Using DB2-Supplied Tools . . . . .	131
<b>Chapter 14. Diagnostic Tools for Macintosh . . . . .</b>	<b>133</b>

---

**Part 3. Appendixes . . . . . 135**

<b>Appendix A. Using the Problem Determination Tools Folder . . . . .</b>	<b>137</b>
Accessing Internet Information for DB2 for OS/2 . . . . .	137
Accessing Internet Information for DB2 for Windows NT . . . . .	137
Updating DB2 Products . . . . .	137
Setting Diagnostic Levels . . . . .	138
Collecting Diagnostic Information . . . . .	138
Sending or Receiving Information . . . . .	140
Tracing . . . . .	140
<b>Appendix B. SQL Communication Area (SQLCA) . . . . .</b>	<b>143</b>
Viewing the SQLCA Interactively . . . . .	143
SQLCA Field Descriptions . . . . .	143
Order of Error Reporting . . . . .	146
DB2 Extended Enterprise Edition Usage of the SQLCA . . . . .	146
<b>Appendix C. DB2 Internal Return Codes . . . . .</b>	<b>147</b>
<b>Appendix D. How the DB2 Library Is Structured . . . . .</b>	<b>163</b>
SmartGuides . . . . .	163
Online Help . . . . .	164
DB2 Books . . . . .	166
About the Information Center . . . . .	170
<b>Appendix E. How to Read Syntax Diagrams . . . . .</b>	<b>173</b>
<b>Appendix F. Notices . . . . .</b>	<b>175</b>
Trademarks . . . . .	175

Trademarks of Other Companies . . . . .	176
<b>Index</b> . . . . .	<b>177</b>
<b>Contacting IBM</b> . . . . .	<b>179</b>



---

## 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

The guide is divided into two parts:

Part 1	Summarizes good troubleshooting practices, and provides troubleshooting tips for administering DB2, connecting to DB2 clients and to host databases, using Extended Enterprise Edition, and developing applications for DB2.
Part 2	Provides information on advanced troubleshooting techniques such as logging, tracing, and platform-specific diagnostics. This information is best used in consultation with DB2 Customer Service.

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 (see the *Administration Getting Started*)
- The information described in the *Quick Beginnings* guides, and (for DB2 Extended Enterprise Edition) the *DB2 Extended Enterprise Edition Quick Beginnings*.

For more information on the information available in the DB2 library, see Appendix D, "How the DB2 Library Is Structured" on page 163.

---

## Roadmap

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

If you want to...	Refer to...
Find out more about DB2	Appendix D, "How the DB2 Library Is Structured" on page 163
Solve a DB2 problem	"Determining Origins and Symptoms of Problems" on page 4, and then the particular chapter or chapters in Part 1, "Basic DB2 Troubleshooting" on page 1 that address your problem

Contact DB2 Customer Service	"Contacting IBM for Support" on page 9
Collect information as instructed by DB2 Customer Service	Part 2, "Advanced DB2 Troubleshooting" on page 69
Find the latest information on the Internet	"Using All Available Information" on page 8
Find and install DB2 fixes	"Updating DB2 Products" on page 9
Solve problems at the DB2 server	Chapter 2, "Troubleshooting the DB2 Universal Database Server" on page 11
Solve problems with client connections to a DB2 server	Chapter 3, "Troubleshooting on the Client" on page 29
Solve problems with DB2 Connect connections to host databases on VM, VSE, MVS, or AS/400, and with host connections to DB2 Universal Database	Chapter 4, "Troubleshooting Host Communications" on page 45
Solve problems with applications working with DB2	Chapter 5, "Troubleshooting Applications" on page 53
Solve problems with DB2 Extended Enterprise Edition	Chapter 6, "Troubleshooting DB2 Universal Database Extended Enterprise Edition" on page 59
Make applications run faster	"Improving Database Performance" on page 27 and Chapter 5, "Troubleshooting Applications" on page 53
Solve problems with data replication	<i>Replication Guide and Reference</i>
Find out about the db2diag.log log file, and about tracing	Part 2, "Advanced DB2 Troubleshooting" on page 69
Solve problems with ADSM	<i>The Administration Guide</i>
Solve problems with components that work in conjunction with DB2 (for example, Net.Data or DB2 Extenders)	The manuals for the products, and the DB2 Product and Service Technical Library at <a href="http://www.software.ibm.com/data/db2/library/">http://www.software.ibm.com/data/db2/library/</a>

---

## 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, 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.
- For information on the syntax diagrams used in this guide, see Appendix E, “How to Read Syntax Diagrams” on page 173.





---

## Part 1. Basic DB2 Troubleshooting

There are different kinds of problems that you may encounter using DB2, and various techniques that you can use to deal with those problems. This part discusses these techniques.

Chapter 1, "Good Troubleshooting Practices" on page 3	Provides techniques for solving DB2 problems
Chapter 2, "Troubleshooting the DB2 Universal Database Server" on page 11	Provides troubleshooting tips for database administrators
Chapter 3, "Troubleshooting on the Client" on page 29	Provides troubleshooting tips for connecting clients to DB2 databases
Chapter 4, "Troubleshooting Host Communications" on page 45	Provides troubleshooting tips on DB2 Connect connections to host databases, and on host connections to DB2 Universal Database
Chapter 5, "Troubleshooting Applications" on page 53	Provides troubleshooting tips for developers and users of applications accessing data stored in DB2
Chapter 6, "Troubleshooting DB2 Universal Database Extended Enterprise Edition" on page 59	Provides troubleshooting tips for DB2 Extended Enterprise Edition users



---

## 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 9
- “Contacting IBM for Support” on page 9

---

### Maintaining a Problem History

It is highly recommended that you maintain a history of reported problems and the actions you took to identify or resolve them. This information can 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 fix a problem, start by getting a good description of the problem. With this description, you can work on determining the origin of the problem. 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

Refer to the following sections for information:

- “Determining the Location of a Problem”
- “Responding to Unexpected Messages or SQL Codes” on page 5
- “Responding to Abends” on page 6
- “Responding to Suspensions and Loops” on page 6

Also, use the detailed information provided in the other chapters of Part 1 to help you determine and solve your particular problem.

## 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

"Understanding First Failure Data Capture" on page 81. 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 SQL code, an 8-digit alphanumeric message identification number. 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, see the *Message Reference*.
  - The text of the message (especially if the message does not include an identification number or a code).
  - The SQLCA if available.
  - Any action suggested in the message.
  - Diagnostic files, such as the db2diag.log file. In addition, note any operating system diagnostic files such as 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 69.
  - 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*", 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 "Understanding First Failure Data Capture" on page 81.
  - The SQLCA structure from the db2diag.log or syslog files, or as captured by an application. See "Interpreting an SQLCA Structure" on page 85.
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 pack 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 “Understanding First Failure Data Capture” on page 81.
      - For UNIX, the syslog and core files. See Chapter 11, “Diagnostic Tools for UNIX-Based Systems” on page 109.
      - For OS/2, the SYSLOG file. See “Using the syslog Command” on page 120.
      - For Windows operating systems, Dr. Watson logs, and, for Windows NT, the event log. See Chapter 13, “Diagnostic Tools for Windows Operating Systems” on page 129.
    - 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 101.
- See Part 2, “Advanced DB2 Troubleshooting” on page 69 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 `sqlib/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 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 114)
  - For the Windows NT operating system, use its performance monitor (see “Diagnostic Tools for Windows NT” on page 129)
  - For the OS/2 system, use tools such as SPM/2 (see Chapter 12, “Diagnostic Tools for OS/2” on page 119)

3. If the problem persists, try to collect the following additional information before contacting DB2 Customer Service:

- Any information logged by DB2. See “Understanding First Failure Data Capture” on page 81.
- 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 101.
  - 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 91.

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 D, “How the DB2 Library Is Structured” on page 163.
- Online help is available:
  - For user interfaces on the OS/2, Windows NT, and Windows 95 operating systems, 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.)
- 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 pack 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](ftp://ftp.software.ibm.com). Then go to the directory `/ps/products/db2/fixes`.
- On OS/2, use the Problem Determination folder on your desktop. See Appendix A, "Using the Problem Determination Tools Folder" on page 137.

Once you download a fix pack, use its readme file (`readme.txt`) for prerequisites and installation instructions.

Note that the readme file may instruct you to rebind the applications and utilities on your DB2 server and clients. For information on binding, see the *Quick Beginnings* for the appropriate 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 69.

---

## 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 69.

When you can proceed no further on your own, contact DB2 Customer Service. For a list of worldwide IBM support locations, see Appendix A of the IBM Software Support Handbook. You can access this document at:

<http://ps.software.ibm.com/pbin-usa-ps/getobj.pl?/pdocs-usa/webhndbk.html>

Alternatively, select the "Roadmap to IBM Support" item at:

<http://www.ibm.com/support/>

---

## Chapter 2. Troubleshooting the DB2 Universal Database Server

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

- IBM DB2 Universal Database Workgroup Edition
- IBM DB2 Universal Database Enterprise Edition
- IBM DB2 Universal Database Extended Enterprise Edition (see also Chapter 6, “Troubleshooting DB2 Universal Database Extended Enterprise Edition” on page 59)

For more information on administering databases, see:

- The *Quick Beginnings* guides for information on setting up and migrating databases
- The *DB2 Extended Enterprise Edition Quick Beginnings* guide for information on setting up Extended Enterprise Edition
- The *Administration Guide* for detailed information on database administration
- The *Administration Getting Started* for a high-level overview of database administration
- The *System Monitor Guide and Reference* for information on monitoring databases
- The *Command Reference* for information on DB2 commands
- The *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. See the information in the DB2 library (especially the *Administration Guide*) to find out more about operations required of DB2 administrators. Typical operations include:

- Migrating a database from a previous version
- 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 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 27.)
- [ ] 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 83.)

---

## 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 27 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 NT and Windows 95 operating systems. For information on these tools, see their online help and the *Administration Getting Started*.

#### 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:
  - Create databases, table spaces, and tables
  - Back up and restore databases
  - Tune the performance of databases by optimizing their configuration
- Use graphical interfaces to perform the following tasks:
  - Analyze the access plans chosen by the DB2 optimizer for SQL statements and packages. This functionality is called “Visual Explain”. To invoke it, click mouse button 2 on a database object in the Control Center, and select **Explain SQL** from the pop-up menu. See also “Improving Database Performance” on page 27.
  - 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, click mouse button 2 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 mouse button 2 on a database object, and select **Monitor events** from a pop-up menu.
- For Extended Enterprise 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 misc, bin, or 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 important tools available to you:

### **db2bfd**

Provides a bind file description. For more information on binding, see the *Embedded SQL Programming Guide*, the *Command Reference*, or the *API Reference*.

### **db2cat**

Dumps the contents of packed descriptors for tables and formats them in a readable form. (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.

### **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.

#### **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 sqllmon() API. (You must have SYSADM, SYSCTRL, or SYSMAINT authority.)

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

- Taking a snapshot. Use the GET SNAPSHOT command from the command line; the Control Center on the OS/2, Windows 95, or Windows NT operating systems

for a graphical interface; or write your own application, using the `sqlmonss()` API call.

- Using an event monitor. An 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 on the **Event Analyzer** icon in the Control Center (on the Windows 95, Windows NT, or OS/2 systems) to use a graphical interface to view the trace, search for keywords, and filter out unwanted data.

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

---

## Troubleshooting Tips

This section describes how to deal with some frequently encountered problems faced by users working with a DB2 server. It addresses the following topics:

- “Migrating to Version 5” on page 16
- “Server Fails” on page 16
- “Can’t Administer Databases” on page 18
- “Backing Up and Restoring Data” on page 19
- “Loading Data” on page 20
- “Importing Data” on page 20
- “Can’t Use Commands, Utilities, or the Command Line Processor” on page 20
- “Tables or Rows Locked” on page 21
- “User Authentication with DB2 for Windows NT” on page 23
- “Improving Database Performance” on page 27

For additional troubleshooting information on this topic, see:

- Chapter 6, “Troubleshooting DB2 Universal Database Extended Enterprise Edition” on page 59 for information on administering DB2 Extended Enterprise Edition systems
- “Troubleshooting Tips for Host Connections to DB2 Universal Database” on page 51 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/>.

## Migrating to Version 5

If you experience problems migrating to Version 5, 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. (See the *Quick Beginnings* book for 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 Fails” for details.
- [ ] If migrating from a Version 1 DB2 for OS/2 database, you use the **db2resdb** command.

## Server Fails

### Symptom

The DB2 server does not function correctly.

### 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 83.
- [ ] *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 32.

- [ ] 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:

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

- [ ] 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**.

- [ ] 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 Extended Enterprise Edition, see also “Database Manager Won’t Start or Hangs” on page 61.

## Can't Administer Databases

### Symptom

You have problems installing or administering a database server.

### Action

Ensure that:

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

You can find out what your group is by using the User Manager tool to display user names and group memberships. To invoke this tool, select **Start**, 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.) See the *Quick Beginnings* guide for your platform for details.

## Backing Up and Restoring Data

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

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

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

For help on backup and recovery see the *Administration Guide* 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.

### Can't restore a database or table space backup

#### Symptom

Data cannot be restored.

#### Action

Ensure that:

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

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

- [ ] 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.)

## Loading Data

### 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, see the *Command 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 Splitting and Loading Data” on page 63 for more information on loading data in Extended Enterprise Edition.)

## Importing Data

### Symptom

You are having problems importing data.

### Action

- [ ] Ensure that there is enough log space available. (See “Server Fails” on page 16.)
- [ ] 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.

## Can't Use 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, see the *Administration Guide*.

### Diagnosing locking problems

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

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

Use an event monitor for deadlocks on a database to keep a 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 indicates 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 14 and the *System Monitor Guide and Reference*.

## Applications slow or appear to hang

### Symptom

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

### Possible Cause

There is lock contention or lock escalation.

### Action

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

If escalation is occurring, ensure that:

- [ ] Applications use an appropriate frequency of commits.  
Take a monitor snapshot for locks to determine which applications are causing other applications to wait for locks. Also, monitor the *db2.lock\_waits* performance variable.
- [ ] The values of the database configuration parameters *maxlocks* and *locklist*, which determine the number of locks allowed, are appropriate.
- [ ] The *locktimeout* database configuration parameter is set appropriately. When it is turned on, this parameter can help avoid hangs due to lock contention, and can tell you if lock contention is your problem.
- [ ] The *dlchktme* database configuration parameter is set appropriately. This parameter defines the frequency at which the database manager checks for deadlocks among all the applications connected to a database. If it is set high, CPU time is saved but deadlocks may not be detected soon enough.
- [ ] Other users are not preparing and binding applications.  
The procedure of preparing and binding applications requires that locks be obtained on system catalog tables. This procedure should be performed in off-peak hours.
- [ ] Lock waits or deadlocks are not caused by next key locking.  
**Next key locking** guarantees a cursor stability (CS) isolation level by automatically locking the next key for all INSERT and DELETE statements and the next higher key value above the result set for SELECT statements. It is required to guarantee ANSI and 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 ANSI and SQL standard CS,

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

## **SQL0911N received**

### **Symptom**

The SQL0911N message is received.

### **Possible Cause**

There are deadlocks or timeouts.

### **Action**

Check the reason code in the SQLCA to determine if deadlocks or timeouts are causing the problem (see “Interpreting an SQLCA Structure” on page 85).

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 timeouts are not captured using the deadlock event monitor.

## **User Authentication with DB2 for Windows NT**

User authentication can cause problems for Windows NT users because of the way the operating system authenticates. This section describes some considerations for user authentication under DB2 for Windows NT:

- “User name and group name limitations”
- “The DB2 for Windows NT security service”
- “Installing DB2 on a backup domain controller” on page 24
- “Authentication with groups and domain security” on page 25

### **User name and group name limitations**

- User names and group names are limited to 8 characters within DB2. Therefore the default administrator user name on Windows NT (“ADMINISTRATOR”) does not work when installing or accessing DB2 for Windows NT.
- User names under Windows NT are not case sensitive; however, passwords are case sensitive.
- User names and group names can be a combination of upper- and lowercase characters. However, they are usually converted to uppercase when used within DB2. For example, if you connect to the database and create the table schema1.table1, this table is stored as SCHEMA1.TABLE1 within the database. (If you wish to use lowercase object names, issue commands from the command line processor, enclosing the object names in quotation marks, or use third-party ODBC front-end tools.)

### **The DB2 for Windows NT security service**

In DB2 Universal Database we have integrated the authentication of user names and passwords into the DB2 System Controller. The Security Service is only required when connecting a client to a server that is configured for authentication CLIENT.

## Installing DB2 on a backup domain controller

In a Windows NT LAN environment a user can be authenticated at either a primary or backup domain controller. This feature is very important in large distributed LANs with one central primary domain controller and one or more backup domain controllers at each site. Users can then be authenticated on the backup domain controller at their site instead of requiring a call to the primary domain controller for authentication.

The advantage of having a backup domain controller (BDC), in this case, is that users are authenticated faster and the LAN is not as congested as it would have been had there been no BDC.

For Version 5, authentication can occur at a backup domain controller under the following conditions:

- The DB2 for Windows NT server is installed on the backup domain controller
- The DB2DMNBCKCTRL profile variable is set appropriately

If the DB2DMNBCKCTRL profile variable is not set or is set to blank, DB2 for Windows NT performs authentication at the primary domain controller.

If the DB2DMNBCKCTRL profile variable is set to a question mark (DB2DMNBCKCTRL=?) then DB2 for Windows NT will perform its authentication on the backup domain controller under the following conditions:

- The Cached Primary Domain under the registry editor is the domain at which the machine is located. (You can find this setting under **HKEY\_LOCAL\_MACHINE -> Software -> Microsoft -> Windows NT -> Current Version -> WinLogon.**)
- The Server Manager shows the backup domain controller as active and available. (That is, the icon for this machine is not greyed out.)
- The registry for the DB2 for Windows NT server indicates that the system is a backup domain controller on the specified domain.

Under normal circumstances the setting DB2DMNBCKCTRL=? will work; however, it will not work in all environments. The information supplied about the servers on the domain is dynamic, and Computer Browser must be running to keep this information accurate and current. Large LANs may not be running Computer Browser and therefore Server Manager's information may not be current. In this case, there is a second method to tell DB2 for Windows NT to authenticate at the backup domain controller: set DB2DMNBCKCTRL=xxx where xxx is the machine name of the backup domain controller and DB2 for Windows NT server. With this setting, authentication will occur on the backup domain controller under the following conditions:

- The Cached Primary Domain under the registry editor is the domain at which the machine is located. (You can find this setting under **HKEY\_LOCAL\_MACHINE -> Software -> Microsoft -> Windows NT -> Current Version -> WinLogon.**)
- The machine is configured as a backup domain controller for the specified domain. (If the machine is set up as a backup domain controller but for another domain, this setting will result in an error.)



## Authentication with groups and domain security

DB2 for Windows NT supports the following types of groups:

- Local groups
- Global groups
- Global groups as members of local groups.

DB2 for Windows NT enumerates the local and global groups that the user is a member of, using the security database where the user was found. DB2 Universal Database provides an override that forces group enumeration to occur on the local Windows NT server where DB2 is installed, regardless of where the user account was found. This override can be achieved using the following commands:

- For global settings: `db2set -g db2_grp_lookup=local`
- For instance settings: `db2set -i db2_grp_lookup=local`

To view all DB2 profile variables that are set, type `db2set -a11`.

For DB2 for Windows NT to work with domain security, you must grant authority and privileges to a local group. User names within the local and global groups **MUST** be defined on the same domain as the local or global group in order to be authenticated correctly.

If the `db2_grp_lookup` profile variable is set to local, then DB2 tries to find a user on the local machine only. If the user is not found on the local machine, or is not defined as a member of a local or global group, then authentication fails. DB2 does **not** try to find the user on another machine in the domain or on the domain controllers.

If the `db2_grp_lookup` profile variable is not set then:

1. DB2 first tries to find the user on the same machine.
2. If the user name is defined locally, the user is authenticated locally.
3. If the user is not found locally, DB2 attempts to find the user name on its domain, and then on trusted domains.

The following examples illustrate how DB2 for Windows NT can support domain security. In the first example, the connection works because the user name and local group are on the same domain. In the second example, the connection does not work because the user name and local or global group are on different domains.

**Example of a Successful Connection:** The connection works in the following scenario because the user name and local or global group are on the same domain.

Note that the user name and local or global group do not need to be defined on the domain where the database server is running, but they must be on the same domain as each other.

Domain1	Domain2
A trust relationship exists with Domain2.	<ul style="list-style-type: none"> <li>• A trust relationship exists with Domain1.</li> <li>• The local or global group grp2 is defined.</li> <li>• The user name id2 is defined.</li> <li>• The user name id2 is part of grp2.</li> </ul>
The DB2 server runs in this domain. The following DB2 commands are issued from it: REVOKE CONNECT ON db FROM public GRANT CONNECT ON db TO GROUP grp2 CONNECT TO db USER id2	
The local or global domain is scanned but id2 is not found. Domain security is scanned.	
	The user name id2 is found on this domain. DB2 gets additional information about this user name (that is, it is part of the group grp2).
The connection works because the user name and local or global group are on the same domain.	

**Example of an Unsuccessful Connection:** The connection does not work in the following scenario because the user name is defined on a different domain than where the local or global group is defined.

Domain1	Domain2
A trust relationship exists with Domain2.	<ul style="list-style-type: none"> <li>• A trust relationship exists with Domain1.</li> <li>• The local or global group grp2 is defined.</li> </ul>
<ul style="list-style-type: none"> <li>• The global group grp1 is defined.</li> <li>• The user name id1 is defined.</li> <li>• The user name id1 is part of grp1.</li> </ul>	
	Domain1\grp1 is part of grp2.
The DB2 server runs on this domain. The following DB2 commands are issued from it: REVOKE CONNECT ON db FROM public GRANT CONNECT ON db TO GROUP grp2 CONNECT TO db USER id1	

The local or global domain is scanned and id1 is found. DB2 gets information for this user name (that is, the user name id1 is part of grp1 and the group grp1 is part of Domain2\grp2).	
	The group grp2 exists on this domain.
The connection does not work because the local or global group is on Domain2 and the actual user name is defined on Domain1.  The connection would work if the following command was issued instead: GRANT CONNECT ON db TO GROUP grp1	

## Improving Database Performance

Database performance is a complicated subject 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.
- 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.
- 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 why a query is executing in a particular way
- Design application programs and databases
- Determine when an application should be rebound

Visual Explain provides an accurate, complete, and easy-to-understand picture of access plans. Alternatively, the following commands are available from the misc subdirectory of the sqllib directory:

- **db2expln** describes the access plan selected for static SQL statements in the packages stored in the system catalog tables. It can be used to obtain a quick explanation of the chosen access plan for packages for which explain data was not captured at bind time.
- **db2fmt** provides a textual description of the explain tables and the operators chosen for access plans of static SQL.

- **dynexpln** converts a dynamic statement into a static statement and runs **db2expln** against it. This is a quick way to see the basic access plan for a dynamic statement, but is not necessarily accurate.

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

- 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. It should also be at least twice as big as *sortheap*, the database configuration parameter for sort heap. For information, see the *Administration Guide*.
- Locking and how it is handled may affect performance of applications. See “Applications slow or appear to hang” on page 22.
- 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 115.

For more information on performance:

- Use the keyword "performance" from the DB2 Product and Service Technical Library at <http://www.software.ibm.com/data/db2/library/>.
- See the *Administration Getting Started* guide, the *Administration Guide*, 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 “Understanding First Failure Data Capture” on page 81)
- For UNIX systems, any system core files (see “Using the Core File” on page 112)
- Output from **db2dart** if you suspect inconsistency in the data (see “Miscellaneous Troubleshooting Tools” on page 13).

---

## Chapter 3. Troubleshooting on the Client

This chapter discusses how to deal with problems with a database client.

For more information, see:

- The *Quick Beginnings* guide for your platform, which provides setup information
- The *Installing and Configuring DB2 Clients* online 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 45, 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 various areas where connectivity problems typically occur.

Client workstations are connected to a DB2 server through a communication protocol (for example, TCP/IP, APPC, 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 actual 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.

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.

See the *Quick Beginnings* guide for your platform for details on maintaining the information in the database and node directories.

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. On Windows NT, Windows 95, OS/2, and UNIX-based systems, 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`.  
  
(On Windows 3.1, environment variables are stored in the `db2.ini` file and can be updated by editing this file. On Macintosh systems, there is a DB2 preferences file.)
- The `db2cli.ini` and `odbc.ini` files are ASCII files that store values for DB2 CLI configuration. For information, see the *CLI Guide and Reference*.

Refer to the *Quick Beginnings* guide for your platform for details on client connectivity.

## Client Connectivity Tools

After installing DB2, you can use the following tools on the OS/2, Windows NT, or Windows 95 operating systems 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. To update communications for a server instance, select its object from the Control Center and select **Setup Communications** from the pop-up menu.

Note that 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. This tool is available for Windows NT, Windows 95, and OS/2 servers and clients. (For Windows 3.x, the Client Setup tool is available instead.)

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

For more information, see your *Quick Beginnings* guide.

---

## 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 31
- “Cannot Make Initial Connection After Installation” on page 33
- “Clients Suddenly Experiencing Problems” on page 33
- “User Name Not Valid on Windows 95” on page 34
- “Using the TCP/IP Protocol” on page 35
- “Using the IPX/SPX Protocol” on page 37
- “Using the NetBIOS Protocol” on page 40
- “Using Named Pipes” on page 42
- “Using the APPC Protocol” on page 43

For related information, see:

- The *Quick Beginnings* guide for your platform
- 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 45 for troubleshooting host communications
- Chapter 5, “Troubleshooting Applications” on page 53 for troubleshooting applications running on clients, including ODBC applications

**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

When you have a problem with a client, you must start with the following questions, which will help you determine exactly where a problem is occurring:

- Were the client and server installed correctly?
- Are the communication products installed and operational on the client and server?
- Is the problem on one or many clients?
- Did the client function correctly in the past?
- Is the database manager started on the server with appropriate communications listeners?
- Can you get a connection from the client to another server?
- Can you get a connection from another client to the server?

This section helps you address these questions by explaining how to test connections to the server and use the db2diag.log file to verify if communications listeners are started.

### Testing connections on the server

If you are having problems with a client connection, try testing 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 probably with the server.
2. Test a loopback connection. That is, try a remote connection on the server using the same protocol as on the client. Catalog a database on the server with a node directory entry that references the server machine, and then try the connection from the server to itself.
  - If the connection works, then the problem is with 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 you set up on the server so that you can use this technique to diagnose connectivity problems whenever they come up.

## 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 “Understanding First Failure Data Capture” on page 81 for details on using this file.

For example, you may receive the SQL5043N message after issuing **db2start** on the server, indicating that one or more protocols failed to start successfully. The db2diag.log file may provide additional information in this situation.

The following steps provide a procedure for you to follow when looking for the cause of server problems that may be affecting your clients:

1. Set the DIAGLEVEL to 4 on the server:

```
db2 UPDATE DATABASE MANAGER CONFIGURATION USING DIAGLEVEL 4
db2 terminate
```
2. Disconnect all applications connected to the server:

```
db2 force application all
```
3. Restart the server:

```
db2stop
db2start
```
4. Examine the db2diag.log file on the server.

For each protocol specified in *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 8, “The DB2 Process Model” on page 75.)



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. Refer to the following sections for checklists:

- "SQL5043N received on server for TCP/IP" on page 36
  - "SQL5043N received on server for IPX/SPX" on page 39
  - "SQL5043N received on server for NetBIOS" on page 41
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.

## Cannot Make Initial Connection After Installation

"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, review the following checklist to ensure that the installation steps were followed:

- [ ] Were the client and server 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, and Windows NT, specify error log and history files with the /I1 and /I2 options respectively. (If you do not include these options with the installation request, the files are not kept.)
- For Windows 3.1, logs are kept in \db2log\db2.log, in the drive where Windows is installed.

- [ ] Were all the prerequisite software products installed?

- [ ] Are the communication products installed and operational?

- [ ] Can you get a connection from another client to the server?

If you can, you know that the server is working. If you cannot, the problem may be with the network or the server.

- [ ] Can you get a connection from the client to another server? If you can, then the client is working.

## Clients Suddenly Experiencing Problems

If you suddenly encounter a problem with one or more clients that could previously connect to the server, review the following questions to narrow down the scope of your problem.

### Is it just a single client that is experiencing difficulties?

- [ ] Can other network-enabled applications run on this client? If not, the problem is probably 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.
- [ ] Have there been recent changes to the client that might be affecting it? (For example, was another product or fix pack installed?)
- [ ] Have you exceeded resource limits on the client (for example, memory)?

### Is more than one client experiencing difficulties?

- [ ] Is the LAN available? For example, can you use World Wide Web browsers, 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 32 for details.
- [ ] Have you exceeded resource limits on the server (for example, memory)?

## User Name Not Valid on Windows 95

### Symptom

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

### Possible Cause

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

Note that Windows 95 allows you to log on without a user name and password. In this situation, the implicit user name and password are null, and the CONNECT statement will fail.

### Action

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

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

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

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

## Using the TCP/IP Protocol

This section covers some common troubleshooting tips related to the TCP/IP communication protocol.

### 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 machines).

#### Possible Cause

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

#### Action

Ensure that:

- [ ] DB2START was issued and the TCP/IP listener was started on the server (see "Using the db2diag.log file to diagnose server communication problems" on page 32).
- [ ] 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 for Windows NT, Windows 95, or OS/2 operating systems
- The DB2 Client Setup window on the Windows 3.1 operating system
- The CATALOG DATABASE and CATALOG TCPIP NODE commands (see the *Command Reference*).

- [ ] 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.

## **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* is not correct.)
- The TCP/IP services file is not correct. (For example, the *svcename* 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 that the KEEPALIVE settings affect all TCP/IP applications running on the machine.

- For Windows NT and Windows 95:

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_keeppreq` with the **net tune** command (for details, type `man net tune`).
- For Solaris systems:  
Change the value of the network option `tcp_keepalive_interval` with the following command:  

```
ndd -set /dev/tcp tcp_keepalive_interval value
```

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

```
/etc/conf/bin/idd tune TCPTV_KEEP_IDLE_SECS value  
/etc/conf/bin/idd tune TCPTV_KEEPINTVL_SECS value
```

(For details, type `man idd tune`.) 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.

## Using the IPX/SPX Protocol

This section covers some common troubleshooting tips related to 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 *fileserv* 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. For more information on the **REGISTER** command, see the *Command Reference*.
- [ ] If the server machine was moved on the network and its IPX/SPX internetwork address was changed, the DB2 server's internetwork address was deregistered before any change, and reregistered after the change.
- [ ] If file server addressing mode is being used to connect to the database, the object name that represents the DB2 server instance (that is, stores the IP address of the server instance) is in the bindery of the file server.
- [ ] If direct addressing mode is being used, the following values are in the client's node directory:
  - The file server entry is specified as an asterisk (\*)
  - The value of *objectname* is the server's IPX/SPX internetwork address.

Issue **db2ipxad** on the server to get the server's IPX/SPX internetwork address. This command is in the misc subdirectory of the sqllib directory.

## SQL30081N on OS/2 or Windows 3.1

### Symptom

For OS/2 and Windows 3.1, 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. Alternatively it may be in the NWCLIENT directory on Windows 3.1.
- [ ] 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 NWCLIENT directory for Windows 3.1 or 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 *fileserv*, *objectname*, or *ipx\_socket* values are not correct.)

### Action

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

## Using the NetBIOS Protocol

This section covers some common troubleshooting tips related to the NetBIOS communication protocol. Note that 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 a cause. Note that 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 41.)

- [ ] 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, verify that NetBIOS is a supported protocol. (Use the **db2set db2comm** command.)
- 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:

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 timeout 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 NT and Windows 95 operating systems, the NetBIOS network control settings
  - For DOS or Windows Version 3.1, the config.sys file or the autoexec.bat file
- 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.

## 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.

## Using the APPC Protocol

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?

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

---

## Still Have Problems?

If you still have problems with your client, contact DB2 Customer 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 “Understanding First Failure Data Capture” on page 81)
- Output from the following commands:
  - LIST DB DIRECTORY
  - LIST NODE DIRECTORY
  - GET DBM CONFIG
- ODBC/CLI traces for ODBC applications (see “Taking a CLI/ODBC/JDBC Trace” on page 105)

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 for OS/390, DB2 for MVS/ESA, DB2 for VM, DB2 for VSE, or DB2 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 (formerly DDCS single user), for single users
  - DB2 Connect Enterprise Edition (formerly DDCS multi-user gateway), for multiple users through a gateway

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

For information on DB2 Connect, see:

- The *DB2 Connect Enterprise Edition Quick Beginnings* for your platform
- The *DB2 Personal Edition Quick Beginnings* for your platform
- The *DB2 Connect User's Guide*
- The *DB2 Connectivity Supplement*

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

- The *DRDA Connectivity Guide*
- The *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 "Using the APPC Protocol" on page 43.)
- The TCP/IP communication protocol is supported by DB2 Universal Database and by Version 5.1 of DB2 for OS/390. (For more information, see "Using the TCP/IP Protocol" on page 35.)

This chapter provides the following troubleshooting sections:

- "Troubleshooting Host Connections" on page 46
- "Troubleshooting Tips for DB2 Connect" on page 47

- “Troubleshooting Tips for Host Connections to DB2 Universal Database” on page 51

**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 “Using the APPC Protocol” on page 43.)
- [ ] For TCP/IP communication, are the host name and server name correct? (See “Using the TCP/IP Protocol” on page 35.)
- [ ] 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.)

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”
- “Cannot Make Initial Connection After Installation”
- “Problems Encountered After Initial Connection” on page 48
- “DB2 Connect Connection Using SNA Does Not Work” on page 49
- “Changing Number of Connections” on page 49
- “Authentication Problems” on page 50

### 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 46.
- [ ] For SNA sessions, see “DB2 Connect Connection Using SNA Does Not Work” on page 49.

#### 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.)

### Cannot Make 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 *Installing and Configuring DB2 Clients*?
- [ ] 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 information, see the chapter on working with instances in the *DB2 Connect Enterprise Edition Quick Beginnings*.)

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

- [ ] See “Troubleshooting Host Connections” on page 46 and, for SNA sessions, “DB2 Connect Connection Using SNA Does Not Work” on page 49.

**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? (See the *DB2 Connect Enterprise Edition Quick Beginnings*.)

## 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 46 and, for SNA sessions, “DB2 Connect Connection Using SNA Does Not Work” on page 49.
- [ ] If you suspect a problem with your communication protocol, see Chapter 3, “Troubleshooting on the Client” on page 29.

**Are there any special or unusual operating circumstances?**

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

**Is there relevant diagnostic information?**

- [ ] Were any SQL messages or SQL states returned? To look up an SQL state or SQL code, see the *Message Reference*.
- [ ] Check the db2diag.log file on the server, particularly for SQLCA information. (See “Understanding First Failure Data Capture” on page 81.)



## 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 "Using the APPC Protocol" on page 43.)
- 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 information, see *DB2 Connect Enterprise Edition 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 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, see the *Administration Guide*.

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, see the MVS Server worksheet in the *DB2 Connect Enterprise Edition 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.

#### 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 Won't 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, see the section on configuring national language support in the *DB2 Connect Enterprise Edition 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 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 files for gateways, clients, and DB2 Universal Database servers. For details on how to work with these files, see “Understanding First Failure Data Capture” on page 81.
- ODBC/CLI traces for ODBC applications. See “Taking a CLI/ODBC/JDBC Trace” on page 105.
- Output from the **LIST DCS DIRECTORY** command.

At that time, you may be requested to take SNA, DB2 Connect, or DRDA traces. (See Chapter 10, “Taking Traces” on page 97.)

---

## Chapter 5. Troubleshooting Applications

For information on creating applications to work with DB2, refer to the following publications:

- *Embedded SQL Programming Guide*, for information on how to design and code application programs that use embedded SQL statements to access DB2 databases in supported languages. This book provides details on how to code user-defined functions and stored procedures (including stored procedures using Java).
- *API Reference*, for information on the use of application programming interfaces (APIs) to execute DB2 administrative functions.
- *CLI Guide and Reference*, for information on coding with Call Level Interface (CLI), including stored procedures using CLI. CLI is a callable SQL interface based on the X/Open CLI specification compatible with Microsoft Corporation's ODBC.
- The *Administration Guide* for information on authorization and on tuning application performance.
- *Building Applications for Windows and OS/2 Environments*, *Building Applications for UNIX Environments*, and *DB2 SDK for Macintosh Building Your Applications*, for environment setup information and step-by-step instructions to compile, link, and run DB2 applications.
- *Road Map to DB2 Programming*, for a new user's introduction to DB2 programming interfaces, including Java applications.
- *SQL Getting Started*, for a new user's introduction to SQL.

---

### Troubleshooting Tips

This section describes how to deal with some frequently encountered problems faced by developers of applications that access a DB2 server. It addresses the following topics:

- "Developing Applications" on page 54
- "Compiling and Linking Applications" on page 54
- "Running Applications" on page 55
- "Using CLI/ODBC Applications" on page 56

In addition, see "Developing Applications in DB2 Extended Enterprise Edition" on page 67.

**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/>.

## 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. (See “Diagnosing locking problems” on page 21.)
- [ ] Your application is bound before you issue any static SQL statements. For information on the `sqlabndx` bind API, see the *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 *Embedded SQL Programming Guide*, and use the following sample programs: `util.c` for C, `checkerr.cbl` for COBOL, and `util.f` for FORTRAN.

Those 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. See the *Embedded SQL Programming 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 *Embedded SQL Programming Guide*.
- [ ] 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*.

## 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. See the *Building Your Applications* book for your platform.
- [ ] No compiler error messages are received. (Correct any syntax errors before proceeding.)

At link time, ensure that:

- [ ] The link options and link libraries are defined properly. For information, see the *Building Applications* guide 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 investigated any error messages you received. (For example, embedded SQL statements, DB2 API, or CLI statements may return 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 21.)

## 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. (For information, see “Diagnosing locking problems” on page 21 and the *System Monitor Guide and Reference*.)

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 preferably just after you run **db2start**. Use the following command:

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

For DB2 Extended Enterprise Edition, use:

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

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

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

For DB2 Extended Enterprise Edition use:

```
;db2_all "db2 GET SNAPSHOT FOR ALL APPLICATIONS ON database"
```

For information on **db2\_all**, see “Commands for DB2 Extended Enterprise Edition” on page 116.

## Using CLI/ODBC Applications

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

### Developing ODBC applications

When developing ODBC applications, note the following:

- [ ] You must use the PATCH1 keyword to specify a workaround for any known problems with CLI/ODBC applications. Set this keyword by editing the db2cli.ini file, or (for OS/2, Windows 95, and Windows NT) by accessing the **Service** tab of the CLI/ODBC Settings in the Client Configuration Assistant. See the *Installing and Configuring DB2 Clients* online guide for information on this keyword.
- [ ] 16-bit applications are supported by the 32-bit IBM DB2 ODBC driver shipped with Windows 95 and Windows NT clients, with the Microsoft ODBC Driver Manager performing the required conversion between the 16-bit application and the 32-bit ODBC driver.  
  
16-bit applications shipped before 1995 use an old set of 16-bit Driver Manager DLLs that do not function well in the 32-bit environment. For the old set of DLLs, problems include:
  - Traps occur in odbc.dll or other ODBC-related DLLs.
  - 16-bit applications do not see data sources cataloged using the 32-bit ODBC Administrator (from the Control Panel).If (and only if) you are experiencing these problems, replace the 16-bit ODBC Driver Manager DLLs in the system subdirectory of the Windows operating system directory with the files in DB2's SQLLIB\MSODBC16 subdirectory.

### CLI/ODBC 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 **Settings** push button. The Settings window appears.
3. Select the **Settings** push button. The ODBC Data Source Configuration 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.)

### Can't perform update

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

For information, see the 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?")

---

## Still Have Problems?

If you still have problems with your applications:

- Go to the DB2 Product and Service Technical Library at <http://www.software.ibm.com/data/db2/library/> to get 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 “Understanding First Failure Data Capture” on page 81)
- An ODBC/CLI trace for CLI/ODBC applications (see “Taking a CLI/ODBC/JDBC Trace” on page 105)

---

## Chapter 6. Troubleshooting DB2 Universal Database Extended Enterprise Edition

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

In addition to the information in this chapter, see:

- “Commands for DB2 Extended Enterprise Edition” on page 116 for some useful UNIX-based commands
- The *DB2 Extended Enterprise Edition Quick Beginnings*
- The *Administration Guide* for conceptual information

**Note:** The subject of parallel databases is complicated, 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 Extended Enterprise Edition, see the *Administration Guide*.

---

### Troubleshooting Tips

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

- “Extended Enterprise Edition Won’t Install” on page 60
- “Severe Error Occurs on System” on page 60
- “Determining Failing Node” on page 61
- “Database Manager Won’t Start or Hangs” on page 61
- “Cannot Create Database” on page 62
- “Commands Cannot Be Issued” on page 63
- “Problems with Splitting and Loading Data” on page 63
- “Problems with Indexing Data” on page 66
- “Problems During Recovery” on page 66
- “Developing Applications in DB2 Extended Enterprise Edition” on page 67

**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/>.

## Extended Enterprise Edition Won't Install

### Symptom

The DB2 Extended Enterprise Edition product won't install.

### Action

Ensure that:

- [ ] The DB2 product directory on each node has enough free disk space. This directory is `/usr/lpp/db2_05_00` on AIX.
- [ ] 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. (That is, transactions stay indoubt.) 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. (See the *Command Reference*.)

- 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 indoubt.
- 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 for additional details. For information, see

"Determining Failing Node" on page 61 and "Interpreting the db2diag.log File" on page 83.

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

## 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 and SQLCA structures, see "Interpreting the db2diag.log File" on page 83.

## Database Manager Won't 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 SYSMANT authority. See the *Administration Guide* 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 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 Extended Enterprise Edition" on page 116.
- [ ] The db2nodes.cfg file (located in the \$HOME/sqllib directory) contains the correct information for all nodes defined in the system (that is, 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 Extended Enterprise Edition 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 sqllib/tmp directory and, if required, update them so that the instance ID has write permission for the directory. Note that if the permissions are not correct, a problem may have occurred during installation and setup of the system.

## **db2start hangs**

### **Symptom**

The **db2start** command hangs.

### **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 **/etc/rc.nfs**, and ensure that this command is in your /etc/inittab directory.

## **Nodes won't start or are very slow**

### **Symptom**

Some nodes cannot be started, or run extremely slowly.

### **Action**

Ensure that:

- [ ] You installed the DB2 Extended Enterprise 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 Extended Enterprise 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. See the *Administration Guide* for details.

## Commands Cannot Be Issued

### Symptom

DB2 or operating system commands are not recognized.

### Action

Ensure that:

- [ ] Your DB2 Extended Enterprise 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 with Splitting and Loading Data

For more information on using the AutoLoader to split data and load it on a node, see the *Administration Guide*.

For more information on the **db2autold** command, type `db2auto1d -h` from the `sqlib/misc` directory.

### Data is not split

#### Symptom

The **db2autold** program apparently completed successfully, but the data is not split.

#### Possible Cause

The **db2autold** program was set only to analyze the data.

#### Action

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

## Data is not split correctly

### Symptom

The **db2autold** program apparently completed successfully, but the data is not split correctly.

### Action

Ensure that:

- Binary data is not in a character column.

The **db2autold** program cannot detect binary values in a column of any data type, unless the **BINARYNUMERICS** or **PACKEDDECIMAL** options are specified on the load command.

- The *SplitNodes* and *OutputNodes* parameters are set correctly.

Otherwise, output data files may not be correct. See the *Administration Guide* for details.

## 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 **db2autold** command, message files are located in the current working directory. There is a file for each node, and each file is named `load_log.nnn`, where *nnn* is the node number as specified in the `db2nodes.cfg` file.

- The format of the data is correct.

See the *Command Reference* for more information on using correct data formats. Note in particular 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.

## Data doesn't 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 **db2autold** program creates a header file to prevent data from being loaded to the wrong node. The header information is checked by LOAD.

However, 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 **db2autold** 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, however, you manually alter the header, this compromises the validation measure built into the AutoLoader.

## 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 **db2autold** to see if all the rows were rejected. If some rows were loaded, see the *Administration Guide*.

## Problems with Indexing Data

### Symptom

The CREATE INDEX statement fails or hangs.

### Possible Cause

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

### Action

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

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

## 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, particularly when operations that require a large volume of logging activities (such as massive inserts) were running before the database crashed. This is a normal situation.

#### Action

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

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

## Developing Applications in DB2 Extended Enterprise Edition

In a multinode 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 85 and Appendix B, “SQL Communication Area (SQLCA)” on page 143.)  
If a node returns an error or warning, a node number that maps to the db2nodes.cfg file is placed in the SQLERRD(6) field of the SQLCA. (If an SQL statement or API call is successful, the node number in the SQLERRD(6) field is that of the coordinator agent.)
- Indoubt transactions may be left in the database if an application ends abnormally during 2-phase commit. (An indoubt transaction pertains to global transactions when some phases complete successfully, but the system fails before all phases can complete. The database is left in an inconsistent state.) For information on handling indoubt transactions, see the *Administration Guide*.
- 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 Extended Enterprise Edition Usage of the SQLCA” on page 146 for more special considerations when using the SQLCA in a multinode environment.

---

## Still Have Problems?

If you still have problems with your DB2 Extended Enterprise 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, and any trap or dump files, for each node (see “Understanding First Failure Data Capture” on page 81).
- Any SYSLOG files (see “Using the System Error Log (syslog)” on page 109).

- Any core files (see “Using the Core File” on page 112).

---

## Part 2. Advanced DB2 Troubleshooting

This part provides information on how to use DB2 and system resources for logging and tracing DB2 problems.

Chapter 7, "Roadmap" on page 71	Provides a roadmap of logging and tracing tools
Chapter 8, "The DB2 Process Model" on page 75	Explains the DB2 process model, which is useful on some platforms for determining if any part of your DB2 system is broken down
Chapter 9, "Using Logged Information" on page 81	Explains logging facilities available to you for diagnosing problems
Chapter 10, "Taking Traces" on page 97	Discusses traces and how to take them
Chapter 11, "Diagnostic Tools for UNIX-Based Systems" on page 109	Discusses additional UNIX system tools
Chapter 12, "Diagnostic Tools for OS/2" on page 119	Discusses additional OS/2 tools
Chapter 13, "Diagnostic Tools for Windows Operating Systems" on page 129	Discusses additional Windows tools
Chapter 14, "Diagnostic Tools for Macintosh" on page 133	Discusses troubleshooting procedures on Macintosh clients



## Chapter 7. 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 table provides a roadmap to the various type of logging, tracing, and alert facilities provided by DB2 and the operating systems it supports.

Facility	Platform	Notes	Refer to...
<b>DB2 Diagnostics</b>			
db2dart	All	<ul style="list-style-type: none"> <li>For servers</li> <li>Analyzes database objects for architectural integrity</li> </ul>	"Miscellaneous Troubleshooting Tools" on page 13
<b>DB2 Logging</b>			
db2diag.log file	All	<ul style="list-style-type: none"> <li>For clients and servers</li> <li>Logs errors and (depending on the diagnostic level) warnings and informational messages</li> </ul>	"Understanding First Failure Data Capture" on page 81
Alert file	All	<ul style="list-style-type: none"> <li>For clients and servers</li> <li>Contains severe DB2 errors</li> </ul>	"Alerts" on page 94
Dump files	All	<ul style="list-style-type: none"> <li>For clients and servers</li> <li>Additional information that is dumped for some errors</li> </ul>	"Dump Files" on page 90
Trap files	All	<ul style="list-style-type: none"> <li>For clients and servers</li> <li>Files that are created by DB2 in the event of a segmentation violation or trap</li> </ul>	"Trap Files" on page 91
DB2 message files	All	<ul style="list-style-type: none"> <li>For servers</li> <li>Logs activities during operations such as bind, load, import and export</li> </ul>	"Using Message Files" on page 93

DB2 Performance monitor	Windows NT, Windows 95, OS/2	<ul style="list-style-type: none"> <li>• For clients and servers</li> <li>• Logs deadlocks and other performance information</li> </ul>	"The Database System Monitor" on page 14, and the <i>System Monitor Guide and Reference</i>
Administrative tools log	Windows NT, Windows 95, OS/2	<ul style="list-style-type: none"> <li>• For clients and servers using the Control Center and other DB2 administrative tools</li> <li>• Logs information for the administrative tools</li> </ul>	"Using the Administrative Tools Log" on page 92
<b>Traces</b>			
<b>db2trc</b>	All	<ul style="list-style-type: none"> <li>• For clients and servers</li> <li>• Traces DB2 events</li> </ul>	"Using the DB2 Trace Facility (db2trc)" on page 97
Administrative tools trace	Windows NT, OS/2	<ul style="list-style-type: none"> <li>• For clients and servers using the Control Center and other DB2 administrative tools</li> <li>• Traces problems with the administrative tools</li> </ul>	"Administration Tools Trace" on page 103
<b>db2drdat</b>	All	<ul style="list-style-type: none"> <li>• Traces DRDA requesters and servers</li> </ul>	"Taking a DRDA Trace (db2drdat)" on page 106 and <i>DB2 Connect User's Guide</i>
<b>ddcstrc</b>	All	<ul style="list-style-type: none"> <li>• Traces DB2 Connect Enterprise Edition gateways</li> </ul>	<i>DB2 Connect User's Guide</i>
CLI trace	All	<ul style="list-style-type: none"> <li>• Traces CLI, JDBC, or ODBC applications</li> </ul>	"Taking a CLI/ODBC/JDBC Trace" on page 105
SNA trace	All	<ul style="list-style-type: none"> <li>• Traces communications via SNA</li> </ul>	"Taking an SNA Trace" on page 107
<b>Process Identification</b>			
<b>ps</b>	UNIX	<ul style="list-style-type: none"> <li>• For clients and servers</li> <li>• Shows current processes (along with process names for AIX)</li> </ul>	Chapter 8, "The DB2 Process Model" on page 75 and "Viewing Process Status Using the ps Command" on page 113



<b>pstat</b>	OS/2	<ul style="list-style-type: none"> <li>• For clients and servers</li> <li>• Shows processes and their threads</li> </ul>	Chapter 8, "The DB2 Process Model" on page 75 and "Using the pstat Command" on page 126
Windows NT Performance monitor	Windows NT	<ul style="list-style-type: none"> <li>• For clients and servers</li> <li>• Shows processes and their threads</li> </ul>	Chapter 8, "The DB2 Process Model" on page 75 and "Using Administrative Tools Provided by Windows NT" on page 130
<b>System Logging</b>			
System error log (syslog)	UNIX and OS/2	<ul style="list-style-type: none"> <li>• Logs alert conditions</li> <li>• For use by DB2 Customer Service</li> </ul>	"Using the System Error Log (syslog)" on page 109 and "Using the OS/2 Logging Facility" on page 120
Core file	UNIX	<ul style="list-style-type: none"> <li>• For UNIX-based systems</li> <li>• Logs severe errors</li> <li>• For use by DB2 Customer Service</li> </ul>	"Using the Core File" on page 112
UNIX commands	UNIX	<ul style="list-style-type: none"> <li>• Let you monitor and troubleshoot UNIX-based systems</li> </ul>	"Other Commands" on page 114
Event logger	Windows NT	<ul style="list-style-type: none"> <li>• Logs system errors, including severe DB2 errors for Windows NT systems</li> <li>• Accessible to remote system administrators</li> </ul>	"Diagnostic Tools for Windows NT" on page 129
SPM/2	OS/2	<ul style="list-style-type: none"> <li>• Monitors the performance of OS/2</li> </ul>	Chapter 12, "Diagnostic Tools for OS/2" on page 119
Administrative tools	Windows NT Version 4.0	<ul style="list-style-type: none"> <li>• Provide various administrative tools to diagnose and monitor Windows NT systems</li> </ul>	"Diagnostic Tools for Windows NT" on page 129
Dr. Watson	All Windows systems	<ul style="list-style-type: none"> <li>• Logs system errors during a General Protection Fault (GPF)</li> </ul>	"Diagnostic Tools for Windows NT" on page 129
Native OS/2 logging	OS/2	<ul style="list-style-type: none"> <li>• Logs error information on OS/2 programs</li> </ul>	"Using the OS/2 Logging Facility" on page 120

<b>Alerts</b>			
DB2 alert file	All	<ul style="list-style-type: none"> <li>• For clients and servers</li> <li>• Contains severe errors</li> </ul>	"Alerts" on page 94
SNA alerts	OS/2	<ul style="list-style-type: none"> <li>• Sends information over an SNA network in the event of a severe error</li> <li>• Can be generated by DB2 servers</li> </ul>	"Using the FFST for OS/2 Utility" on page 121
SNMP subagent	Windows NT, AIX, OS/2	<ul style="list-style-type: none"> <li>• Sends information over an SNMP network in the event of a severe error</li> <li>• Can be set up for DB2 servers</li> </ul>	"SNMP Alerts" on page 94

---

## Chapter 8. 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.

Unfortunately, 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.

In this chapter we use the terms "process/thread" to represent either processes or threads, and "process model" to represent the model for DB2 processes and/or 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 summarizes the DB2 process model.

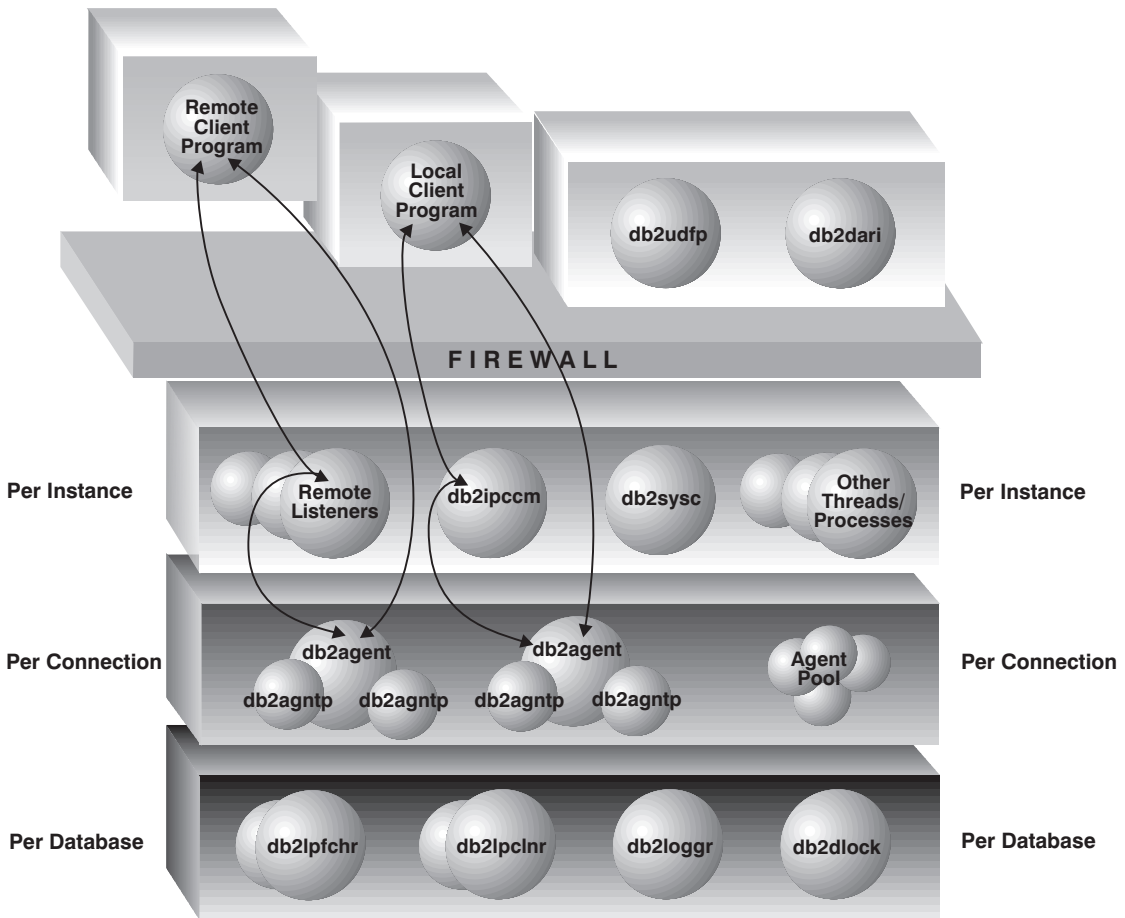


Figure 1. Process Model for DB2 Systems

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
- **db2tccpm**, 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. (See "Using the `db2diag.log` file to diagnose server communication problems" on page 32.)

## 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 symmetric multiprocessor (SMP) environments, partitioned database environments, and non-partitioned database environments in which the *intra\_parallel* database manager configuration parameter is enabled, the coordinator agent distributes the database requests to subagents (**db2agntp**), and 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 Extended Enterprise Edition)
- **db2pdbc**, the parallel system controller, which handles parallel requests from remote nodes (used only in DB2 Extended Enterprise Edition)
- **db2panic**, the panic agent, which handles urgent requests after agent limits have been reached at a particular node (used only in DB2 Extended Enterprise Edition)

### Differences Between Intel and UNIX

The Intel systems supported by DB2 (that is, 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 abnormal terminations. 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 “Understanding First Failure Data Capture” on page 81.)

For a listing of threads/processes, use the following:

- On UNIX-based systems: the **ps** command (see “Viewing Process Status Using the ps Command” on page 113)
- On OS/2: the **pstat** command (see “Using the pstat Command” on page 126)
- On Windows NT Version 4.0: the Task Manager (see “Using Administrative Tools Provided by Windows NT” on page 130)
- On Windows NT Version 3.5.1, and Windows 95: the Performance Monitor for the Windows operating system, with the Process and ID Process variables (see “Using DB2-Supplied Tools” on page 131)





---

## Chapter 9. Using 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.

---

### Understanding First Failure Data Capture

First Failure Data Capture (FFDC) information is diagnostic information captured automatically by DB2 about an error when the error occurs. This information reduces the need to reproduce errors to get diagnostic information.

The information captured by FFDC includes the following:

#### **db2diag.log**

When an error occurs, the db2diag.log file is updated with information about the error. For information, see “Interpreting the db2diag.log File” on page 83. Also see “Setting DIAGLEVEL and DIAGPATH” for information on the DIAGLEVEL setting, which determines how much information is logged in the file.

#### **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/native logging facility. See “Alerts” on page 94.

#### **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 90.

#### **Trap files**

The database manager generates a trap file if it cannot continue processing because of a trap, segmentation violation, or exception (depending on the platform). Trap files contain a function flow of the last steps that were executed before a problem occurred. See “Trap Files” on page 91.

This diagnostic information is contained in a single location (the *DIAGPATH* specified in the database manager configuration). It contains internal traces and dumps which usually can be read only by trained DB2 Customer Service personnel. However, this information can contain some crucial information to help you diagnose and fix problems.

### Setting DIAGLEVEL and DIAGPATH

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.

- `DIAGPATH` specifies the fully qualified path where DB2 writes its diagnostic information. This directory contains dump, error log, and alert log files, and is created automatically.

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 A, “Using the Problem Determination Tools Folder” on page 137 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, see the *Administration Guide*, the *Command Reference*, and the *API Reference*.

---

## 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 much error information, including pointers to dump files.

Most of the time, you can assume that 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, go to the first entry in the file marked with this time.
- The db2diag.log file **grows continuously**. When it gets too large, back it up if you need it 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 Entry Headers

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

---

```
1997-03-16-11.53.18.001160 1 Instance:payro11 2 Node:000 3
PID:44829(db2agent (SAMPLE)) 4 Appid:*LOCAL.payro11.970317140834 5
lock_manager 6 sqlpirq 7 Probe:111 8 Database:SAMPLE 9
DIA9999E 10 A 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 Extended Enterprise 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.
- 7** The name of the function that is providing the message. This function operates within the DB2 subcomponent that is writing the message.  
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>b</b>	Buffer pools
<b>c</b>	Communication between clients and servers
<b>d</b>	Data management
<b>e</b>	Engine processes
<b>o</b>	Operating system calls (such as opening and closing files)
<b>p</b>	Data protection (such as locking and logging)
<b>r</b>	Relational database services
<b>s</b>	Sorting
<b>x</b>	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 (see “Interpreting Hexadecimal Codes”).

## 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 B, “SQL Communication Area (SQLCA)” on page 143.

The following diagram provides an example of a db2diag.log file with an SQLCA dump, with labels indicating some important features.

---

```

1997-03-16-11.53.18.001160 Instance:payroll Node:000
PID:44829(db2agent (SAMPLE)) Appid:*LOCAL.payroll.970317140834
relation_data_serv sqlreri1g Probe:17 Database:SAMPLE
DIA9999E An internal return code occurred. Report the following : "0xFFFFE101".
Data Title :SQLCA pid(14358) 1
sqlcaid : SQLCA sqlcab: 136 sqlcode: -980 2 sqlerrml: 0
sqlerrmc: 3
sqlerrp : sqlrita
sqlerrd 4 : (1) 0xFFFFE101 5 (2) 0x00000000 (3) 0x00000000
(4) 0x00000000 (5) 0x00000000 (6) 0x00000000
sqlwarn : (1) (2) (3) (4) (5) (6)
(7) (8) (9) (10) (11)
sqlstate:

```

---

### Legend:

- 1** Beginning of the SQLCA entry.
- 2** The SQL state (when negative, an error has occurred).
- 3** Any reason codes associated with the SQL error code.
- 4** Sometimes there are several errors leading to the final SQL error code. These errors are shown in sequence in the sqlerrd area.
- 5** The hexadecimal representation of an SQL error (see “Interpreting Hexadecimal Codes”).

## Interpreting Hexadecimal Codes

On OS/2 and Windows systems, some db2diag.log or SQLCA error codes are byte-reversed. If they are in the form *ffff nnnn*, they can be interpreted as is. If they are in the form *nnnn ffff*, you must byte-reverse them before they are meaningful.

To do this, switch the first four characters with the last four characters, and then the fifth and sixth characters with the seventh and eighth characters.

For example, the error code "0ae6 ffff" is translated to "ffff e60a".

When you have the error code in a meaningful form:

1. Convert it to decimal format (using a hexadecimal conversion tool). If you can find 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. Refer to Appendix C, "DB2 Internal Return Codes" on page 147 for a list of important 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 turns out to be empty (that is, it contains no entries). Therefore, 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 sqlbrcres 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 sqlbrcres 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 (that is, a database page may not be consistent).
- 3** ZRC=FFFE119 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. This hexadecimal value indicates a table ID of 16.
  - 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 turns out to be empty (that is, it contains no entries). Therefore, 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. (See "Interpreting Hexadecimal Codes" on page 85.)

The error code in this example translates to ffff e60a. When translated to decimal form (-6646), it is not a valid SQL code, meaning that it is a return code rather than an SQL code. Looking up return code E60A in Appendix C, "DB2 Internal Return Codes" on page 147, we see that the file does not exist.

Database restart and recovery requires all log files that were in use at the time the database went down. This db2diag.log entry indicates that the log file S0000000.LOG cannot be found in the expected location. Restart and recovery cannot proceed.



- 4** Because restart and recovery could not complete, DB2 marks the database as "bad" to ensure that complete restart and recovery are done before the database is used again.

**Solution:** The best option to get around this problem is to restore from a backup that is known to be good. 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 (if backups exist).

### 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 look up 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 C, "DB2 Internal Return Codes" on page 147, we see that the disk is full.
- 2** The `sqlpgifl` function repeatedly encounters the error. This function is a logging function (because the fourth character is "p"; see "Interpreting the `db2diag.log` File" on page 83). 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 by using the following formula on the database configuration parameters for logging:

$$\text{logfilsiz} * 4 \text{ KB} * \text{logprimary} + \text{logfilsiz} * 4 \text{ KB} * \text{logsecond}$$

This calculation shows the maximum amount of space that may be required by the logs, assuming log retain is not enabled.

If log retain is enabled, the logs will continue to grow unless user exits are enabled. With log retain, ensure that the log path has as much space available as possible.

3. If you determine that the file system is full, do one of the following:
  - On UNIX-based environments, extend the size of the file system containing the logs.
  - Move the logs to another log path that has enough space available to contain the primary and secondary logs.
  - Examine your logging parameters to see if their values can be reduced (see the *Administration Guide*).

Sometimes the db2diag.log file does not contain enough information to solve a problem. Refer to “Example of Using db2diag.log and the DB2 Trace” on page 101 for an example of how to perform a trace when the db2diag.log file is not enough.

---

## 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 entries resemble the following:

---

```
1997-03-16-11.53.18.001160 Instance:payro11 Node:000
PID:44829(db2agent (SAMPLE)) Appid:*LOCAL.payro11.970317140834
relation_data_serv sqlrerrlg Probe:17 Database:SAMPLE
DIA9999E An internal return code occurred. Report the following : "0xFFFFE101".
Dump File: /home/db2/sqllib/db2dump/56772.000 Data : SECTION STMT 1
```

---

### Legend:

- 1** In this UNIX example, SECTION STMT data is stored in a file named 56772.000 located in the /home/db2/sqllib/db2dump directory.

### Notes:

- For DB2 Extended Enterprise 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/sql1lib/db2dump/56772.010 Data : SECTION STMT`
- For UNIX-based systems other than the SINIX operating system, dump files might be created in core dump directories. These files are called **DB2 core files**, and are specific to DB2 (rather than to the system). 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.
3. The stack traceback information is located in trap files. There is one file for each process. For DB2 Extended Enterprise 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 Extended Enterprise Edition:

- For systems with multiple nodes, use `db2_a11 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_a11 "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. You must erase or prune it occasionally if lack of disk space is a problem.
- You can specify where the log file is stored with the following command:

```
set DBTLOG_FILE=location
```

- If *location* is a file name with a fully qualified path, the file is stored at that location.
- If *location* contains a file name only, the file has that name, and is located in the directory specified by the `DB2INSTPROF` registry value. If this directory is read-only or does not exist, the file is located in the directory specified by the registry value `DB2PATH`.
- If you do not use the **set DBTLOG\_FILE** command, or do not specify a value for *location*, the file's name is `dbtcc.log`, and it is stored in the directory specified by the `DB2INSTPROF` or `DB2PATH` registry values (as described above).

For example, the following command will store the log file in d:\aelog.txt:

```
set DBTRACE_FILE=d:\aelog.txt
```

- If a log file cannot be opened for any reason, then an error message indicating this will appear on the screen.
- Occasionally, DB2 Customer Service may require more information than is in this log and request a DB2 tools trace. See “Administration Tools Trace” on page 103 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:

- |          |  |
|----------|--|
| <b>1</b> | A line indicating when the tool was started                    |
| <b>2</b> | The type of message (Informational, Warning, Error, or Severe) |
| <b>3</b> | A timestamp  |
| <b>4</b> | The process ID   |
| <b>5</b> | The thread ID  |
| <b>6</b> | An SQL code and message text (truncated in this example)       |
| <b>7</b> | 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*.

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 that:

- You can only view the contents of a message file after the operation is finished.
- For the load, import, and export utilities, a message file contains only those messages associated with the last operation.
- Each message in a message file begins on a new line and contains information provided by the DB2 message retrieval facility.

---

## Alerts

An **alert** is an error notification issued when a severe error occurs. It can be sent to a central machine, for the attention of network administrators.

Alerts are written to both the DB2 diagnostic log (db2diag.log) and to the alert log (db2alert.log). Like the db2diag.log 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 109.
- For OS/2, FFST/2 and SNA alerts. See “Using the FFST for OS/2 Utility” on page 121.
- For Windows NT, the event log. See “Diagnostic Tools for Windows NT” on page 129.

## 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**.
  4. 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**.
  4. 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 10. Taking Traces

This chapter describes various tracing methods you can use to capture information about a recurring and reproducible problem.

Traces are time-consuming because you must set them up, re-create your problem, and then collect the data. Typically you will use them 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 recreatable** scenario 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 have only one option: you must use the **db2trc** command. On other platforms, such as OS/2 or Windows NT, you can either use this command or initiate a trace through the Problem Determination Tools folder (see Appendix A, "Using the Problem Determination Tools Folder" on page 137).

#### Preliminary Considerations

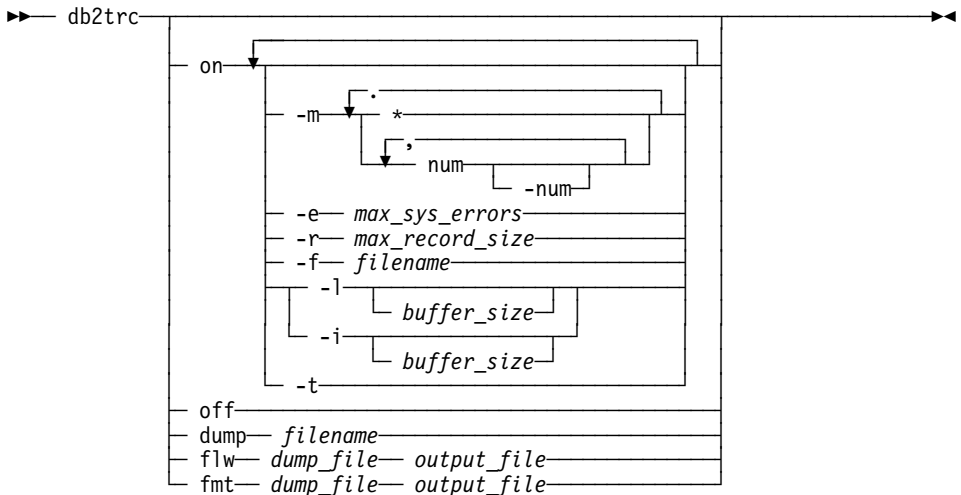
Before attempting to use the DB2 trace facility, read the following information:

- 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 Extended Enterprise Edition, you must run a trace for each node.
- You need one of SYSADM, SYSCTRL or SYSMANT 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 the next section 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 itself 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:

<code>-m *.*.*</code>	Trace everything
<code>-e -1</code>	Collect all errors
<code>-r 16000</code>	Maximum record size is 16 KB
<code>-s</code>	Trace destination to shared memory (instead of to a file)
<code>-l 2000000</code>	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:

<code>-m mask</code>	Specifies trace record types to focus the search. The <i>mask</i> 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.
<code>-e max_sys_errors</code>	Limits the number of DB2 internal system errors that the trace will hold to <i>max_sys_errors</i> . The default value is -1 (collect all errors).
<code>-r max_record_size</code>	Limits the size of trace records to <i>max_record_size</i> bytes. Longer trace records are truncated.
<code>-s   -f filename</code>	By default, trace output is stored in shared memory (the <code>-s</code> option). If the workstation is being suspended preventing you from accessing this output, you can trace to a file with the <code>-f</code> 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 101 for an example.
<code>-l [buffer_size]   -i [buffer_size]</code>	'-l' (lowercase "L") specifies that the last trace records are retained (that is, the first records are overwritten when the buffer is full). '-i' specifies that the initial trace records are retained (that is, 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 recreate the error situation.
- The error situation was recreated, 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 or vice versa.

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 -l -f db2trc.dmp
```
3. Reproduce the problem scenario.

The db2trc.dmp file will be present when you reboot your workstation. It will contain the events leading up to the event that caused the system to crash.

## Example of Using db2diag.log and the DB2 Trace

**Problem:** The SQL1042 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 information, see “Interpreting the db2diag.log File” on page 83.)

Assume that the db2diag.log file on the client machine turns out to be empty (that is, it contains no entries). Therefore, 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 C, "DB2 Internal Return Codes" on page 147, 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 sqllopenp (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 sqllopenp (1.25.15.140)
      pid 55; tid 38; cpid 112; time 365535; trace_point 7
      ffff ffff
3480  DB2 cei_errcode oper_system_services sqllopenp (1.6.15.140)
      pid 55; tid 38; cpid 112; time 365535; trace_point 254
      return_code = 0xffffe60a 1 = -6646 = SQL0_FNEX

```

---

### Interpretation:

- 1** Because trace file entries are ordered chronologically, we search backwards from the bottom of the trace file. We find the return code E60A in trace entry 3480.
- 2** Scanning backwards from trace entry 3480, we see that the file being opened was SQL00010.DAT in the table space SQLT0002.0

**Solution:** Restore from a backup known to be good, and roll forward to the end of the logs to ensure that no data is lost.

---

## Taking Other Traces

There are other traces you can perform to diagnose problems. This section discusses:

- “Administration Tools Trace”
- “Taking a Graphical Interface Trace” on page 104
- “Taking a CLI/ODBC/JDBC Trace” on page 105
- “Taking a DB2 Connect Trace (ddcstrc)” on page 106
- “Taking a DRDA Trace (db2drdat)” on page 106
- “Taking an SNA Trace” on page 107

### 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 104.)

This trace is only recommended for problems that can be re-created. It provides detailed information on requests, connections, attachments to instances, CLI 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 (see “Using the Administrative Tools Log” on page 92), except that more information is recorded.

Depending on how much information is written to it, the trace file can grow to the maximum file size imposed by the operating system. The file is overwritten the next time you perform an administration tools trace.

## Invoking an Administration Tools Trace

To invoke the administration tools trace facility:

1. Ensure that the tool is closed.
2. Set tracing on with the following command:  

```
set DBTRACE=Y
```
3. You can specify where the trace file will be stored with the following command:  

```
set DBTRACE_FILE=location
```

  - If *location* is a file name with a fully qualified path, the file will be created at that location.  
  
For example, the following command will send trace output to the file `d:\aetrace`:  

```
set DBTRACE_FILE=d:\aetrace
```
  - If *location* contains a file name only, the trace file will have that name, and be located in the directory specified by the `DB2INSTPROF` registry value. If this directory is read-only or does not exist, the file is created in the directory specified by the registry value `DB2PATH`.
  - If you do not use the **set DBTRACE\_FILE** command, or do not specify a value for *location*, the file's name is `dbttrace.trc`, and it is stored in the directory specified by the `DB2INSTPROF` or `DB2PATH` registry values (as described in the previous bullet).
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 ICUI 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 CLI/ODBC/JDBC Trace

The CLI trace can be used for any CLI, ODBC, or JDBC application that connects to DB2 using the CLI Driver.

To turn on this trace facility, set the db2cli.ini keyword option TRACE on (1). CLI trace records are appended to the file specified by the TRACEFILENAME keyword, or placed in the directory specified by the TRACEPATHNAME keyword.

For example, to set up a CLI trace file that is written to disk after each trace entry, include the following lines in the common section of the initialization file (DB2CLI.INI).

```
[COMMON]
TRACE=1
TRACEFILENAME=E:\TRACES\CLI\MONDAY.CLI
TRACEFLUSH=1
```

In the above example:

- The trace is activated by setting the value of TRACE to 1, and deactivated by setting it to 0.
- The trace output will be placed in the file E:\TRACES\CLIMONDAY.CLI  
Alternatively, you can use the keyword TRACEPATHNAME to store files in a particular subdirectory. For information, see the *Installing and Configuring DB2 Clients* online book.
- With the value of TRACEFLUSH set to 1, the trace will write files to disk every time they are updated. This value is recommended to avoid loss of data in the event of a crash.
- These settings will apply to all connections to DB2:

You can configure the DB2 CLI/ODBC driver further by using the DB2 Client Configuration Assistant tool:

1. Select the DB2 database alias you are configuring.
2. Select the **Settings** push button. The **Settings** window appears.
3. Select the **Settings** push button. The **ODBC Data Source Configuration** window appears.
4. Select the **Advanced** push button. A notebook appears where you can set the configuration keywords.

For more information on taking CLI traces, see the online help for ODBC data sources in the Client Configuration tool. For information on determining when to take the traces, see the *CLI Guide and Reference*.

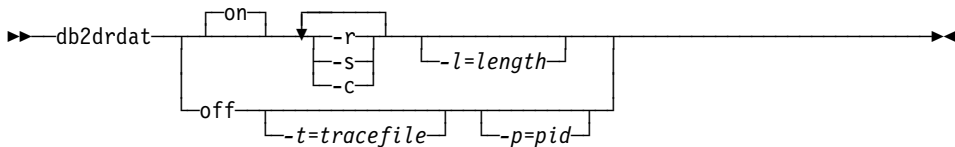
## Taking a DB2 Connect Trace (ddcstrc)

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

## Taking a DRDA Trace (db2drdat)

Use the DRDA trace (**db2drdat** command) to capture the DRDA data stream exchanged between a DRDA Application Requestor (AR) and the DRDA Application Server (AS). This tool is most often used for problem determination, but you can also use it for performance tuning in a client/server environment by determining how many sends and receives are required to execute an application. You need one of SYSADM, SYSCTRL, or 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.
- Note: Find out which process identifiers to trace from the *agent* field returned by the LIST APPLICATIONS command. For details, see the *Command Reference*.
- 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 97).
- 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: SNMP (see “SNMP Alerts” on page 94), and SNA (for communication with MVS/ESA, VM, VSE, or AS/400 systems).

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



---

## Chapter 11. 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 you or 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 112.)

### **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 113.)

### **Other UNIX-based commands**

UNIX-based systems provide various tools and utilities to diagnose problems. For a summary, see “Other Commands” on page 114.

---

## 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, except 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 “Understanding First Failure Data Capture” on page 81.

## Setting Up the System Error Log

To route alerts to the system error log, follow the steps listed below.

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 (you must have system authority to do this):

```
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 either:

- 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
```

Alternatively, for AIX you may wish to 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 bluejay DB2[46827]: DB2 (db2inst1.000 1) oper_system_services sqlobeep 2 reports: 3
Jun 18 15:02:53 4 bluejay 5 DB2[46827 6]: extra symptom string provided: 7 RIDS/sqlsysc_
Jun 18 15:02:53 bluejay DB2[46827]: data: 8 54686973 20697320 616e2065 78616d70
Jun 18 15:02:53 bluejay DB2[46827]: data: 6c65206f 66206c6f 67676564 20646174
Jun 18 15:02:53 bluejay DB2[46827]: data: 61
Jun 18 15:02:53 bluejay DB2[46827]: 2 piece(s) of dump data provided and dumped to file 9 /u/db2inst1/
Jun 18 15:02:53 bluejay DB2[46827]: 1. 'DUMP EXAMPLE #1' has been dumped 10
Jun 18 15:02:53 bluejay DB2[46827]: 2. 'DUMP EXAMPLE #2' has been dumped
```

---

### Legend:

- 1** The instance name and, for Extended Enterprise Edition 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 from a command prompt to get information about the reporting process in this example:  

```
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 90 for information on DB2 core files.)

## Accessing Core File Information

The **dbx** system command helps you determine which function caused a core file to be created. This is a simple check that will help you identify whether the database manager or DB2 Connect is in error, or whether an operating system or application error is responsible for the problem.

### Notes:

- To determine which function caused the core file dump to occur, enter the following command from a UNIX-based command prompt:

```
dbx program_name core_filename
```

where *program\_name* is the name of the program that terminated abnormally, and *core\_filename* is the name of the file containing the core file dump.

The *core\_filename* parameter is optional. If you do not specify it, the default name "core" is used.

- To obtain symbolic information, compile the application using the "-g" option.
- To end the **dbx** command, type **quit** at the dbx prompt.
- For the HP-UX operating system, use the **xdb** command for similar function.
- On Version 4.1 of AIX, ensure that the full core option has been enabled using the **chdev** command or **smitty**.
- The **dbx** command provides much more function than is described in this section. To find out more, enter `man dbx` from a UNIX-based command prompt.

## Example of the dbx Command

The following example shows how to use the **dbx** command to read the core file for a program called "main".

1. At a command prompt, enter:



```
dbx main
```

2. Output similar to the following appears on your display:

```
dbx version 3.1 for AIX.  
Type 'help' for help.  
reading symbolic information ...  
[using memory image in core]  
segmentation.violation in freeSegments at line 136  
136      (void) shmdt((void *) pcAddress[i]);  
(dbx) where  
freeSegments(numSegs = 2, iSetId = 0x2ff7f730, pcAddress = 0x2ff7f758, line  
136  
in "main.c"  
main (0x1, 2ff7f7d4), line 96 in "main.c"
```

In the above example, the name of the function that caused the core dump is "freeSegments". If the function name begins with "db2", "sql", or "ddcs", it may indicate an error in the database manager or DB2 Connect products.

The example also illustrates the use of the **where** subcommand. Enter **where** at the dbx prompt to display the program path to the point of failure. In this example, the error occurred at line 136 of freeSegments, which was called from line 96 in "main.c".

3. To end the **dbx** command, type **quit** at the dbx prompt.

---

## Viewing Process Status Using the ps Command

The **ps** command is a UNIX-based system command that returns process status information about active processes (to standard output). Use it to look for discrepancies between DB2 processes that are running and DB2 processes that you expect to be there. (For a description of DB2 processes, see Chapter 8, "The DB2 Process Model" on page 75.)

Flags control the types of information displayed for each active process, and may be applied simultaneously to yield a cumulative effect. For information, use the **man ps** command from a system command prompt.

**Example:** To show all processes of the instance ID "svtldb" use: `ps -fu svtldb`

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

---

```
svtldb 1 5112 2 27894 3 0 10:48:17 4 - 0:00 db2ipccm 5  
svtldb 6135 27894 0 10:48:17 - 0:00 db2gds  
svtldb 14329 27894 0 10:48:17 - 0:00 db2resyn  
svtldb 15356 27894 0 10:48:17 - 0:00 db2snacm 6  
svtldb 18682 27894 0 10:48:17 - 0:00 db2tccpm 6  
svtldb 27894 31989 0 10:48:17 - 0:00 db2sysc 7  
svtldb 33275 27894 0 10:48:17 - 0:00 db2tcpim 6  
svtldb 39939 18682 7 10:48:19 - 0:00 db2agent  
svtldb 56074 34761 2 10:48:37 pts/10 0:00 db2
```

---

### Legend:

- 1** The instance ID
- 2** The process identifier (pid)
- 3** The parent process identifier
- 4** The timestamp
- 5** The name of the process (see Chapter 8, “The DB2 Process Model” on page 75 for a description of their 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 engine 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.

## **Commands for DB2 Extended Enterprise Edition**

For DB2 Extended Enterprise 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 semi colon (;) improves performance by issuing commands simultaneously to all nodes. For information on other syntax for **db2\_all** and **rah**, see the file `rawREADME` in the `misc` subdirectory of the `sqliib` directory.

The following commands are also useful for DB2 Extended Enterprise 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 91.



---

## Chapter 12. Diagnostic Tools for OS/2

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 OS/2 system.

Once the data is collected, it can be examined by you or 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 120.

### **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. See "Using the FFST for OS/2 Utility" on page 121 and "Taking an SNA Trace" on page 107.

### **pstat**

The **pstat** command lists status information on which processes are running. See "Using the pstat Command" on page 126.

### **SPM/2**

The System Performance Monitor for OS/2 (SPM/2) is a separately orderable 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 information, see your Version 4 OS/2 Warp 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.





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. Country-specific date and time formats can be found in 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 (see “Using the OS/2 Logging Facility” on page 120 for information on how to view this file)
- 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**.

See “Configuring and Enabling the FFST for OS/2 Utility” for details on the FFST for OS/2 configuration parameters.

## 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:

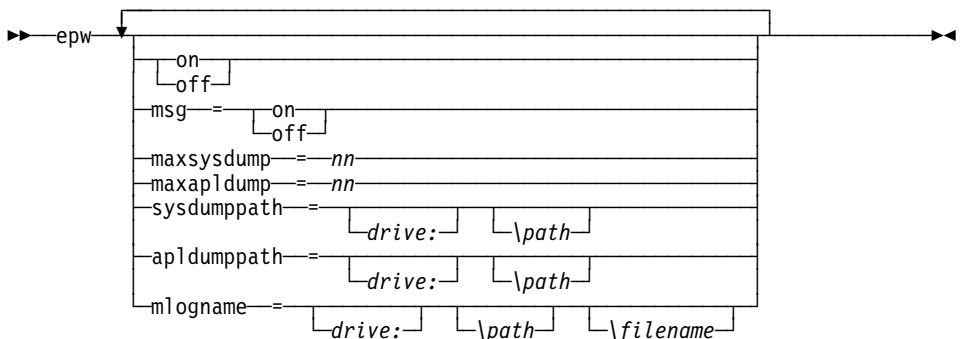
- Enable and disable FFST for OS/2
- Enable and disable the FFST for OS/2 message display
- Select the maximum number of dump data sets of a given type (either system dumps or application dumps) that may exist at one time in the current subdirectory for that type of dump
- Select the subdirectory in 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. See "Using OS/2 Logging Facility Commands" on page 120.
2. The **syslog** command impacts operations. Enter **syslog /s** at the OS/2 command prompt to suspend FFST for OS/2 error logging, and **syslog /r** to resume it.

## **MSG={ ON|OFF}**

Specifies whether FFST for OS/2 will display messages:

- |     |                            |
|-----|----------------------------|
| ON  | Displays messages          |
| OFF | Suppresses message display |

## **MAXSYSDUMP=*nn***

Specifies the maximum number of system software dumps that may occur in the current system dump directory. System software dump files are named OS2SYSxx.DMP, where *xx* ranges from 01 to *nn*.

### **Notes:**

1. When FFST for OS/2 is initialized, *xx* is initialized to the index of the newest dump file in the current dump directory plus one. Whenever *xx* exceeds *nn*, it is reset to 01. The value *nn* must be between 08 and 99. Its predefined default value is 32.
2. If the **epw** command is invoked in initialization mode and this keyword is not specified or is invalid, the default value of 32 will be used.
3. If the **epw** command is invoked in configuration mode and this keyword is not specified or is invalid, the current value remains unchanged.

## **MAXAPLDUMP=*nn***

Specifies the maximum number of software dumps that may occur in the current dump directory. System software dump files are named 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. Refer to "Using the OS/2 Logging Facility" on page 120 for information about the **syslog** command. (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/*cccccccc* LVLS/111  
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, and 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 epwrou -1**
- Run **detach epwrou -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.

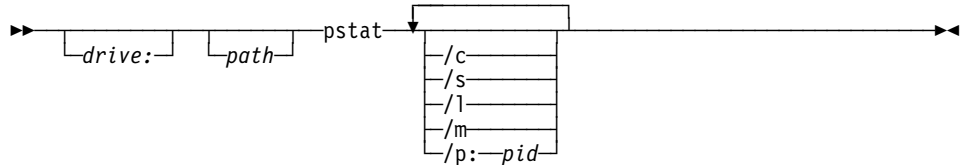
This command helps you determine:

- 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 8, "The DB2 Process Model" on page 75.

The **pstat** command displays the process ID that has been assigned for tracing on a per-process basis and that is used to get global system information.

Use the **pstat** command from the OS/2 command prompt. Its syntax is as follows:



### Parameters

If you use the **pstat** command without specifying any options, all process information will be displayed. If you want particular information only, use one or more of the following parameters:

**/C** This switch displays the current process- and thread-related information on the system. It returns the following information for each process:

- Process ID
- Parent process ID
- Session ID
- Process name

It returns the following information for each thread:

- Thread ID
- Thread priority
- Block ID
- Thread state (ready, blocked)

**/S** This switch displays system semaphore information for each thread on the system. It displays the following information for each thread:

- Process name
- Process ID
- Session ID
- Index
- Number of references
- Number of requests
- Flag
- System semaphore name

**/L** This switch displays the run-time libraries that are linked for each process on the system. It returns the following information for each process:

- Process name
- Process ID
- Session ID
- Library list

**/M** This switch displays the named shared-memory information for each process on the system:

- Handle
- Selector
- Number of references
- Shared memory name

**/P: *pid*** (where *pid* is the hexadecimal process ID to be displayed) This switch displays information related to the specified process. It returns the following information for the specified process ID:

- Process ID
- Parent process ID
- Session ID
- Process name
- Run-time link libraries
- Shared memory names

It returns the following information for each thread of the specified process ID:

- Thread ID
- Thread priority
- Thread state (ready, blocked)
- Block ID
- Owned semaphores

The following sample shows an example of **pstat** output:

---

Process and Thread Information									
Process ID	Parent Process ID	Session ID	Process Name	Thread ID	Priority	Block ID	State		
0279	000A	12	E:\SQLLIB\BIN\DB2SYSC.EXE			01	0200	FE11E6C4	Block
						02	0200	FDfE7870	Block
						03	0200	FE11E544	Block
						04	0200	FE1191D8	Block
						05	0200	FDf58ECC	Block
						06	0200	FE034B24	Block
						07	0200	FE11DA34	Block
						08	0200	FE11DA24	Block
						09	0200	FE11DA58	Block
						0A	0200	FE11DA14	Block

---



---

## Chapter 13. Diagnostic Tools for Windows Operating Systems

Use the diagnostic tools described in this chapter to gather and process data that can help identify the cause of a problem when using Windows Version 3.1, Windows 95, or Windows NT operating systems.

Once collected, the data can be examined by you or by someone who is familiar with the problem, or provided to DB2 Customer Service for analysis.

---

### Diagnostic Tools for Windows Version 3.1 and Windows 95

The following diagnostic tools are available for Windows Version 3.1 and Windows 95 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, system information on memory allocation, hardware, and devices is available from the **System** icon in the Control Panel.

#### **DB2-supplied tools**

For Windows 95 sessions, DB2 supplies administrative and development tools to help you identify DB2 problems. For example, use the Snapshot Monitor to identify bottlenecks. See "Using DB2-Supplied Tools" on page 131.

#### **DB2 traces**

DB2 traces provide some information on reproducible problems. See Chapter 10, "Taking Traces" on page 97.

#### **ODBC/CLI traces**

CLI traces help to identify problems in CLI and ODBC applications. See "Taking a CLI/ODBC/JDBC Trace" on page 105 for information.

---

### Diagnostic Tools for Windows NT

The following diagnostic tools are available for Windows NT systems:

#### **Event log, performance monitor, and other administrative tools**

NT Administrative Tools folder provides a variety of diagnostic information, including access to the event log and access to performance information. See "Using Administrative Tools Provided by Windows NT" on page 130 for information.

#### **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. For example, you can determine memory usage, memory limits, swapper space used, and memory leakage for a process.

To use it, 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 explicitly by typing `drwatson` on the command line.

#### **DB2-supplied tools**

DB2 supplies administrative and development tools to help you identify DB2 problems. See “Using DB2-Supplied Tools” on page 131.

#### **ODBC/CLI traces**

CLI traces help to identify problems in CLI and ODBC applications. See “Taking a CLI/ODBC/JDBC Trace” on page 105 for information.

#### **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. See “SNMP Alerts” on page 94.

## **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. See “Using the Windows NT Event Viewer.”
- Performance statistics for the local machine and for remote connections, select **Performance Monitor**. You can use this tool to find out and monitor information such as 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 remotely by system administrators or other personnel. Because this information can be

monitored remotely, there is a better chance that critical errors can be caught as quickly as possible.

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

An important feature provided by Windows 95 and Windows NT is the ability to take screen captures. If there is a window with information that you want to keep, do the following:

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. You can then send this file to someone who can help you.

---

## Using DB2-Supplied Tools

DB2 supplies administrative and development tools to help you identify DB2 problems on Windows NT and Windows 95 operating systems. For example:

- The DB2 database manager lets you identify remote connections, threads (with the Process and ID Process variables), and other performance-related information. See “The Database System Monitor” on page 14.
- The db2diag.log file identifies serious problems. (See “Understanding First Failure Data Capture” on page 81.)



---

## Chapter 14. Diagnostic Tools for Macintosh

Use the diagnostic tools described in this chapter to gather and process data that can help identify the cause of a problem when using Macintosh systems.

Once collected, the data can be examined by you or by someone who is familiar with the problem, or provided to IBM for analysis.

For Macintosh systems, the following tools are available:

### **DB2 trace**

In the DB2 Preferences alias file add the following:

```
db2traceon=y  
db2tracename=harddrive:db2:filename
```

You can format the trace by dragging and dropping the filename icon onto the **Trace formatter** icon.

For more information on DB2 traces, see "Using the DB2 Trace Facility (db2trc)" on page 97.

### **db2diag.log file**

See "Understanding First Failure Data Capture" on page 81 for more information on this important diagnostic tool.



---

## Part 3. Appendixes





---

## Appendix A. 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** menu choices.

**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\db2inet.htm
```

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

You may wish to create a bookmark in your Web browser to access this file.

---

### 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 pack.
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 pack you need. A list of all files associated with the fix pack 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 “Understanding First Failure Data Capture” on page 81.

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 B. SQL Communication Area (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 (see the *Road Map to DB2 Programming* for information on declaring SQLSTATE or SQLCODE variables in specific programming languages). 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 (see the *Road Map to DB2 Programming* for information on declaring the SQLCA 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 (Page 1 of 4). Fields of SQLCA

Name <sup>1</sup>	Data Type	Field values								
sqlcaid	CHAR(8)	An "eye catcher" for storage dumps containing 'SQLCA'.								
sqlcab	INTEGER	Contains the length of the SQLCA, 136.								
sqlcode	INTEGER	Contains the SQL return code. For specific meanings of SQL return codes, see the message section of the <i>Message Reference</i> . <table border="1"><thead><tr><th>Code</th><th>Means</th></tr></thead><tbody><tr><td>0</td><td>Successful execution (although one or more SQLWARN indicators may be set).</td></tr><tr><td>positive</td><td>Successful execution, but with a warning condition.</td></tr><tr><td>negative</td><td>Error condition.</td></tr></tbody></table>	Code	Means	0	Successful execution (although one or more SQLWARN indicators may be set).	positive	Successful execution, but with a warning condition.	negative	Error condition.
Code	Means									
0	Successful execution (although one or more SQLWARN indicators may be set).									
positive	Successful execution, but with a warning condition.									
negative	Error condition.									
sqlerrml	SMALLINT	Length indicator for <i>sqlerrmc</i> , in the range 0 through 70. 0 means that the value of <i>sqlerrmc</i> is not relevant.								

---

<sup>1</sup> The field names shown are those present in an SQLCA that is obtained via an INCLUDE statement.

Table 1 (Page 2 of 4). Fields of SQLCA

Name <sup>1</sup>	Data Type	Field values
sqlerrmc	VARCHAR (70)	<p>Contains one or more tokens, separated by X'FF', that are substituted for variables in the descriptions of error conditions.</p> <p>This field is also used when a successful connection is completed.</p> <p>When a NOT ATOMIC compound SQL statement is issued, it may contain information on up to 7 errors.</p> <p>For specific meanings of SQL return codes, see the message section of the <i>Message Reference</i>.</p>
sqlerrp	CHAR(8)	<p>Begins with a three-letter identifier indicating the product, followed by five digits indicating the version, release, and modification level of the product. For example, SQL05000 means DB2 Universal Database versions for Version 5 Release 0 Modification level 0.</p> <p>If SQLCODE indicates an error condition, then this field identifies the module that returned the error.</p> <p>This field is also used when a successful connection is completed.</p>
sqlerrd	ARRAY	<p>Six INTEGER variables that provide diagnostic information. These values are generally empty if there are no errors, except for sqlerrd(6) from a partitioned database.</p>
sqlerrd(1)	INTEGER	<p>If connection is invoked and successful, contains the maximum expected difference in length of mixed character data (CHAR data types) when converted to the database code page from the application code page. A value of 0 or 1 indicates no expansion; a value greater than 1 indicates a possible expansion in length; a negative value indicates a possible contraction. <sup>a</sup></p>
sqlerrd(2)	INTEGER	<p>If connection is invoked and successful, contains the maximum expected difference in length of mixed character data (CHAR data types) when converted to the application code page from the database code page. A value of 0 or 1 indicates no expansion; a value greater than 1 indicates a possible expansion in length; a negative value indicates a possible contraction. <sup>a</sup> If the SQLCA results from a NOT ATOMIC compound SQL statement that encountered one or more errors, the value is set to the number of statements that failed.</p>
sqlerrd(3)	INTEGER	<p>If PREPARE is invoked and successful, contains an estimate of the number of rows that will be returned. After INSERT, UPDATE, and DELETE, contains the actual number of rows affected. If compound SQL is invoked, contains an accumulation of all sub-statement rows. If CONNECT is invoked, contains 1 if the database can be updated; 2 if the database is read only.</p>
sqlerrd(4)	INTEGER	<p>If PREPARE is invoked and successful, contains a relative cost estimate of the resources required to process the statement. If compound SQL is invoked, contains a count of the number of successful sub-statements. If CONNECT is invoked, contains 0 for a one-phase commit from a down-level client; 1 for a one-phase commit; 2 for a one-phase, read-only commit; and 3 for a two-phase commit.</p>



Table 1 (Page 3 of 4). Fields of SQLCA

Name <sup>1</sup>	Data Type	Field values
sqlerrd(5)	INTEGER	<p>Contains the total number of rows deleted, inserted, or updated as a result of both:</p> <ul style="list-style-type: none"> <li>• The enforcement of constraints after a successful delete operation</li> <li>• The processing of triggered SQL statements from activated triggers.</li> </ul> <p>If compound SQL is invoked, contains an accumulation of the number of such rows for all substatements. In some cases when an error is encountered, this field contains a negative value that is an internal error pointer. If CONNECT is invoked, contains an authentication type value of 0 for a server authentication; 1 for client authentication; 2 for authentication using DB2 Connect; 3 for DCE security services authentication; 255 for unspecified authentication.</p>
sqlerrd(6)	INTEGER	<p>For a partitioned database, contains the partition number of the partition that encountered the error or warning. If no errors or warnings were encountered, this field contains the partition number of the coordinator node. The number in this field is the same as that specified for the partition in the db2nodes.cfg file.</p>
sqlwarn	Array	<p>A set of warning indicators, each containing a blank or W. If compound SQL is invoked, contains an accumulation of the warning indicators set for all substatements.</p>
sqlwarn0	CHAR(1)	<p>Blank if all other indicators are blank; contains W if at least one other indicator is not blank.</p>
sqlwarn1	CHAR(1)	<p>Contains W if the value of a string column was truncated when assigned to a host variable. Contains N if the null terminator was truncated.</p>
sqlwarn2	CHAR(1)	<p>Contains W if null values were eliminated from the argument of a function. <sup>b</sup></p>
sqlwarn3	CHAR(1)	<p>Contains W if the number of columns is not equal to the number of host variables.</p>
sqlwarn4	CHAR(1)	<p>Contains W if a prepared UPDATE or DELETE statement does not include a WHERE clause.</p>
sqlwarn5	CHAR(1)	<p>Reserved for future use.</p>
sqlwarn6	CHAR(1)	<p>Contains W if the result of a date calculation was adjusted to avoid an impossible date.</p>
sqlwarn7	CHAR(1)	<p>Reserved for future use.</p>
sqlwarn8	CHAR(1)	<p>Contains W if a character that could not be converted was replaced with a substitution character.</p>
sqlwarn9	CHAR(1)	<p>Contains W if arithmetic expressions with errors were ignored during column function processing.</p>
sqlwarn10	CHAR(1)	<p>Contains W if there was a conversion error when converting a character data value in one of the fields in the SQLCA.</p>
sqlstate	CHAR(5)	<p>A return code that indicates the outcome of the most recently executed SQL statement.</p>

Table 1 (Page 4 of 4). Fields of SQLCA

---

Name <sup>1</sup>	Data Type	Field values
-------------------	-----------	--------------

---

**Note:**

- a See the “Character Conversion Expansion Factor” section of the “Programming in Complex Environments” chapter in the *Embedded SQL Programming 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 Extended Enterprise 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 Extended Enterprise Edition Usage of the SQLCA

In DB2 Universal Database Extended Enterprise Edition, one SQL statement may be executed by a number of agents on different partitions, and each agent may return a different SQLCA for different errors or warnings. The coordinator agent also has its own SQLCA.

To provide a consistent view for applications, all SQLCA values are merged into one structure and SQLCA fields indicate global counts. For example:

- For all errors and warnings, the *sqlwarn* field contains the warning flags received from all agents.
- Values in the *sqlerrd* fields indicating row counts are accumulations from all agents.

Note that SQLSTATE 09000 may not be returned in all cases of an error occurring while processing a triggered SQL statement.

## Appendix C. 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, but is not exhaustive. 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 (see "Interpreting Hexadecimal Codes" on page 85).

Table 2 (Page 1 of 16). DB2 Internal Return Codes

Internal Return Code	SQL State	Reason Code	Description
81A7			Invalid alternate
811E	-996		Invalid user-specified directory
812F	-290		Access not allowed: table space is quiesced
8130	-290		Access not allowed: table space is load-pending
8131	-290		Access not allowed: table space is delete-pending
8132	-290		Access not allowed: table space is backup-pending
8133	-290		Access not allowed: table space is rollforward-pending
8134	-290		Access not allowed: table space is rollforward-in-progress
8135	-290		Access not allowed: table space is recovery-pending
8136	-290		Access not allowed: table space is disabled
8137	-290		Access not allowed
8138	-291		Invalid state transition
8139	-294		Container is already being used
813A	-295		Container names are too long
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

Table 2 (Page 2 of 16). DB2 Internal Return Codes

Internal Return Code	SQL State	Reason Code	Description
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
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

Table 2 (Page 3 of 16). DB2 Internal Return Codes

Internal Return Code	SQL State	Reason Code	Description
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
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)

Table 2 (Page 4 of 16). DB2 Internal Return Codes

Internal Return Code	SQL State	Reason Code	Description
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
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

Table 2 (Page 5 of 16). DB2 Internal Return Codes

Internal Return Code	SQL State	Reason Code	Description
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
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

Table 2 (Page 6 of 16). DB2 Internal Return Codes

<b>Internal Return Code</b>	<b>SQL State</b>	<b>Reason Code</b>	<b>Description</b>
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
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



Table 2 (Page 7 of 16). DB2 Internal Return Codes

<b>Internal Return Code</b>	<b>SQL State</b>	<b>Reason Code</b>	<b>Description</b>
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
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

Table 2 (Page 8 of 16). DB2 Internal Return Codes

Internal Return Code	SQL State	Reason Code	Description
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
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

Table 2 (Page 9 of 16). DB2 Internal Return Codes

<b>Internal Return Code</b>	<b>SQL State</b>	<b>Reason Code</b>	<b>Description</b>
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
D107	-960		No more file tokens
D121	-289		Container full
D122			Exceeded maximum quiescers
D123	-296		Table space limit exceeded
D124	-259		Map too big
D20C	-968		Disk full
D21A	-930		No memory: UNDO heap
D315			No memory on requester
D316			No memory on server
D31D			Vectored I/O request too big
D408	-962		Maximum tables in file
D40D	-912		Too many locks
D411	-902		Maximum long field file size
D505	-930		Memory allocation failure
D509	-964		Log file full
D50A	-912		Lock list full
D57F	-1004		Disk full log file
D601	-954		No memory heap (for application)
D602	-956		No memory heap (for database)
D603	-101		No memory heap
D604	-930		Share buffer exceeded
D605	-930		Memory allocation failure
D606	-958		Too many open files
D607	-960		No more file tokens
D60B	-955		No memory sort heap
D60C	-968		Disk full

Table 2 (Page 10 of 16). DB2 Internal Return Codes

Internal Return Code	SQL State	Reason Code	Description
D60D	-912		Too many locks
D60E	-973		No memory
D610	-902		No memory BSU heap
D612	-953		No memory: AgentHeap
D613	-957, -959		No memory: ComHeap
D614	-961, -962		No memory: RSHeap
D615			No memory: Users/groups
D616	-930		No memory: DrIdx heap
D617	-930		No memory: ASP heap
D619	-930		No memory: Queue heap
D61A	-930		No memory: UND heap
D61B	-930		No memory: Lock heap
D61C	-930		No memory: System heap
D61D	-930		No memory
D61E	-930		No memory
D620	-930		No memory
D625	-930		No memory
D62B	-930		No memory
D67D			Shared memory set exists
D67E	-9		No memory for DosLoadMod
D905	-930		Memory allocation failure
D90F	-990		Index structure problem
DC26	-6042		No FCM MSG_ANCHOR
DC27	-6040		No FCM buffer
DC28	-6041		No FCM connection entry
DC29	-6043		No FCM request block
DC2A	-902		No high priority buffer
E101	-980	1	Bad page
E10A	-980	10	File does not exist
E10E	-982		Bad signature
E119	-1034		Page CHECKSUM error
E11A	-1035		Bad database, won't flush it
E11B	-980	27	Both primary and shadow ORFs are bad
E11C	-980	28	Primary ORF is bad

Table 2 (Page 11 of 16). DB2 Internal Return Codes

Internal Return Code	SQL State	Reason Code	Description
E11D	-980	29	Secondary ORF is bad
E11E	-980	30	Both primary and shadow SSFs are bad
E11F	-980	31	Primary SSF bad
E120	-980	32	Secondary SSF bad
E40B	-980	11	Object does not exist
E50D	-980	13	Bad log file
E510	-5123		Error in log control file
E511	-1258		Log control file not found
E512	-1259		I/O error accessing log control file
E513	-1260		Database not recoverable
E514	-1261		Recovery not pending
E515	-1263		Invalid log extent file
E516	-1264		Log extent file does not belong to the database
E517	-1265		Log extent file is the incorrect version
E518	-1266		Point in time prior to recovery
E521	-1034		Recovery failed
E522	-1269		Error while retrieving file during forward recovery
E57F	-1036		Adjust log file error
E602	-980	2	CRC error
E603	-980	3	Disk error
E604	-980	4	General failure
E605	-980	5	Read fault
E606	-980	6	Seek error
E607	-980	7	Sector not found
E608	-980	8	Unknown media error
E609	-980	9	Write fault
E60A	-980	10	File does not exist
E60C	-980	12	Cannot open file
E60F	-980	15	Network access denied
E623	-931		Too many open system files
F051	-1042		Invalid log record encountered during redo or undo: unknown component
F102	-902	2	BPS logic error

Table 2 (Page 12 of 16). DB2 Internal Return Codes

Internal Return Code	SQL State	Reason Code	Description
F103	-902	3	Invalid buffer pointer
F104	-902	4	No buffers
F109	-902	9	Data does not exist
F10A	-902	10	File already exists
F10B	-902	11	Unfixed buffer page
F10C	-902	12	Invalid file token
F10D	-902	13	Invalid file type
F110	-902	16	Invalid mode parameter
F117	-902	23	Invalid reference
F121	-902	33	RAM semaphore error
F124	-902	36	Bad configuration file
F136	-902	54	Bad header
F13B	-902	59	File not found in the reorg linked list
F149	-902	73	Bad database path
F156	-902	86	Page already exists
F176	-902	118	Invalid pool ID
F17D			State already off
F210	-902	16	Invalid mode parameter
F21B	-902	27	Bad transaction ID
F225	-902	37	Invalid file handle
F331	-902	49	Bad selector
F33C	-902	60	Tokenizer stack overflow
F345			The server STARTDBM failed
F34A	-902	74	Node manager thread failed
F34B	-902	75	Database is bad
F34C	-902	76	Parser stack overflow
F34D	-902	77	Token buffer overflow
F34E	-902	78	Bad value in code page table
F34F	-902	79	Configuration mismatch
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

Table 2 (Page 13 of 16). DB2 Internal Return Codes

<b>Internal Return Code</b>	<b>SQL State</b>	<b>Reason Code</b>	<b>Description</b>
F42F	-902	47	Invalid mode
F432	-902	50	Floating point error
F451	-1042		Invalid log record encountered during redo or undo: bad record length
F455	-902	85	Program error
F47A	-902	122	Record deleted
F47C	-902	124	Missing defaults
F51B	-902	27	Bad transaction ID
F51C	-902	28	Log file overflow
F51D	-902	29	Fatal logic error
F51E	-902	30	No active transaction
F51F	-902	31	Maximum save points
F520	-902	32	No active save point
F527	-902	39	Bad record type
F528	-902	40	Transaction ID table overflow
F529	-902	41	Invalid LSN
F52A	-902	42	Transaction already started
F550	-902	80	Database in recovery mode
F605	-1068		Message file not found
F606	-1068		Message not found
F609	-902	9	Data does not exist
F60A	-902	10	File already exists
F60B	-902	11	Unfixed buffer page
F60C	-902	12	Invalid file token
F60D	-902	13	Invalid file type
F60E	-902	14	Lock violation
F60F	-902	15	Directory overflow
F610	-902	16	Invalid mode parameter
F611	-902	17	Invalid path
F612	-902	18	Invalid page number
F613	-902	19	Sector boundary error
F614	-902	20	System internal error
F616	-902	22	File sharing error
F617	-902	23	Invalid reference
F61A	-902	26	Invalid selector

Table 2 (Page 14 of 16). DB2 Internal Return Codes

<b>Internal Return Code</b>	<b>SQL State</b>	<b>Reason Code</b>	<b>Description</b>
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
F867	-902	103	BSEG size/address conflict
F868	-902	104	BSEG not free or allocated
F869	-902	105	Incorrect BSEG size



Table 2 (Page 15 of 16). DB2 Internal Return Codes

Internal Return Code	SQL State	Reason Code	Description
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
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

*Table 2 (Page 16 of 16). DB2 Internal Return Codes*

<b>Internal Return Code</b>	<b>SQL State</b>	<b>Reason Code</b>	<b>Description</b>
FD8E			PMAP is of a 1-node nodegroup

---

## Appendix D. How the DB2 Library Is Structured

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

To help you access product information online, DB2 provides the Information Center on OS/2, Windows 95, and the Windows NT operating systems. You can view task information, DB2 books, troubleshooting information, sample programs, and DB2 information on the Web. "About the Information Center" on page 170 has more details.

---

### SmartGuides

SmartGuides help you complete some administration tasks by taking you through each task one step at a time. SmartGuides are available on OS/2, Windows 95, and the Windows NT operating systems. The following table lists the SmartGuides.

<b>SmartGuide</b>	<b>Helps you to...</b>	<b>How to Access...</b>
<i>Add Database</i>	Catalog a database on a client workstation.	From the Client Configuration Assistant, click on <b>Add</b> .
<i>Create Database</i>	Create a database, and to perform some basic configuration tasks.	From the Control Center, click with the right mouse button on the <b>Databases</b> icon and select <b>Create-&gt;New</b> .
<i>Performance Configuration</i>	Tune the performance of a database by updating configuration parameters to match your business requirements.	From the Control Center, click with the right mouse button on the database you want to tune and select <b>Configure performance</b> .
<i>Backup Database</i>	Determine, create, and schedule a backup plan.	From the Control Center, click with the right mouse button on the database you want to backup and select <b>Backup-&gt;Database using SmartGuide</b> .
<i>Restore Database</i>	Recover a database after a failure. It helps you understand which backup to use, and which logs to replay.	From the Control Center, click with the right mouse button on the database you want to restore and select <b>Restore-&gt;Database using SmartGuide</b> .
<i>Create Table</i>	Select basic data types, and create a primary key for the table.	From the Control Center, click with the right mouse button on the <b>Tables</b> icon and select <b>Create-&gt;Table using SmartGuide</b> .
<i>Create Table Space</i>	Create a new table space.	From the Control Center, click with the right mouse button on the <b>Table spaces</b> icon and select <b>Create-&gt;Table space using SmartGuide</b> .

---

## Online Help

Online help is available with all DB2 components. The following table describes the various types of help.

Type of Help	Contents	How to Access...
<i>Command Help</i>	Explains the syntax of commands in the command line processor.	From the command line processor in interactive mode, enter:  <i>? command</i>  where <i>command</i> is a keyword or the entire command.  For example, <i>? catalog</i> displays help for all the CATALOG commands, whereas <i>? catalog database</i> displays help for the CATALOG DATABASE command.
<i>Control Center Help</i>	Explains the tasks you can perform in a window or notebook. The help includes prerequisite information you need to know, and describes how to use the window or notebook controls.	From a window or notebook, click on the <b>Help</b> push button or press the F1 key.
<i>Message Help</i>	Describes the cause of a message number, and any action you should take.	From the command line processor in interactive mode, enter:  <i>? message number</i>  where <i>message number</i> is a valid message number.  For example, <i>? SQL30081</i> displays help about the SQL30081 message.  To view message help one screen at a time, enter:  <i>? XXXnnnnn   more</i>  where <i>XXX</i> is the message prefix, such as SQL, and <i>nnnnn</i> is the message number, such as 30081.  To save message help in a file, enter:  <i>? XXXnnnnn &gt; filename.ext</i>  where <i>filename.ext</i> is the file where you want to save the message help.  <b>Note:</b> On UNIX-based systems, enter:  <i>\? XXXnnnnn   more</i> or  <i>\? XXXnnnnn &gt; filename.ext</i>

Type of Help	Contents	How to Access...
<i>SQL Help</i>	Explains the syntax of SQL statements.	<p>From the command line processor in interactive mode, enter:</p> <p><b>help</b> <i>statement</i></p> <p>where <i>statement</i> is an SQL statement.</p> <p>For example, <b>help</b> <i>SELECT</i> displays help about the SELECT statement.</p>
<i>SQLSTATE Help</i>	Explains SQL states and class codes.	<p>From the command line processor in interactive mode, enter:</p> <p><b>? <i>sqlstate</i></b> or <b>? <i>class-code</i></b></p> <p>where <i>sqlstate</i> is a valid five digit SQL state and <i>class-code</i> is a valid two digit class code.</p> <p>For example, <b>? 08003</b> displays help for the 08003 SQL state, whereas <b>? 08</b> displays help for the 08 class code.</p>

---

## DB2 Books

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

- Cross-platform books: These books are for DB2 on any of the supported platforms.
- Platform-specific books: These books are for DB2 on a specific platform. For example, there is a separate *Quick Beginnings* book for DB2 on OS/2, Windows NT, and UNIX-based operating systems.

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

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

- View** To view an HTML book, you can do the following:
- If you are running DB2 administration tools on OS/2, Windows 95, or the Windows NT operating systems, you can use the Information Center. “About the Information Center” on page 170 has more details.
  - Use the open file function of the Web browser supplied by DB2 (or one of your own) to open the following page:  

```
sqlib/doc/html/index.htm
```

The page contains descriptions of and links to the DB2 books. The path is located on the drive where DB2 is installed.

You can also open the page by double-clicking on the **DB2 Online Books** icon. Depending on the system you are using, the icon is in the main product folder or the Windows Start menu.
- Search** To search for information in the HTML books, you can do the following:
- Click on **Search the DB2 Books** at the bottom of any page in the HTML books. Use the search form to find a specific topic.
  - Click on **Index** at the bottom of any page in an HTML book. Use the Index to find a specific topic in the book.
  - Display the Table of Contents or Index of the HTML book, and then use the find function of the Web browser to find a specific topic in the book.
  - Use the bookmark function of the Web browser to quickly return to a specific topic.
  - Use the search function of the Information Center to find specific topics. “About the Information Center” on page 170 has more details.
- Print** To print a book on a PostScript printer, look for the file name shown in the table.
- Order** To order a hardcopy book from IBM, use the form number.

<b>Book Name</b>	<b>Book Description</b>	<b>Form Number File Name</b>
<b>Cross-Platform Books</b>		
<i>Administration Getting Started</i>	Introduces basic DB2 database administration concepts and tasks, and walks you through the primary administrative tasks.	S10J-8154 db2k0x50
<i>Administration Guide</i>	Contains information required to design, implement, and maintain a database to be accessed either locally or in a client/server environment.	S10J-8157 db2d0x50
<i>API Reference</i>	Describes the DB2 application programming interfaces (APIs) and data structures you can use to manage your databases. Explains how to call APIs from your applications.	S10J-8167 db2b0x50
<i>CLI Guide and Reference</i>	Explains how to develop applications that access DB2 databases using the DB2 Call Level Interface, a callable SQL interface that is compatible with the Microsoft ODBC specification.	S10J-8159 db2l0x50
<i>Command Reference</i>	Explains how to use the command line processor, and describes the DB2 commands you can use to manage your database.	S10J-8166 db2n0x50
<i>DB2 Connect Enterprise Edition Quick Beginnings</i>	Provides planning, installing, configuring, and using information for DB2 Connect Enterprise Edition. Also contains installation and setup information for all supported clients.	S10J-7888 db2cyx50
<i>DB2 Connect Personal Edition Quick Beginnings</i>	Provides planning, installing, configuring, and using information for DB2 Connect Personal Edition.	S10J-8162 db2c1x50
<i>DB2 Connect User's Guide</i>	Provides concepts, programming and general using information about the DB2 Connect products.	S10J-8163 db2c0x50
<i>DB2 Connectivity Supplement</i>	Provides setup and reference information for customers who want to use DB2 for AS/400, DB2 for OS/390, DB2 for MVS, or DB2 for VM as DRDA Application Requesters with DB2 Universal Database servers, and customers who want to use DRDA Application Servers with DB2 Connect (formerly DDCS) application requesters.  <b>Note:</b> Available in HTML and PostScript formats only.	No form number db2h1x50
<i>Embedded SQL Programming Guide</i>	Explains how to develop applications that access DB2 databases using embedded SQL, and includes discussions about programming techniques and performance considerations.	S10J-8158 db2a0x50
<i>Glossary</i>	Provides a comprehensive list of all DB2 terms and definitions.  <b>Note:</b> Available in HTML format only.	No form number db2t0x50

<b>Book Name</b>	<b>Book Description</b>	<b>Form Number</b> <b>File Name</b>
<i>Installing and Configuring DB2 Clients</i>	Provides installation and setup information for all DB2 Client Application Enablers and DB2 Software Developer's Kits.  <b>Note:</b> Available in HTML and PostScript formats only.	No form number db2iyx50
<i>Master Index</i>	Contains a cross reference to the major topics covered in the DB2 library.  <b>Note:</b> Available in PostScript format and hardcopy only.	S10J-8170 db2w0x50
<i>Message Reference</i>	Lists messages and codes issued by DB2, and describes the actions you should take.	S10J-8168 db2m0x50
<i>Replication Guide and Reference</i>	Provides planning, configuring, administering, and using information for the IBM Replication tools supplied with DB2.	S95H-0999 db2e0x50
<i>Road Map to DB2 Programming</i>	Introduces the different ways your applications can access DB2, describes key DB2 features you can use in your applications, and points to detailed sources of information for DB2 programming.	S10J-8155 db2u0x50
<i>SQL Getting Started</i>	Introduces SQL concepts, and provides examples for many constructs and tasks.	S10J-8156 db2y0x50
<i>SQL Reference</i>	Describes SQL syntax, semantics, and the rules of the language. Also includes information about release-to-release incompatibilities, product limits, and catalog views.	S10J-8165 db2s0x50
<i>System Monitor Guide and Reference</i>	Describes how to collect different kinds of information about your database and the database manager. Explains how you can use the information to understand database activity, improve performance, and determine the cause of problems.	S10J-8164 db2f0x50
<i>Troubleshooting Guide</i>	Helps you determine the source of errors, recover from problems, and use diagnostic tools in consultation with DB2 Customer Service.	S10J-8169 db2p0x50
<i>What's New</i>	Describes the new features, functions, and enhancements in DB2 Universal Database.  <b>Note:</b> Available in HTML and PostScript formats only.	No form number db2q0x50
<b>Platform-Specific Books</b>		
<i>Building Applications for UNIX Environments</i>	Provides environment setup information and step-by-step instructions to compile, link, and run DB2 applications on a UNIX system.	S10J-8161 db2axx50
<i>Building Applications for Windows and OS/2 Environments</i>	Provides environment setup information and step-by-step instructions to compile, link, and run DB2 applications on a Windows or OS/2 system.	S10J-8160 db2a1x50



<b>Book Name</b>	<b>Book Description</b>	<b>Form Number</b> <b>File Name</b>
<i>DB2 Extended Enterprise Edition Quick Beginnings</i>	Provides planning, installing, configuring, and using information for DB2 Universal Database Extended Enterprise Edition for AIX.	S72H-9620 db2v3x50
<i>DB2 Personal Edition Quick Beginnings</i>	Provides planning, installing, configuring, and using information for DB2 Universal Database Personal Edition on OS/2, Windows 95, and the Windows NT operating systems.	S10J-8150 db2i1x50
<i>DB2 SDK for Macintosh Building Your Applications</i>	Provides environment setup information and step-by-step instructions to compile, link, and run DB2 applications on a Macintosh system.  <b>Note:</b> Available in PostScript format and hardcopy for DB2 Version 2.1.2 only.	S50H-0528 sqla7x02
<i>DB2 SDK for SCO OpenServer Building Your Applications</i>	Provides environment setup information and step-by-step instructions to compile, link, and run DB2 applications on a SCO OpenServer system.  <b>Note:</b> Available for DB2 Version 2.1.2 only.	S89H-3242 sqla9x02
<i>DB2 SDK for Silicon Graphics IRIX Building Your Applications</i>	Provides environment setup information and step-by-step instructions to compile, link, and run DB2 applications on a Silicon Graphics system.  <b>Note:</b> Available in PostScript format and hardcopy for DB2 Version 2.1.2 only.	S89H-4032 sqlaax02
<i>DB2 SDK for SINIX Building Your Applications</i>	Provides environment setup information and step-by-step instructions to compile, link, and run DB2 applications on a SINIX system.  <b>Note:</b> Available in PostScript format and hardcopy for DB2 Version 2.1.2 only.	S50H-0530 sqla8x00
<i>Quick Beginnings for OS/2</i>	Provides planning, installing, configuring, and using information for DB2 Universal Database on OS/2. Also contains installing and setup information for all supported clients.	S10J-8147 db2i2x50
<i>Quick Beginnings for UNIX</i>	Provides planning, installing, configuring, and using information for DB2 Universal Database on UNIX-based platforms. Also contains installing and setup information for all supported clients.	S10J-8148 db2ixx50
<i>Quick Beginnings for Windows NT</i>	Provides planning, installing, configuring, and using information for DB2 Universal Database on the Windows NT operating system. Also contains installing and setup information for all supported clients.	S10J-8149 db2i6x50

**Notes:**

1. The character in the sixth position of the file name indicates the language of a book. For example, the file name db2d0e50 indicates that the *Administration Guide* is in English. The following letters are used in the file names to indicate the language of a book:

<b>Language</b>	<b>Identifier</b>	<b>Language</b>	<b>Identifier</b>
Brazilian Portuguese	B	Hungarian	H
Bulgarian	U	Italian	I
Czech	X	Norwegian	N
Danish	D	Polish	P
English	E	Russian	R
Finnish	Y	Slovenian	L
French	F	Spanish	Z
German	G	Swedish	S

2. For late breaking information that could not be included in the DB2 books, see the README file. Each DB2 product includes a README file which you can find in the directory where the product is installed.

---

## About the Information Center

The Information Center provides quick access to DB2 product information. The Information Center is available on OS/2, Windows 95, and the Windows NT operating systems. You must install the DB2 administration tools to see the Information Center.

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

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

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

<b>Tasks</b>	Lists tasks you can perform using DB2.
<b>Reference</b>	Lists DB2 reference information, such as keywords, commands, and APIs.
<b>Books</b>	Lists DB2 books.
<b>Troubleshooting</b>	Lists categories of error messages and their recovery actions.
<b>Sample Programs</b>	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.
<b>Web</b>	Lists DB2 information on the World Wide Web. To access this information, you must have a connection to the Web from your system.

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

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



## Appendix E. How to Read Syntax Diagrams

In this book, syntax for some commands is presented in syntax diagrams. This section shows some of the structures contained in these diagrams.

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

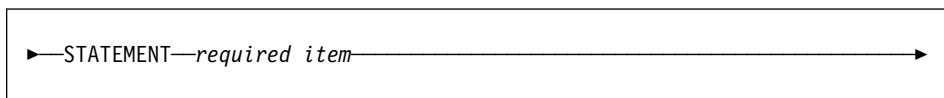
The  $\blacktriangleright$ — symbol indicates the beginning of a statement.

The — $\rightarrow$  symbol indicates that the statement syntax is continued on the next line.

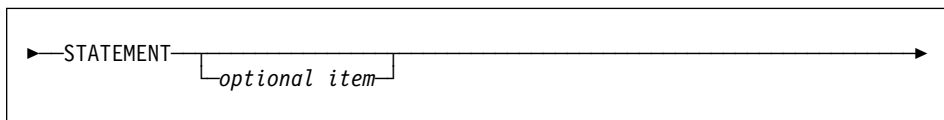
The  $\blacktriangleright$ — symbol indicates that a statement is continued from the previous line.

The — $\blacktriangleright$  symbol indicates the end of a statement.

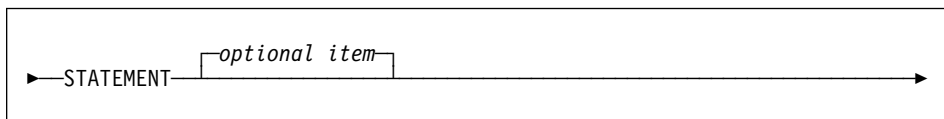
Required items appear on the horizontal line (the main path).



Optional items appear below the main path.

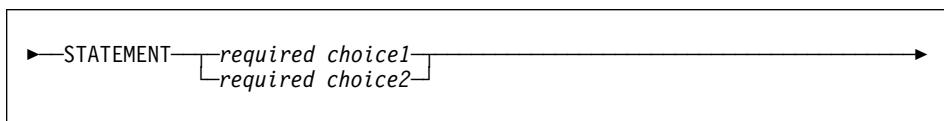


If an optional item appears above the main path, it can be added for improved readability of the statement but it has no effect on the statement's execution.

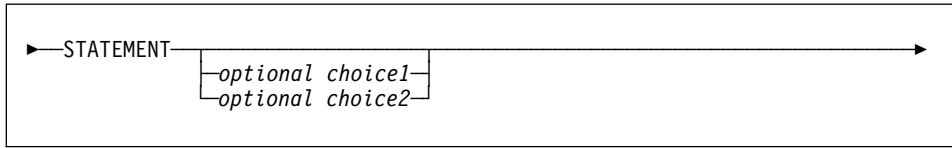


If you can choose from two or more items, they appear in a stack.

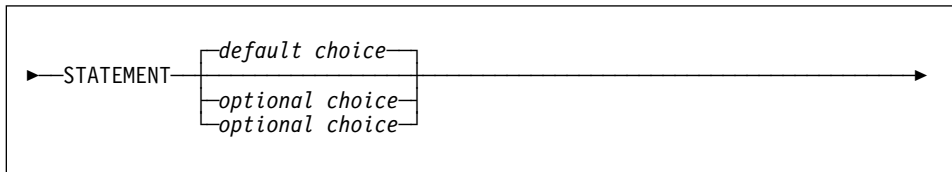
If you *must* choose one of the items, one item of the stack appears on the main path.



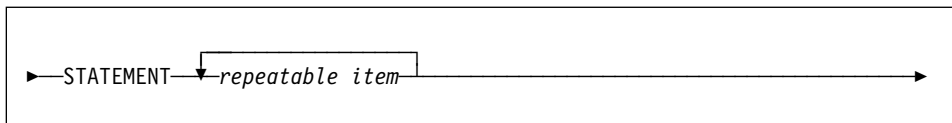
If choosing none of the items is an option, the entire stack appears below the main path.



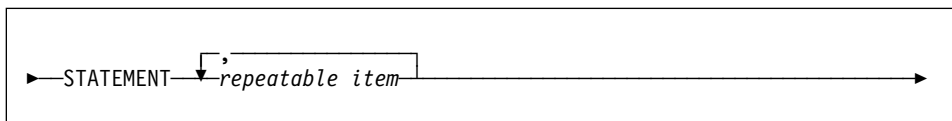
If one of the items is the default, it will appear above the main path and the remaining choices will be shown below.



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, CREATE TABLE). 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.

---

## Appendix F. Notices

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

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

IBM Director of Licensing,  
IBM Corporation,  
500 Columbus Avenue,  
Thornwood, NY, 10594  
USA.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Canada Limited  
Department 071  
1150 Eglinton Ave. East  
North York, Ontario  
M3C 1H7  
CANADA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

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

---

## Trademarks

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

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

---

## Trademarks of Other Companies

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

C-bus is a trademark of Corollary, Inc.

HP-UX is a trademark of Hewlett-Packard.

Java and HotJava are trademarks of Sun Microsystems, Inc.

Microsoft, Windows, Windows NT, and the Windows 95 logo are trademarks or registered trademarks of Microsoft Corporation.

PC Direct is a trademark of Ziff Communications Company and is used by IBM Corporation under license.

SINIX is a trademark of Siemens Nixdorf.

Solaris is a trademark of Sun Microsystems, Inc.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Other company, product, or service names, which may be denoted by a double asterisk (\*\*), may be trademarks or service marks of others.



---

# Index

## A

- abends, responding to 6
- about this guide ix
- access plan, viewing with Control Center 12
- adapter number (NetBIOS) 40
- adm subdirectory, tools 13
- administration of databases 11
- administrative tools (DB2)
  - description 12
  - logging 92
  - tracing 103
- administrative tools (Windows NT) 130
- administrator authority 18
- agent 77
- agent pool 77
- AIX commands 114
  - See also* commands, UNIX-based
- alert
  - overview 94
  - SNA 107
  - SNA, routing (OS/2) 125
- alias 29
- APPC protocol 43
- application requesters (AR) 45
- application servers (AS) 45
- applications
  - CLI and ODBC 56
    - determining status 6
  - for Extended Enterprise Edition 67
  - Java 53
  - slow or hanging because of locking 22
  - suspended (Extended Enterprise Edition) 66
  - suspended (TCP/IP) 36
  - suspended or looping (applications) 55
  - troubleshooting 53
- architectural integrity, checking 13
- AS/400, connections 45
- authentication
  - DB2 for Windows NT 23
  - host connections 50
- AutoLoader (Extended Enterprise Edition) 63

## B

- backup
  - plan 11
  - troubleshooting 19
  - using Control Center 12
- bin subdirectory, tools 13
- binary data, loading (Extended Enterprise Edition) 64
- boldface type xi

## C

- call stack 91
- CLI trace 105
- client
  - APPC connections 43
  - host 51
  - host connections 45
  - IPX/SPX connections 37
  - named pipe connections 42
  - NetBIOS connections 40
  - TCP/IP connections 35
  - troubleshooting 4, 29
- Client Configuration Assistant 30
- Client Setup tool (Windows 3.x) 30
- code page conversions 54
- collecting data for DB2 customer service 9
- command line processor, unable to use 20
- commands, DB2
  - db2\_all (Extended Enterprise Edition) 116
  - db2\_call\_stack 91
  - db2autold (Extended Enterprise Edition) 63
  - db2bfd 13
  - db2cat 13
  - db2ckmig 16
  - db2dart 13
  - db2drdat 106
  - db2evmon 15
  - db2expln 27
  - db2fmt 27
  - db2look 13
  - db2recr 14
  - db2resdb 14
  - db2set 30
  - db2snmpd 95
  - db2start 61

- commands, DB2 (*continued*)
  - db2tbst 14
  - db2trc 97
  - db2untag 14
  - ddcstrc (DB2 Connect) 106
  - dynexpln 28
  - list applications 6
  - load 20
  - monitoring 14
  - netstat (Extended Enterprise Edition) 117
  - rah (Extended Enterprise Edition) 116
  - register 38
  - reorg 27
  - reorgchk 27
  - runstats 27
  - troubleshooting 13
- commands, OS/2
  - cmtrace (SNA trace) 107
  - epw 122
  - epwinst 121
  - epwrout 125
  - pstat 126
  - syslog 120
- commands, UNIX-based
  - cmtrace (SNA trace) 107
  - correct syntax 21
  - dbx 112
  - df 115
  - errpt (AIX) 114
  - iostat 7, 116
  - ipcrm 7
  - ipcs 7
  - kill 7
  - lsattr (AIX) 114
  - lsps (AIX) 114
  - netstat 116
  - ps 113
  - pstack (Solaris) 115
  - spmon (Extended Enterprise Edition) 116
  - truss (SVR4) 115
  - vmstat 115
  - xmperf 115
- communicating with IBM 9
- compiling 54
- configuration parameters
  - DB2\_RR\_TO\_RS 22
  - diaglevel 81
  - diagpath 81
  - dlchktime 22
  - locklist 22

- configuration parameters (*continued*)
  - locktimeout 22
  - logfilsiz 17
  - logprimary 17
  - logsecond 17
  - maxagents 77
  - maxlocks 22
  - nname (NetBIOS) 40
  - num\_poolagents 77
  - sheapthres 28
  - sortheap 28
- configuring clients 30
- connection
  - analyzing events 12
  - changing number for DRDA 49
  - testing on the server 31
  - troubleshooting 29
  - unsuccessful 33
- Control Center 12
- conventions x
- coordinator agent 77
- core files (DB2) 91
- core files (UNIX) 112
- corruption, checking for 13
- customer service, contacting 9

## D

- data
  - architectural integrity, checking 13
  - backups 19
  - binary, loading (Extended Enterprise Edition) 64
  - format (Extended Enterprise Edition) 64
  - importing 20
  - indexing (Extended Enterprise Edition) 66
  - loading 20
  - restoring 19
  - splitting and loading (Extended Enterprise Edition) 63
- database
  - administration troubleshooting 11
  - alias 29
  - consistency 16
  - creating (Extended Enterprise Edition) 62
  - design 11
  - objects, displaying with Control Center 12
  - production, mimicking 13
  - recovery (Extended Enterprise Edition) 66
  - remote 29
  - system catalog statistics, mimicking 13

- database directory 29
- database manager
  - monitoring 14
  - monitoring with Control Center 12
- database partition 59
- DB2
  - administrative tools 12
    - See also* administrative tools (DB2)
  - Extended Enterprise Edition 59
  - host connections 45
  - information, finding 8
  - maintaining 9
- DB2 Connect
  - tracing 106
  - troubleshooting 45
- DB2 customer service, contacting 9
- DB2COMM registry value 32
- db2dart tool 13
- db2diag.log file
  - additional information in syslog (UNIX) 109
  - diagnosing if listeners are up 77
  - dump files 90
  - example with DB2 trace 101
  - examples 86
  - header 83
  - inconsistent data 17
  - interpreting 83
  - NetBIOS entries 41
  - pruning 83
  - relationship to Windows NT event viewer 131
  - SQLCA structures 85
  - tracing if insufficient 97
  - trap files 91
    - troubleshooting client connections 32
- DB2MNBCKCTLR profile variable (Windows NT) 24
- deadlocks 21
- Desktop Management Interface (DMI) service layer 94
- diaglevel, setting 81
- diagnostic tools
  - Macintosh 133
  - OS/2 119
  - UNIX 109
  - Windows NT 129
  - Windows Version 3.1 and Windows 95 129
- diagpath, setting 81
- directories
  - changing with Control Center 12
- Distributed Relational Database Architecture (DRDA)
  - tracing 106
  - troubleshooting 45

- domain controller (Windows NT) 24
- domain security (Windows NT) 25
- Dr. Watson (Windows) 129
- dump files 90
  - core file (UNIX) 112
  - OS/2 alerts 126
  - traces 98
- dump utility, standalone (OS/2) 119

## E

- environment variables 30
- error code 85
- error log (OS/2) 120
- error record, FFST for OS/2 error log 125
- Event Analyzer 12
- event snapshots 15
- event viewer (Windows NT) 130
- exceptions
  - responding to 6
  - trap file 91
- explain facility 27
- export message files 93
- Extended Enterprise Edition
  - commands 116
  - determining failing nodes 61
  - developing applications 67
  - monitoring applications 56
  - processes 78
  - severe error 60
  - starting 61
  - troubleshooting 59

## F

- FFDC 81
  - See also* first failure data capture (FFDC)
- FFST/2 implementation (OS/2) 121
- files
  - autoexec.bat (OS/2) 39
  - config.sys 38
  - core (UNIX) 112
  - db2alert.log 94
  - db2cc.log 92
  - db2cli.ini 30, 57
  - db2diag.log 83
    - See also* db2diag.log file
  - dbttrc.trc 104
  - dump 90
  - net.cfg (OS/2) 38, 39

- files (*continued*)
  - nwcalls.dll (Win-OS/2) 39
  - odbc.ini 30
  - services (TCP/IP) 36
  - size limit 20
  - syslog (UNIX) 109
  - system (Solaris) 116
  - tli\_spx.dll (Win-OS/2) 39
  - TLI\*.DLL (Novell IPX/SPX) 38
  - trap 91
- firewall 75
- first failure data capture (FFDC)
  - alerts 94
  - collecting on OS/2 137
  - configuration parameters 81
  - db2diag.log file 83
  - dump files 90
  - first failure data capture 81
  - introduction 9
  - overview 81
  - SQLCA structures 85
  - stack traceback 91
  - tracing if insufficient 97
  - trap files 91
- fix packs, applying 9
- format of data (Extended Enterprise Edition) 64
- forums 8
- free space, identifying (UNIX) 115
- ftp sites 9

**G**

- general protection faults (GPF), responding to 6
- graphical user interface (GUI)
  - administrative tools 12
  - tracing 104
- group names (Windows NT) 23
- groups (Windows NT) 25

**H**

- hangs, responding to 6
- hardware errors
  - identifying (AIX) 114
  - identifying (HP-UX) 115
- hexadecimal values in db2diag.log file 85
- history of problems, maintaining 3
- host
  - troubleshooting connections to 45

- HPS switch, checking (Extended Enterprise Edition) 116
- hung applications 55

**I**

- I/O activities, monitoring (UNIX) 116
- IBM, contacting 9
- ICLUI trace 104
- idle agent 77
- import message files 93
- importing data 20
- index
  - architectural integrity, checking 13
  - failing in Extended Enterprise Edition 66
  - inconsistent, re-creating 14
- indoubt transactions (Extended Enterprise Edition) 67
- Information Center 12
- information on DB2, finding 8
- Internet Engineering Task Force (IETF) 94
- IPX/SPX protocol 37
- isolation levels (CS, RS) 22
- italic type xi

**J**

- JDBC trace 105
- job scheduling, using Control Center 12

**K**

- keepalive settings (TCP/IP) 36
- kernel debugging (OS/2) 119
- kernel, updating (UNIX) 18

**L**

- LAN, determining availability 34
- Lana number (Windows 95) 40
- link time 55
- listeners
  - definition 77
  - testing 31
- load message files 93
- loading data, troubleshooting tips 20
- location of problem, determining 4
- locking 21
- log space 17
- logging, all platforms 81

- logs
  - increasing space for importing data 20
  - system log (OS/2) 120
- loops
  - in applications 55
  - responding to 6
- Lotus Approach 56
  - See also ODBC applications

## M

- maintaining DB2 9
- Management Information Base (MIB) 94
- media management with Control Center 12
- memory, identifying (AIX) 114
- message files, import, export and load 93
- messages, responding to 5
- Microsoft Access 56
  - See also ODBC applications
- Microsoft Query 56
  - See also ODBC applications
- migration
  - troubleshooting 16
  - Version 1 of DB2 for OS/2 14
- misc subdirectory, tools 13
- monitor switches 14
- monitoring database manager with Control Center 12
- monospace type xi
- MVS, connections 45

## N

- named pipes 42
- NetBIOS protocol 40
- network traffic, identifying (UNIX) 116
- newsgroups 8
- next key locking 22
- node directory 29
- nodes (Extended Enterprise Edition)
  - definition 59
  - failing 61
  - slow 62
  - won't start 62

## O

- ODBC applications
  - 16-bit 56
  - configuring driver 57
  - tracing 105

- ODBC applications (*continued*)
  - troubleshooting tips 56
- online help 8
- optimizer, viewing access plans 12
- origins of problems, determining 4
- OS/2 commands 119
  - See also commands, OS/2
- OS/390, connections 45

## P

- packed descriptors for tables, viewing 13
- page rate, monitoring (UNIX) 115
- paging space, identifying (AIX) 114
- passwords 18
- patches, applying 9
- performance
  - data, displaying (UNIX) 115
  - improving 27
  - monitoring with Control Center 12
  - using the database monitor 14
- performance monitor (Windows NT) 130
- point-in-time monitoring 14
- problem description, determining 4
- problem determination 3
  - See also troubleshooting
- Problem Determination Tools folder 137
- process model
  - differences between Intel and UNIX 78
  - displaying threads (OS/2) 126
  - overview 75
  - ps command (UNIX) 113
  - viewing 78
- process status command 113
- PTFs, applying 9

## R

- rebinding 20
- recovery
  - DB2 server 19
  - Extended Enterprise Edition 66
  - operating system 6
- redbooks 8
- registry values 30
- remote debugging (OS/2) 119
- resources, increasing (IPX/SPX) 38
- responding to DB2 users 4
- restoring data 19

- restoring Version 1 database image (DB2 for OS/2) 14
- return codes
  - byte-reversal 85
  - example 88
  - list 147
- roadmaps
  - logging, tracing, and alerts 71
  - Part 1 1
  - Part 2 69
  - troubleshooting tasks ix

## S

- SAM tool (HP-UX) 115
- schema names, considerations during migration 16
- screen captures (Windows) 131
- security service (Windows NT) 23
- segmentation violation
  - logs 91
  - responding to 6
- server
  - administrative tools 12
    - See also* administrative tools (DB2)
  - client connections 32
  - collecting data for DB2 Customer Service 28
  - failure 16
  - timing out (NetBIOS) 41
  - troubleshooting 4, 11
- service tools 137
- services file (TCP/IP) 36
- Setup Communications tool 30
- signal 91
- Simple Network Management Protocol (SNMP) 94
- SmartGuides 12
- Snapshot Monitor 12
- SNMP subagent, starting 95
- software fixes, applying 9
- splitting data (Extended Enterprise Edition) 63
- SQL messages
  - definition 5
  - getting help for 5
  - responding to 5
  - SQL0294N 14
  - SQL0805N 20
  - SQL0818N 20
  - SQL0902C 19
  - SQL0911N 23
  - SQL0984C 17
  - SQL1003N 56
  - SQL1015N 60

- SQL messages (*continued*)
  - SQL1016N 18
  - SQL1018N 18
  - SQL1034C 60
  - SQL1042C 18
  - SQL1061W 66
  - SQL1109N 39
  - SQL1224N 60
  - SQL1402N 50
  - SQL1403N 34, 50
  - SQL30073N 51
  - SQL30081N (IPX/SPX) 37
  - SQL30081N (NetBIOS) 40
  - SQL30081N (TCP/IP) 35
  - SQL30082N 50
  - SQL30082N (NetBIOS) 42
  - SQL3805N 20
  - SQL5043N 32
- SQL state 5
- SQL statement
  - syntax conventions for 173
- SQLCA (SQL communication area) 143
  - Extended Enterprise Edition nodes 61
  - interpreting 85
- stack traceback
  - displaying (Solaris) 115
  - for suspended system 8
  - gathering 91
- starting a trace 99
- stored procedure
  - developing 53
  - fenced 77
- subagent 77
- suspended applications 55
- suspensions, responding to 6
- symptom record, FFST for OS/2 125
- syntax diagrams, reading 173
- syslog (UNIX) 109
- system calls, tracing (UNIX) 115
- system catalog statistics, mimicking 13
- system configuration, using Control Center 12
- system file (Solaris) 116
- system log (SYSLOG)
  - OS/2 120
  - UNIX 109
- System Performance Monitor for OS/2 (SPM/2) 119
- system resources, increasing (IPX/SPX) 38
- Systems Network Architecture (SNA)
  - overview 45
  - routing alerts (OS/2) 125

Systems Network Architecture (SNA) (*continued*)  
tracing 107  
troubleshooting 49  
SystemView agent 94

## T

table  
system catalog tables, mimicking 13  
viewing packed descriptors 13  
table space  
container, using again with db2untag 14  
state, getting description of with db2tbst 14  
TCP/IP protocol 35  
test tool to mimic production databases 13  
threads 75  
tools  
client connectivity 30  
database administration 12  
*See also* administrative tools (DB2)  
trace facility, DB2 97  
traces  
administration tools (DB2) 103  
CLI/JDBC/ODBC 105  
DB2 97  
DB2 Connect 106  
DRDA 106  
dumping (DB2) 98  
examples (DB2) 100  
graphical interface (ICLUI) 104  
SNA 107  
turning off (DB2) 98  
verifying (DB2) 100  
trap 91  
troubleshooting  
abends, responding to 6  
APPC protocol 43  
applications 53  
clients 29  
contacting IBM 9  
database administration 11  
db2diag.log 81  
Extended Enterprise Edition 59  
getting started 3  
history of problems, maintaining 3  
host connections 45  
information, finding 8  
IPX/SPX protocol 37  
loops, responding to 6  
maintaining products 9

troubleshooting (*continued*)  
messages, responding to 5  
named pipes 42  
NetBIOS protocol 40  
origins and symptoms of problems, determining 4  
problem description, determining 4  
return codes 147  
servers 11  
suspensions, responding to 6  
TCP/IP protocol 35  
where to find a problem 4

## U

UNIX-based commands 114  
*See also* commands, UNIX-based  
updates to DB2, applying 9  
user forums 8  
user name  
authentication on host 50  
case sensitivity 50  
not valid on Windows 95 34  
Windows NT 23  
user names 18  
user-defined functions (UDFs)  
developing 53  
fenced 77  
utilities, unable to use 20

## V

Visual Explain 12  
VM, connections 45  
VSE, connections 45

## W

WarpCenter (OS/2 Warp Version 4) 119  
watchdog process 78  
World Wide Web 8





---

## Contacting IBM

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

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

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

### Telephone

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

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

For information on how to contact IBM outside of the United States, see Appendix A of the IBM Software Support Handbook. You can access this document by selecting the "Roadmap to IBM Support" item at: <http://www.ibm.com/support/>.

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

### World Wide Web

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

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

### Anonymous FTP Sites

<ftp.software.ibm.com>

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

### Internet Newsgroups

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

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

### CompuServe

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

All DB2 products are supported through these forums.

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



Part Number: 10J8169



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

S10J-8169-00



10J8169

