IBM DB2 Universal Database Express Edition

Release Notes for DB2 UDB Express Edition

Version 8.2 FixPak 1 (equivalent to Version 8.1 FixPak 8)

IBM DB2 Universal Database Express Edition

Release Notes for DB2 UDB Express Edition

Version 8.2 FixPak 1 (equivalent to Version 8.1 FixPak 8)

Before using this information and the product it supports, be sure to read the general information under *Notices*.

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.

You can order IBM publications online or through your local IBM representative.

- To order publications online, go to the IBM Publications Center at www.ibm.com/shop/publications/order
- To find your local IBM representative, go to the IBM Directory of Worldwide Contacts at www.ibm.com/planetwide

To order DB2 publications from DB2 Marketing and Sales in the United States or Canada, call 1-800-IBM-4YOU (426-4968).

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 2002 - 2004. All rights reserved.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

About the Release Notes vii

About this release 1

New in this release 1
- DB2 UDB Express Edition enhancements 1
- Creation of 32–bit and 64–bit instances 1
- New registry variable: DB2_ASYNC_IO_MAXFILOP 1
- New SMS user table space for automatic maintenance and maintenance health indicators . 2

Product fix history 2

Backward compatibility 2
- Backward compatibility of DB2 UDB Version 8.2 databases 2
- Clarification of DB2 UDB client support 3
- Health registry changes when migrating from DB2 UDB Version 8.2 back to DB2 UDB Version 8.1 . . 3
- Two versions of SQL Assist are launched from DB2 UDB 3
- Change in Unicode server behavior 3
- Database configuration parameter changes during migration 3
- db2diag.log format message enhancements . . . 4
- db2set profile registry variables and DB or DBM configuration parameters are now logged 4

Product compatibility 5
- Microsoft XP fix is needed on 64-bit operating systems 5
- Windows XP operating systems 5
- DB2 UDB HADR separately priced option available 5
- DB2 Warehouse Manager (Version 8.2) and IBM DB2 OLAP Server FP3 and later 6
- Red Hat Linux support with the Data Warehouse Center 6
- Alternative Unicode conversion tables for the coded character set identifier (CCSID) 5039 . . . 6
- Replacing the Unicode conversion tables for coded character set (CCSID) 5039 with the Microsoft conversion tables 7
- Alternative Unicode conversion tables for the coded character set identifier (CCSID) 954 . . . 7
- Replacing the Unicode conversion tables for coded character set (CCSID) 954 with the Microsoft conversion tables 8
- Alternative Unicode conversion tables for the coded character set identifier (CCSID) 943 . . . 8
- Replacing the Unicode conversion tables for coded character set (CCSID) 943 with the Microsoft conversion tables 10

Installation, migration, upgrade, and configuration information. 13

Hardware and software requirements. 13

Installation notes 13
- Directory path cannot contain blanks (Linux) . . 13
- JDK levels for DB2 UDB (Linux on IA64 and Linux on PowerPC). 13
- Uncompressing installation images (Linux) . . . 14
- DB2 UDB does not allow non-English installation path (Windows) 14
- Downloading and uncompressing FixPaks for multiple products (Windows) 15
- Installing DB2 UDB from installation images (Linux) 15
- Installing DB2 UDB from installation images (Windows). 15
- Disk space required for a DB2 UDB response file installation 16
- Restrictions to adding products using the db2setup command (Linux) 16
- DB2 Web Tools 17
- Recreating beta versions of databases. 17
- Installing MDAC files for national language versions of DB2 UDB 18
- Installing additional Asian fonts (Linux) 18
- Configuring the Development Center to use Java Development Kits (Linux) 19
- Creating group and user IDs on United Linux 1.0 and SuSE Linux distributions 20
- Help system daemon does not start after installing with the db2_install command (Linux) . 20
- Installation image reduction tool (Windows) . . 20
- DB2 Universal Database, Version 8 HTML documentation installation restriction (Windows) . 20
- Previous installations updated to the latest level (Windows). 21
- System requirements for the DB2 .NET Data Provider (Windows) 21
- Installing DB2 Version 8 Clients and DB2 Connect PE as a non-Administrator (Windows) . 21

Migration notes 21
- Migrating DB2 Universal Database (Windows). . 21
- Migrating DB2 Universal Database when using DataJoiner or replication 22
- Migrating a DB2 Version 8 Windows 32–bit database to Windows 64-bit 22
- Migrating XML Extender from previous versions 23

Uninstallation information 25

Uninstalling DB2 UDB silently (Windows) 25

Product code for DB2 Information Center removal using a silent uninstall (Windows). 26

Limitations 27

IMPORT REPLACE does not honor the Not Logged Initially clause 27

Data export with ODBC to file warehouse program 27

Data types that are not supported in the Development Center's integrated SQL debugger . . 27

© Copyright IBM Corp. 2002 - 2004

Structured types in the Development Center . . . 27
Development Center limitations for 64–bit operating systems. 27
Development Center (Linux). 28
Debugging stored procedures with double quotation marks 28
Path settings needed to enable Java routines to compile in the Development Center 28
Development Center limitations to run and debug Java stored procedures simultaneously 28
Cursors in PHP applications. 29
Four unsupported SQL administrative routines . . 29
Bind option limitations for CLI packages 29
CLI LOAD restriction for specifying column names 30
Tools catalog database creation not supported (Linux for AMD64) 30
Memory limitations for DB2 UDB Express and DB2 Workgroup Server Edition V8.2. 30
Scheduling a warehouse process to run at intervals 30
Load and Import Columns page does not support DBCS characters in IXF files 30
Security plug-in problem and restriction for DB2 UDB clients (Windows) 31
Two-part user ID not supported (Windows ME) . . 31
Minimum display settings for GUI tools. 31
Improper display of GB18030 characters in the title bar of a window. 31
Health indicator restrictions 31

Known problems and workarounds . . 33

The db2nkill tool is not shipped in DB2 UDB Express Edition 33
The "db2gcf –k" command fails on DB2 UDB Express Edition 33
Hotkeys not working in Microsoft Visual Studio .NET Framework 1.1 33
The simplified Chinese locale (Red Hat Linux) . . 34
SQLFLAG (STD) precompiler option error 34
DB2 Connect Custom Advisor 34
Displaying Indic characters in the DB2 GUI tools. . 34
Enclose DB2 Information Center search terms within quotation marks if they contain numbers 35
Secure environments (Windows) 35
XML Extender sample programs renamed 36
Using the new sample programs with the sample sqx files 37
Decomposing documents in XML Extender that contain non unique attribute and element names . . 37

Documentation updates 41

Administration: Implementation 41
Authentication considerations for remote clients 41
Direct I/O (DIO) and concurrent I/O (CIO) support. 41
Distributor technology and automatic client rerouting 41
Local system account support (Windows) . . . 42
Two-part user ID support. 43
Additional information for Kerberos support . . 43
Administration: Performance 44
New system environment variables (Linux). . . . 44
New communication registry variable 45
SQL compiler variables 46
Configuration parameter updates 46
SQLDBCONF database configuration parameter file 47
Change to the DB2_HASH_JOIN default value 47
DB2NTNOCACHE registry variable is deprecated. 48
Explain tables and organization of explain information 48
Guidelines for capturing explain information . . 48
Additional return codes from db2CfgGet API, collate_info parameter 49
Automatic setting of default prefetch size and update defaults 49
Administration: Planning 50
Supported territory codes and code pages . . . 50
XA function supported by DB2 Universal Database 51
Updated TPM and tp_mon_name values for xa_open string formats 53
Activating conversion tables for code pages 923 and 924. 54
Conversion table files for euro-enabled code pages 55
API Reference 60
SYNCPOINT option support 60
New field for SQLEDBDESC structure 60
Correction to new field in the SQLB-TBSPQRY-DATA structure 60
Application development: Building and Running Applications 60
Customizing precompile and bind options for SQL procedures 60
C/C++ compile option required (Linux on PowerPC 64-bit) 61
Setting environment variables for Micro Focus COBOL stored procedures (Windows) 61
Application development: Call Level Interface (CLI) 61
MapBigintCDefault CLI/ODBC configuration keyword 61
DescribeOutputLevel CLI/ODBC configuration keyword 62
Application development: Programming Client Applications 63
OleDbReportIsLongForLongTypes CLI/ODBC configuration keyword 63
OleDbSQLColumnsSortByOrdinal CLI/ODBC configuration keyword 64
DB2 Data Source property group for the IBM DB2 OLE DB Provider. 64
Incorrect URL syntax in the DB2Binder syntax diagram 66
Rerouting DB2 Universal JDBC driver clients . . 66
Customizing the DB2 Universal JDBC driver configuration properties 68
db2secFreeToken function removed 69
Deploy custom security plug-ins carefully . . . 69
Security plug–ins 70
Security plug–in APIs 70

Security plug–in naming conventions (Linux) . . 70
GSS-API security plug-ins do not support multiple-flow authentication. 70
GSS-API security plug-ins do not support message encryption and signing 70
Implicit ending of transactions in stand-alone applications 70
Distributed transaction support. 70
Application development: Programming Server Applications 71
Common language run time (CLR) routine execution control modes (EXECUTION CONTROL clause) 71
Maximum decimal precision and scale in common language run time (CLR) routines. . . 72
Command Reference 72
db2inidb - Initialize a mirrored database command 72
Usage note for the db2iupdt command 72
db2pd - Monitor and troubleshoot DB2 command 72
New parameter for the db2sqljcustomize command 73
New parameter for the sqlj command 73
Update to the SET CLIENT command 73
Update to the PRECOMPILE command 73
Update to the UPDATE HISTORY FILE command 73
Updates to the EXPORT and IMPORT commands 74
Update to the LOAD command 74
File type modifiers for the load utility 74
File type modifiers for the import utility. . . . 75
ATTACH command 76
RECOVER DATABASE command 76
UPDATE HISTORY FILE command 76
db2updv8 - Update database to version 8 current level command 77
Formatting trap files (Windows) 79
Data Movement 79
Index information lost when the import utility is used. 79
Data Recovery and High Availability 80
High availability disaster recovery overview . . 80
Cross-platform backup and restore support. . . 80
Backing up to tape (Linux) 80
Tivoli Storage Manager 80
Value restrictions for the HADR local host and local service parameters 81
Additional system requirements for high availability disaster recovery. 81
Non-replicated operations for high availability disaster recovery 81
HADR does not support raw logs 82
Data Warehouse Center 82
Defining DB2 warehouse sources and targets . . 82
Updates to the Business Intelligence tutorial . . 82
Setting the purge limit for warehouse log files. . 83
Data Warehouse Center support for CURSOR load 83
Unicode warehouse control database migration and limitations 83
Change in date format for Modified column . . 83
Defining statistical transformers in the Data Warehouse Center 84
Prerequisite for the iSeries warehouse agent . . 84
DB2 .NET Data Provider 84
DB2Connection.ConnectionString property . . . 84
Development Center 84
Restrictions to DB2 Universal type 2 and type 4 drivers 84
GUI Tools 85
Modifying the statement termination character 85
Database unavailable status in the database details pane of the Control Center 86
Default write to table output generation (Create Event Monitor) 87
Installation and Configuration Supplement 87
Application server for DB2 UDB 87
DB2 Embedded Application Server enablement 89
Deploying DB2 Web Tools 89
Direct I/O on block devices (Linux) 94
DB2 Information Center daemon (Linux) . . . 94
Response file keywords and sample files . . . 95
Response file installation error codes 95
Required user accounts for installation of DB2 servers (Windows) 96
Asynchronous I/O support (Linux) 96
db2ln command modified to create DB2 UDB 64-bit library links (Linux) 96
Message Reference 97
ADM message updates 97
CLP message updates 97
DBI message updates 97
SQL message additions and updates 98
Quick Beginnings 102
Verifying that your databases are ready for migration. 102
Common Criteria certification for DB2 UDB . . 102
Spatial Extender 102
Verifying the Spatial Extender installation . . . 102
SQL Administrative Routines 102
EXEC_DB2_SCRIPT procedure unavailable . . 102
GET_DB_CONFIG is a stored procedure . . . 103
Column expansion for snapshot UDFs 103
SQL Reference 105
CONNECT statement. 105
Schema used by the Explain facility 105
String representations of datetime values . . . 105
System monitor 106
Decoupled applications list without connection concentrator enabled 106
Progress monitoring of the run time rollback process 106
XML Extender 107
Decomposing documents larger than 1 MB . . 107
Configuring MQ XML UDFs with XML Extender 107
XML Extender environment variable DB2DXX_MIN_TMPFILE_SIZE 108
DB2XML.XMLVarchar UDT redefinition . . . 108

Appendix A. DB2 UDB FixPak CD directory structure 109

Appendix B. Contacting IBM 111
Product information 111

Appendix C. Notices 113
Trademarks 115

About the Release Notes

Content:

The release notes contain the latest information for the following DB2®, Version 8 products:

```
DB2 Universal Database Personal Edition
DB2 Universal Database Workgroup Server Edition
DB2 Universal Database Workgroup Server Unlimited Edition
DB2 Universal Database Enterprise Server Edition
DB2 Personal Developer's Edition
DB2 Universal Developer's Edition
DB2 Warehouse Manager
DB2 Warehouse Manager Sourcing Agent for z/OS
DB2 Data Links Manager
DB2 Net Search Extender
DB2 Spatial Extender
DB2 Intelligent Miner Scoring
DB2 Intelligent Miner Modeling
DB2 Intelligent Miner Visualization
DB2 Connect Application Server Edition
DB2 Connect Enterprise Edition
DB2 Connect Personal Edition
DB2 Connect Unlimited Edition
DB2 Query Patroller
```

Version information:

The most up-to-date documentation is available in the latest version of the DB2 Information Center, which is accessed through a browser. The URL for downloading the latest documentation is provided in the Additional resources section that follows this section.

Directory paths:

Windows® systems use backslashes (\) to delimit directories in a directory path. Linux™ systems use forward slashes (/) as delimiters. The Release Notes follow this convention when the information is platform specific. However, when the information is platform independent, you might need to enter the directory path differently than shown. For example, if you have a Windows system, you must enter the directory path using backslashes (\) if the Release Notes show forward slashes (/). Conversely, if you have a Linux system, you must enter the directory path using forward slashes (/) if the Release Notes show backslashes (\).

DB2 Information Integrator:

For information about currently known outstanding issues with DB2 Information Integrator and its associated technologies, including federated systems, SQL replication, Q replication, event publishing, Web services, and metadata management, see the DB2 Information Integrator Release Notes. The DB2 Information Integrator Release Notes are available on the DB2 Information Integrator support Web page at:
http://www.ibm.com/software/data/integration/db2ii/support.html

Additional resources:

© Copyright IBM Corp. 2002 - 2004

The *Data Links Manager Administration Guide and Reference* was updated in PDF form (book number SC27-1221-01) at the time of FixPak 1 and is available for download at the DB2 support site:
http://www.ibm.com/software/data/db2/udb/support.html.

Documentation for the DB2 Life Sciences Data Connect product is available for download from the IBM® software site:
http://www.ibm.com/software/data/db2/lifesciencesdataconnect/

If you want to view the DB2 documentation in HTML format, you can access the DB2 HTML Information Center online from
http://publib.boulder.ibm.com/infocenter/db2help/.

Alternatively, you can install the DB2 HTML Information Center on your system; a *DB2 HTML Documentation* CD image is available for download from the same Web site. Updates are made to the DB2 HTML documentation with every release. For the latest documentation, access the DB2 HTML Information Center online or download the *DB2 HTML Documentation* CD image for installation on your system. PDF documentation is updated less frequently than the HTML Information Center.

More information on the DB2 Development Center and DB2 for z/OS® is available at http://www.ibm.com/software/data/db2/os390/spb/.

For the latest information about the DB2 family of products, obtain a free subscription to *DB2 Magazine*. The online edition of the magazine is available at http://www.db2mag.com; instructions for requesting a subscription are also posted on this site.

About this release

New in this release

This section provides a brief summary of the enhancements available with this release of DB2 Universal Database.

DB2 UDB Express Edition enhancements

DB2 Universal Database (UDB) Express Edition Version 8.2 FixPak 1 (equivalent to Version 8.1 FixPak 8) introduces 64–bit support on the following platforms:

- Linux for AMD64
- Linux for IA64
- Linux for PowerPC 64
- Windows Server 2003 64–bit Itanium based systems
- Windows XP 64–bit Edition

The benefits of 64–bit coverage include more address space for code, variables, shared memory, and buffer pools.

Creation of 32–bit and 64–bit instances

You can create 32-bit instances on any DB2 UDB Version 8.2 product on a 64-bit platform. You can create 64-bit instances on all DB2 UDB Version 8.2 products except DB2 UDB Personal Edition and DB2 Connect Personal Edition.

New registry variable: DB2_ASYNC_IO_MAXFILOP

A new general registry variable has been added at Version 8.2 FixPak 1 (equivalent to Version 8.1 FixPak 8).

Variable Name
DB2_ASYNC_IO_MAXFILOP

Operating System
All

Values
Default: the value of MAXFILOP (the configuration parameter).

Values: From the value of MAXFILOP to the value of MAX_INT.

Description
DB2 engine dispatchable units (EDUs), such as agents, prefetchers, page cleaners, that read and write data from, and to, table spaces, use the MAXFILOP configuration parameter to limit the number of file handles each process or thread has open at any one time. Since most I/O on systems is done by EDUs that perform I/O asynchronous to agents (page cleaners and prefetchers), those EDUs can benefit from a larger value for the maximum number of file handles per EDU than MAXFILOP. This is especially the case on systems where there are a lot of files. This registry variable is used to allow the upper limit of file handles to increase beyond the value of MAXFILOP for page cleaners and prefetchers.

© Copyright IBM Corp. 2002 - 2004

New SMS user table space for automatic maintenance and maintenance health indicators

A new SMS user table space, SYSTOOLSPACE, is created by default by the `db2hmon` process to store tables used by automatic maintenance and the maintenance health indicators. The table space is created in the default database directory.

If you do not want the tables used by the automatic maintenance and the maintenance health indicators to be created within the SYSTOOLSPACE table space, you can create the automatic maintenance tables in a table space of your choice. However, you must drop the tables in SYSTOOLSPACE and drop SYSTOOLSPACE before using stored procedure calls to create the tables.

With valid authorization, use the SQL DROP statement to drop SYSTOOLSPACE if you do not need it. Ensure that there are no tables within the table space before you drop the table space. If there are tables within the tables space and it is dropped, the tables and table space are automatically recreated the next time the database is started.

To create the needed tables in a table space of your choice, issue the following stored procedure calls:

```
CALL SYSINSTALLOBJECTS('DB2AC','C','<tablespace_name>', NULL)
CALL SYSINSTALLOBJECTS('POLICY','C','<tablespace_name>', NULL)
```

where the `tablespace_name` is a table space of your choice that has 4 KB pages.

Automatic reorganization and the `Reorganization required` health indicator will automatically create a user temporary table space, SYSTOOLSTMPSPACE, if a user temporary table space does not exist.

Product fix history

For a list of Authorized Program Analysis Reports (APARs) addressed by this fix pack, visit the following Web page:

http://www.ibm.com/software/data/db2/udb/support.html

Backward compatibility

Backward compatibility of DB2 UDB Version 8.2 databases

If you create a database with DB2 Universal Database™ Version 8.2, you cannot use that database at a Version 8.1 level. That database can only be used at a Version 8.2 or later level.

Databases created at the DB2 UDB Version 8.2 level may have additional functionality that was not available on earlier versions. This difference may result in unexpected and undesirable behavior if you attempt to move your new database to a previous release of DB2 UDB.

Note: The only way to move a database from Version 8.2 back to Version 8.1 is if the database was originally created under Version 8.1. Even then, backward migration is possible only after running the **db2demigdb** tool. However, you might encounter problems if you used built-in functions that have changed in Version 8.2.

Clarification of DB2 UDB client support

The "DB2 client overview" section of the *DB2 Quick Beginnings for Clients* book states the following:

> DB2 clients can connect to DB2 servers two releases later or one release earlier than the client's release level, as well as to servers at the same release level.

The amendment to that statement is as follows:

> While connections from Version N clients to Version N + 2 servers are possible in some environments, this connection is a supported configuration only as long as Version N is in service. Once Version N is withdrawn from service, this configuration is no longer supported.
>
> DB2 Version 6 clients connecting to a DB2 Version 8 server is no longer supported because Version 6 has been withdrawn from service.
>
> Similarly for DB2 UDB server support, a Version N client can connect to a Version N - 1 server, unless the Version N - 1 server is out of service.

Health registry changes when migrating from DB2 UDB Version 8.2 back to DB2 UDB Version 8.1

Any registry changes made at the DB2 UDB Version 8.2 level are lost when you migrate back to DB2 UDB Version 8.1. The registry reverts to the version 8.1 HealthRules.reg file that contains the settings that existed before you upgraded to DB2 UDB Version 8.2 and started using the settings in the HealthRules2.reg file.

Two versions of SQL Assist are launched from DB2 UDB

You can invoke both Version 7 and Version 8 of SQL Assist from within DB2 Universal Database, Version 8. You can start Version 7 from the DB2 Data Warehouse Center. All other centers start the latest Version 8. The product online help has additional information for SQL Assist, Version 7.

Change in Unicode server behavior

In Version 7, Unicode servers ignored any graphic code pages sent by applications at connect time and assumed that UCS2 Unicode (code page 1200) was being used. Version 8 Unicode servers now respect the code page sent by the client.

Database configuration parameter changes during migration

DB2 UDB Version 8.2 uses a new 16K database configuration parameter file named SQLDBCONF. This is a separate file from the DB2 UDB Version 8.1 4K database configuration parameter file named SQLDBCON.

After migrating to DB2 UDB Version 8.2, the product migrates the contents of the Version 8.1 4K file and uses the 16K file for logging database configuration parameter changes. The Version 8.1 4K file is retained, but it is not used.

If you migrate back to DB2 UDB Version 8.1, the DB2 UDB Version 8.1 product reverts to using the original Version 8.1 4K file for logging database configuration parameter changes. The Version 8.2 16K file is retained, but it is not recognized by the DB2 UDB Version 8.1 product. Changes made to the 16K database configuration parameter file between migrating to Version 8.2 and migrating back

to Version 8.1 are, in effect, concealed from the earlier DB2 UDB level because the changes are not migrated to the original 4K file.

In addition, if you migrate to DB2 UDB Version 8.2 again, the DB2 UDB Version 8.2 product recognizes that the 16K database configuration file already exists and reverts to using the Version 8.2 16K file for logging database configuration parameter changes. The Version 8.1 4K file is retained, but it is not recognized by the DB2 UDB Version 8.2 product. Changes made to the 4K database configuration parameter file between migrating back to Version 8.1 and remigrating to Version 8.2 are, in effect, concealed from the more recent DB2 UDB level because the changes are not migrated to the existing 16K file.

db2diag.log format message enhancements

The db2diag.log file format has been improved in a number of ways for Version 8.2. The log file is now easier to read manually and easier to parse in software. The improvements include:

- Each entry has a complete set of header information
- Message and data portions of the logging record are clearly marked, making the logging record easier to use and to understand
- Timestamps with time zone are included for each record
- Each field has a clear field name in uppercase letters
- Header and message field line lengths are restricted to 80 characters
- New fields have been added, most notably a severity-level field to help you find the most important entries

Other changes have been made as well, such as changing the **database** field name to **DB**.

Event records have been added as diagnostic messages to the db2diag.log file. Examples of such events are:

- Updates to the profile registry
- Updates to configuration parameters

Event records have "Event" specified in the **LEVEL** field. Although events are not errors, they might be logged at diagnostic levels higher than 4 (Informational) or 3 (Warning) depending on their importance.

db2set profile registry variables and DB or DBM configuration parameters are now logged

Starting with Version 8.2, db2set profile registry updates and DB and DBM configuration parameters and are logged in the db2diag.log file. These messages are logged at high diagnostic levels due to their importance.

The following types of db2set profile registry updates are logged:

Modify
The `db2set` *`variableName=value`* command yields a db2diag.log entry such as:

```
2004-04-22-19.19.14.156959-240 I79582C286         LEVEL: Event
PID     : 2437242              TID  : 1           PROC : db2set
INSTANCE: db2user              NODE : 000
FUNCTION: DB2 UDB, oper system services, db2set_main, probe:40
CHANGE  : CFG DB2SET: DB2DBDFT: From: "OLDDB" To: "SAMPLE"
```

Delete The `db2set -r` command yields a db2diag.log entry such as:

```
CHANGE  : CFG DB2SET: DB2DBDFT: From: "SAMPLE" To: ""
```

Note: The header information is omitted in the preceding example.

Reset The `db2set variableName=value` command yields a db2diag.log entry such as:

```
CHANGE  : CFG DB2SET: Profile registry was reset
```

Note: The header information is omitted in the preceding example.

Examples for DB and DBM configuration parameter updates are

```
CHANGE  : CFG DB SAMPLE: "Maxlocks" From: "10" To: "20"

CHANGE  : CFG DBM: "Diaglevel" From: "3" To: "1"

CHANGE  : CFG DBM: Reset to the system defaults
```

Note: The header information is omitted in the preceding examples.

To find these configuration update messages, use db2diag tool. For example:

- To select all events: **db2diag -level event**
- To select change events: **db2diag -g "changeevent="**

Product compatibility

Microsoft XP fix is needed on 64-bit operating systems

If you are using the Microsoft® XP operating system (2600) configured to use the NETBIOS protocol with the DB2 family of products, you need to obtain a hotfix from Microsoft. Contact Microsoft with the Knowledge Base article number Q317437.

Windows XP operating systems

The Windows XP Home Edition operating system is supported only by DB2 Universal Database (UDB) Personal Edition products.

The Windows XP Professional operating system is supported by the following DB2 products:

- DB2 UDB Express Edition
- DB2 UDB Personal Edition
- DB2 UDB Workgroup Server Edition
- DB2 Connect™ Personal Edition
- DB2 Connect Enterprise Edition

Note: DB2 Connect Enterprise Edition is supported on Windows XP for development or test purposes only. Production environments require Windows 2000 or Windows Server 2003.

DB2 UDB HADR separately priced option available

In DB2 Universal Database (UDB) Version 8.2, customers of DB2 UDB Workgroup Server Edition and DB2 UDB Express Edition (when licensed based on per user pricing model) were not able to install the DB2 UDB High Availability Disaster Recovery (HADR) separately priced option. This problem has been fixed in DB2 UDB Version 8.2 FixPak 1 (equivalent to Version 8.1 FixPak 8).

DB2 Warehouse Manager (Version 8.2) and IBM DB2 OLAP Server FP3 and later

The OLAP utilities in DB2 Warehouse Manager Standard Edition, Version 8.2 are not compatible with IBM DB2 OLAP Server™ FP3 (Essbase API level 6.5.4) and later. You are advised to use DB2 OLAP Server FP2 (Essbase 6.5.3) or earlier until this problem is resolved.

Red Hat Linux support with the Data Warehouse Center

DB2 Universal Database, Version 8.2 supports Red Hat Enterprise Linux AS Versions 3.1 and 2.1. However, the Data Warehouse Center supports only Red Hat Enterprise Linux AS, Version 2.1. The Data Warehouse Center uses DataDirect ODBC drivers that do not support Red Hat Enterprise Linux AS, Version 3.1. Therefore, the Data Warehouse Center does not support ODBC warehouse sources and warehouse targets from a Red Hat Enterprise Linux AS, Version 3.1 agent site.

Alternative Unicode conversion tables for the coded character set identifier (CCSID) 5039

The Microsoft Japanese Windows Shift-JIS code page is registered as the IBM coded character set identifier (CCSID) 943. However, the Shift-JIS code page on HP-UX platform is registered as CCSID 5039. CCSID 5039 contains characters in the Japanese Industry Standard (JIS) only, and does not have any vendor defined characters. You can use a DB2 Universal Database (UDB) database of CCSID 5039 on HP-UX to store Shift-JIS characters, but there will be code page conversion between CCSID 5039 and CCSID 943. When using Microsoft ODBC applications, you might encounter a problem when converting data in CCSID 5039 to Unicode, due to differences between IBM's code page conversion table and Microsoft's code page conversion table.

The following list of characters, when converted from CCSID 5039 to Unicode, will result in different code points depending on which conversion table is used (IBM or Microsoft). For these characters, the IBM conversion table conforms to the Japanese Industry Standard JISX0208 and JISX0221.

Table 1. CCSID 5039 to Unicode code point conversion

Shift-JIS code point (character name)	**IBM primary code point (Unicode name)**	**Microsoft primary code point (Unicode name)**
X'815C' (EM dash)	U+2014 (EM dash)	U+2015 (Horizontal bar)
X'8160' (Wave dash)	U+301C (Wave dash)	U+FF5E (Fullwidth tilde)
X'8161' (Double vertical line)	U+2016 (Double vertical line)	U+2225 (Parallel to)
X'817C' (Minus sign)	U+2212 (Minus sign)	U+FF0D (Fullwidth hyphen-minus)

For example, the character EM dash with the CCSID 5039 code point of X'815C' is converted to the Unicode code point U+2014 when using the IBM conversion table, but is converted to U+2015 when using the Microsoft conversion table. This can create potential problems for Microsoft ODBC applications because they would treat U+2014 as an invalid code point. To avoid these potential problems, DB2 UDB provides the alternate Microsoft conversion table from CCSID 5039 to Unicode, in addition to the default IBM conversion table. You need to replace the default IBM conversion table with the alternate Microsoft conversion table. Note that the default IBM conversion table from Unicode to CCSID 5039 matches the Microsoft version.

Replacing the Unicode conversion tables for coded character set (CCSID) 5039 with the Microsoft conversion tables

When you convert from CCSID 5039 to Unicode, the DB2 Universal Database (UDB) default code page conversion table is used. If you want to use a different version of the conversion table, such as the Microsoft version, you must manually replace the default conversion table (.cnv) file.

Prerequisites:

Before replacing the existing code page conversion table file in the `sqllib/conv` directory, you should back up the file in case you want to change it back. On Linux, the `sqllib/conv` directory is linked to the DB2 UDB installation path.

Restrictions:

For conversion table replacement to be effective, every DB2 UDB client that connects to the same database must have its conversion table changed. Otherwise, the different clients might store the same character using different code points.

Procedure:

To replace the DB2 UDB default conversion table for converting from CCSID 5039 to Unicode, follow these steps:

1. Copy `sqllib/conv/ms/5039ucs2.cnv` to `sqllib/conv/5039ucs2.cnv`
2. Restart DB2 UDB.

Alternative Unicode conversion tables for the coded character set identifier (CCSID) 954

The IBM coded character set identifier (CCSID) for the Japanese EUC code page is registered as CCSID 954. CCSID 954 is a common encoding for Japanese Linux platforms. When using Microsoft ODBC applications to connect to a DB2 Universal Database (UDB) database of CCSID 954, you might encounter a problem when converting data from CCSID 954 to Unicode. The potential problem is due to differences between IBM's code page conversion table and Microsoft's code page conversion table. The IBM conversion table conforms to the character names as specified in the Japanese Industry Standard (JIS) JISX0208, JISX0212, and JISX0221.

The following characters, when converted from CCSID 954 to Unicode, will result in different code points depending on whether the IBM or Microsoft conversion table is used.

Table 2. CCSID 954 to Unicode code point conversion

EUC-JP code point (character name)	IBM primary code point (Unicode name)	Microsoft primary code point (Unicode name)
X'A1BD' (EM dash)	U+2014 (EM Dash)	U+2015 (Horizontal Bar)
X'A1C1' (Wave dash)	U+301C (Wave Dash)	U+FF5E (Fullwidth Tilde)
X'A1C2' (Double vertical line)	U+2016 (Double vertical line)	U+2225 (Parallel To)
X'A1DD' (Minus sign)	U+2212 (Minus sign)	U+FF0D (Fullwidth hyphen-minus)
X'8FA2C3' (Broken bar)	U+00A6 (Broken bar)	U+FFE4 (Fullwidth broken bar)

For example, the character EM dash with the CCSID 954 code point of X'A1BD' is converted to the Unicode code point U+2014 when using the IBM conversion table, but is converted to U+2015 when using the Microsoft conversion table. Due to this difference of conversion mapping, you might have two different code points for the same character in a DB2 UDB Unicode database, or in a graphic column of a DB2 UDB 954 database. This can create potential problems for Microsoft ODBC applications because they would treat U+2014 as an invalid code point. To avoid these potential problems, DB2 UDB provides the alternate Microsoft conversion table from CCSID 954 to Unicode, in addition to the default IBM conversion table. You need to replace the default IBM conversion table with the alternate Microsoft conversion table. Note that the default IBM conversion table from Unicode to CCSID 954 matches the Microsoft version.

Replacing the Unicode conversion tables for coded character set (CCSID) 954 with the Microsoft conversion tables

When you convert from CCSID 954 to Unicode, the DB2 Universal Database (UDB) default code page conversion table is used. If you want to use a different version of the conversion table, such as the Microsoft version, you must manually replace the default conversion table (.cnv) file.

Prerequisites:

Before replacing the existing code page conversion table file in the `sqllib/conv` directory, you should back up the file in case you want to change it back. On Linux, the `sqllib/conv` directory is linked to the install path of DB2 UDB.

Restrictions:

For this to be effective, every DB2 UDB client that connects to the same CCSID 954 database must have its conversion table changed. If your client is Japanese Windows, whose ANSI code page is Shift-JIS (CCSID 943), you will also need to change the DB2 default conversion tables between CCSID 943 and Unicode to the Microsoft version. Otherwise, the different clients might store the same character using different code points.

Procedure:

To replace the DB2 UDB default conversion table for converting from CCSID 954 to Unicode, follow these steps:

1. Copy `sqllib/conv/ms/0954ucs2.cnv` to `sqllib/conv/0954ucs2.cnv`
2. Restart DB2 UDB.

To replace the DB2 UDB default conversion tables for converting between CCSID 943 and Unicode, follow these steps:

1. Copy `sqllib/conv/ms/0943ucs2.cnv` to `sqllib/conv/0943ucs2.cnv`
2. Copy `sqllib/conv/ms/ucs20943.cnv` to `sqllib/conv/ucs20943.cnv`
3. Restart DB2 UDB.

Alternative Unicode conversion tables for the coded character set identifier (CCSID) 943

When using the Microsoft Japanese Windows Shift-JIS code page that is registered as the IBM coded character set identifier (CCSID) 943, you might encounter the following two problems when converting characters between CCSID 943 and

Unicode. The potential problem is due to differences between the IBM and Microsoft code page conversion tables. To avoid these potential problems, DB2 Universal Database (UDB) provides the alternate Microsoft conversion tables between CCSID 943 and Unicode, in addition to the default IBM conversion tables.

Problem 1:

For historical reasons, over 300 characters in the CCSID 943 code page are represented by two or three code points each. The use of input method editors (IMEs) and code page conversion tables cause only one of these equivalent code points to be entered. For example, the lower case character for Roman numeral one 'i' has two equivalent code points: X'EEEF' and X'FA40'. Microsoft Windows IMEs always generate X'FA40' when 'i' is entered. In general, IBM and Microsoft use the same primary code point to represent the character, except for the following 13 characters:

Table 3. CCSID 943 Shift-JIS code point conversion

Character name (Unicode code point)	IBM primary Shift-JIS code point	Microsoft primary Shift-JIS code point
Roman numeral one (U+2160)	X'FA4A'	X'8754'
Roman numeral two (U+2161)	X'FA4B'	X'8755'
Roman numeral three (U+2162)	X'FA4C'	X'8756'
Roman numeral four (U+2163)	X'FA4D'	X'8757'
Roman numeral five (U+2164)	X'FA4E'	X'8758'
Roman numeral six (U+2165)	X'FA4F'	X'8759'
Roman numeral seven (U+2166)	X'FA50'	X'875A'
Roman numeral eight (U+2167)	X'FA51'	X'875B'
Roman numeral nine (U+2168)	X'FA52'	X'875C'
Roman numeral ten (U+2169)	X'FA53'	X'875D'
Parenthesized ideograph stock (U+3231)	X'FA58'	X'FA58'
Numero sign (U+2116)	X'FA59'	X'8782'
Telephone sign (U+2121)	X'FA5A'	X'8754'

IBM products such as DB2 UDB primarily use IBM code points, such as X'FA4A' to present the upper case Roman numeral one 'I', but Microsoft products use X'8754' to represent the same character. An Microsoft ODBC application can insert the 'I' character as X'8754' into a DB2 UDB database of CCSID 943, and the DB2 UDB Control Center can insert the same character as X'FA4A' into the same CCSID 943 database. However, ODBC applications can find only those rows that have 'I' encoded as X'8754', and DB2 UDB Control Center can locate only those rows that have 'I' encoded as X'FA4A'. To enable DB2 UDB Control Center to select 'I' as X'8754', you need to replace the default IBM conversion tables between CCSID 943 and Unicode with the alternate Microsoft conversion tables.

Problem 2:

The following list of characters, when converted from CCSID 943 to Unicode, will result in different code points depending on whether the IBM conversion table or the Microsoft conversion table is used. For these characters, the IBM conversion table conforms to the Japanese Industry Standard JISX0208, JISX0212, and JISX0221.

Table 4. CCSID 943 to Unicode code point conversion

Shift-JIS code point (character name)	IBM primary code point (Unicode name)	Microsoft primary code point (Unicode name)
X'815C' (EM dash)	U+2014 (EM dash)	U+2015 (Horizontal bar)
X'8160' (Wave dash)	U+301C (Wave dash)	U+FF5E (Fullwidth tilde)
X'8161' (Double vertical line)	U+2016 (Double vertical line)	U+2225 (Parallel to)
X'817C' (Minus sign)	U+2212 (Minus sign)	U+FF0D (Fullwidth hyphen-minus)
X'FA55' (Broken bar)	U+00A6 (Broken bar)	U+FFE4 (Fullwidth broken bar)

For example, the character EM dash with the CCSID 943 code point of X'815C' is converted to the Unicode code point U+2014 when using the IBM conversion table. However, it is converted to U+2015 when using the Microsoft conversion table. Due to this difference of conversion mapping, you might have two different code points for the same character in a DB2 UDB Unicode database. This can create potential problems for Microsoft ODBC applications because they would treat U+2014 as an invalid code point. To avoid this potential problem, you need to replace the default IBM conversion tables between CCSID 943 and Unicode with the alternate Microsoft conversion tables.

The use of the alternate Microsoft conversion tables between CCSID 943 and Unicode should be restricted to closed environments, where the DB2 UDB clients and the DB2 UDB databases all have a code page of CCSID 943 and are all using the same alternate Microsoft conversion tables. If you have a DB2 UDB client using the default IBM conversion tables, and another DB2 UDB client using the alternate Microsoft conversion tables, and both clients are inserting data to the same DB2 UDB database of CCSID 943, the same character may be stored as different code points in the database.

Replacing the Unicode conversion tables for coded character set (CCSID) 943 with the Microsoft conversion tables

When you convert between CCSID 943 and Unicode, the DB2 Universal Database (UDB) default code page conversion tables are used. If you want to use a different version of the conversion tables, such as the Microsoft version, you must manually replace the default conversion table (.cnv) files.

Prerequisites:

Before replacing the existing code page conversion table files in the `sqllib/conv` directory, you should back up the files in case you want to change them back. On Linux, `sqllib/conv` is linked to the DB2 UDB installation path.

Restrictions:

For conversion table replacement to be effective, every DB2 UDB client that connects to the same database must have its conversion table changed. Otherwise the different clients might store the same character using different code points.

Procedure:

To replace the DB2 UDB default conversion tables for converting characters between CCSID 943 and Unicode:

1. Copy `sqllib/conv/ms/0943ucs2.cnv` to `sqllib/conv/0943ucs2.cnv`.
2. Copy `sqllib/conv/ms/ucs20943.cnv` to `sqllib/conv/ucs20943.cnv`.
3. Restart DB2 UDB.

Installation, migration, upgrade, and configuration information

Hardware and software requirements

Hardware and software requirements can be viewed at the following Web site:

http://www.ibm.com/software/data/db2/udb/sysreqs.html

Installation notes

On Linux platforms, these Installation notes apply to installation of new products only. To apply a FixPak, see the FixPak's Readme.

On Windows platforms, these Installation notes apply to both the installation of new products and the application of FixPaks. If you are applying a FixPak, these Installation notes should be used in conjunction with the FixPak Readme.

Review this entire section before installing DB2 Universal Database.

Directory path cannot contain blanks (Linux)

If the db2setup program is run from a directory whose path includes a blank, the setup will fail with the following error:

```
<file>: not found
```

Place the installable image in a directory whose path does not include spaces.

JDK levels for DB2 UDB (Linux on IA64 and Linux on PowerPC)

When you install Version 8.2 of DB2 Universal Database (UDB) on Linux, the RPM-based installation attempts to install the IBM Java™ RPM (IBMJava2–SDK-1.4.1.-2.0.i386.rpm).

Note: DB2 UDB on Linux for IA64 only supports Java 1.3.1. For the most recent Linux SDK information, see the IBM developer kit for Linux Web page at http://www.ibm.com/developerworks/java/jdk/linux/tested.html.

If a later level of the RPM (such as IBMJava2-SDK-1.5.0.-2.0.i386.rpm) already exists, the back-level RPM is not installed.

However, in this case, the installation leaves the JDK_PATH database configuration parameter pointing to the Java 1.4 path, `/opt/IBMJava2-14/`. As a result, none of the Java-dependant functionality, including the installation of the DB2 Tools Catalog, will work.

Prerequisite:

You must run the following command as the instance owner.

Procedure:

1. To point DB2 UDB to the correct IBM Developer Kit, run the following command:

© Copyright IBM Corp. 2002 - 2004

```
db2 update dbm cfg using JDK_PATH path
```

where *path* the 1.5 install path, such as /opt/IBMJava2-15.

Uncompressing installation images (Linux)

Some installation images are delivered in compressed or gzipped format on the product CDs. Before you can install DB2 Universal Database (UDB) from these formats, you need to copy the installation image to a temporary directory and uncompress or gunzip the installation image.

The compressed or gzipped installation images on the CD have the file name PRODUCT.tar.Z or PRODUCT.tar.gz, where PRODUCT is the DB2 product you are installing. For example, if you are installing DB2 UDB Enterprise Server Edition, the compressed image on the relevant CD might be called either ese.tar.Z or ese.tar.gz.

In addition to the software disk requirements, you must have a file system with 2 gigabytes of free space to contain the tar.Z or tar.gz file and the uncompressed installation image.

Procedure:

To uncompress the installation images, perform the following steps:

1. Copy the compressed or gzipped installation image to a temporary file system containing at least 2 gigabytes of free space.
2. Change to the directory where you copied the installation image by entering **cd /TMP**, where /TMP represents the directory where you copied the compressed image.
3. If the product has the *.tar.Z extension, enter the following command to uncompress the installation image:

   ```
   zcat PRODUCT.tar.Z | tar -xvf -
   ```

   where PRODUCT is the DB2 product you are installing.
4. If the product has the *.tar.gz extension, enter the following command to uncompress the installation image:

   ```
   gunzip -c PRODUCT.tar.gz | tar -xvf -
   ```

   where PRODUCT is the DB2 product you are installing.

   Notes:

   a. gunzip is part of the AIX 5L™ default installation setup. If you do not have gunzip, install the rpm.rte file set from the AIX 5L installation media. The rpm.rte file set contains gunzip.
   b. You can also download gzip for AIX 5L, which includes gunzip, from http://www.ibm.com/servers/aix/products/aixos/linux/rpmgroups.html.
5. To perform the installation using the DB2 Setup wizard, see the *Installing DB2 UDB from installation images (Linux)* section.

DB2 UDB does not allow non-English installation path (Windows)

During DB2 Universal Database (UDB) installation on Windows, you can enter the directory path for DB2 UDB. However, the directory path name you enter can contain only English characters.

Downloading and uncompressing FixPaks for multiple products (Windows)

Starting with DB2 Version 8 for Windows Fixpak 3, IBM is providing product-specific FixPaks instead of one general FixPak. This change affects only DB2 Version 8 products on Windows platforms.

For example, if you have DB2 UDB Enterprise Server Edition Version 8 and DB2 Spatial Extender Version 8 installed on the same Windows system, you must download the DB2 UDB Enterprise Server Edition FixPak image and the Spatial Extender FixPak image. You must uncompress each image under a common parent directory. All images must be uncompressed for the GUI installation or silent installation to proceed.

For complete FixPak installation instructions, see the latest DB2 UDB FixPak Readme.

Installing DB2 UDB from installation images (Linux)

Prerequisites:

Before you start the DB2 Setup wizard:

- Ensure your system meets installation, memory, and disk requirements. See the *Hardware and software requirements* section.
- Ensure you have read the Installation Prerequisites. The file install.txt or install.htm is located on the CD at: */cdrom*/doc/*language*/ where:
  - *cdrom* refers to the mount point
  - *language* refers to the language directory consisting of a five-character code.
- You require root authority to perform the installation.
- The DB2 product CD must be mounted on your system.

Procedure:

If the installation images are delivered in compressed or gzipped format on the product CDs, you must uncompress or gunzip the installation image before you can install DB2 UDB. See the *Uncompressing installation images (Linux)* section for details.

To perform the installation using the DB2 Setup wizard, enter the following command:

```
./PRODUCT/db2setup
```

where `PRODUCT` is the DB2 product you are installing. For example, if you are installing DB2 UDB Enterprise Server Edition, enter ./ese/db2setup.

The installation Launchpad opens. Proceed through the installation panels.

Installing DB2 UDB from installation images (Windows)

Prerequisites:

Before you start the DB2 Setup wizard:

- Ensure your system meets installation, memory, and disk requirements. See the *Hardware and software requirements* section.

- Ensure you have read the Installation Prerequisites. The file install.txt or install.htm is located on the CD at: *x*:\doc*language*\ where:
  - *x* represents the CD drive
  - *language* refers to the language directory consisting of a five-character code.
- You must have a local Administrator user account with the recommended user rights to perform the installation.

To perform the installation using the DB2 Setup wizard, insert the CD into the drive. If enabled, the auto-run feature automatically starts the DB2 Setup launchpad.

Procedure:

If the auto-run feature is not enabled, you can start the DB2 Setup wizard manually:

1. Click **Start** and select the **Run** option.
2. In the **Open** field, enter the following command:

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

   where:
   - *x:* represents the CD drive
   - *language* is the territory identifier for your language (for example, EN for English).

   The `/i language` parameter is optional. If it is not specified, the DB2 Setup wizard runs in the same language as your operating system.
3. Click **OK**. The DB2 Setup launchpad opens.

Proceed through the installation panels.

Disk space required for a DB2 UDB response file installation

When you install a DB2 Universal Database product using a response file, 1 MB of free disk space is required in the `etc` directory.

If you receive error message DBI1191I during a response file installation, the message indicates there is not enough free disk space in the root directory. This message is misleading. Check the amount of free disk space in the `etc` directory. A minimum of 1 MB of free disk space is required before re-running the installation.

Restrictions to adding products using the db2setup command (Linux)

Once a DB2 product has been installed, additional DB2 products can be added. If you use the **db2setup** command to add products the following recommendations and restrictions apply.

Recommendations:

Both the installed product and the product being added should be at the same code level. For example, DB2 Universal Database (UDB) Enterprise Server Edition Server Version 8 FixPak 5 is already installed and you want to add the DB2 Information Integrator product. In this case, DB2 Information Integrator should also be at the Version 8 FixPak 5 level.

Restrictions:

- If the DB2 fixpak level is higher than the fixpak level of the product being added, the combination is allowed. However, since the fixpak level of the product being added is lower than the DB2 fixpak level, the DB2 fixpak level must be reapplied after installing the additional product. See the appropriate FixPak Readme for instructions to reapply the fixpak.
- If the DB2 fixpak level is lower than the fixpak level of the product being added, an error is generated. The product being added cannot be at a higher fixpak level than DB2 UDB. In this case, you must first bring DB2 UDB to the appropriate level, then install the additional product. See the appropriate FixPak Readme for instructions.

The following table summarizes the db2setup combinations:

Table 5. db2setup combinations

DB2 fixpak level	Additional product fixpak level	Is this combination permitted?
Version 8 FixPak 3	Version 8 FixPak 3	Yes. This is recommended.
Version 8 FixPak 3	Version 8 GA	Yes, but Version 8 FixPak 3 must be reapplied. See the appropriate FixPak Readme for instructions to reapply the fixpak by going to the DB2 Support Web site.
Version 8 FixPak 3	Version 8 FixPak 5	No. DB2 must first be brought to the higher fix pack level (in this example, Version 8 FixPak 5) before installing the additional product. See the appropriate FixPak Readme for instructions to install the required Version 8 FixPak by going to the DB2 Support Web site.

The DB2 Support Web site address is
http://www.ibm.com/software/data/db2/udb/support.html

DB2 Web Tools

The application servers supported by DB2 Web Tools for the following languages are required to be compliant with the Servlet 2.3 specification:

- Japanese
- Korean
- Simplified Chinese
- Traditional Chinese
- Russian
- Polish

Recreating beta versions of databases

If you created databases in a beta version of DB2 UDB Version 8.2, you must recreate them in the official version of Version 8.2.

This includes recreation of spatial databases for of DB2 Geodetic Extender Version 8.2.

Installing MDAC files for national language versions of DB2 UDB

If you do not install the national language version of Microsoft Data Access Components (MDAC) 2.7 prior to installing the national language version of DB2 Universal Database (UDB) Version 8.2, DB2 UDB installs English MDAC files by default. This installation causes the Windows ODBC Data Source Administrator panels to appear untranslated if your operating system is in a language other than English. To fix this problem, install the "MDAC 2.7 RTM - Refresh" bundle from the Microsoft Web site at http://msdn.microsoft.com/data/downloads/updates/default.aspx.

Choose the language that you want to install, download the required executable file, and run it. This procedure installs the translated ODBC Data Source Administrator files.

Installing additional Asian fonts (Linux)

IBM offers additional font packages for Linux that contain additional double-byte character set (DBCS) support for Asian characters. These font packages are necessary with some versions of Linux that install only the fonts required to display the country-specific or region-specific characters.

If you run the **db2setup** command and find missing characters in the DB2 setup wizard interface, it is likely that your Linux system does not have all the necessary fonts installed. To enable the **db2setup** command to properly refer to the fonts embedded in the installation CD, perform the following task:

1. Enter the following command:

   ```
   export JAVA_FONTS=/<cdrom>/db2/<linux_platform>/java/jre/lib/fonts
   ```

   where <cdrom> is the location of the install image and <linux_platform> is a directory name with a *Linux* prefix.
2. Re-run the **db2setup** command.

If you notice missing characters when using the DB2 GUI tools after installation, install the necessary fonts provided with the DB2 product. These fonts can be found in the `fonts` directory on any of the following CDs:

- *IBM Developer Kit, Java Technology Edition (64-bit) for AIX 5L*
- *DB2 Embedded Application Server and applications (XML registry, Web Administration tools and Java distributed debugger)* for your operating system

In the `fonts` directory, there are two typefaces available: Times New Roman WorldType and Monotype Sans Duospace WorldType. For each typeface, there is a country-specific or region-specific font. The following table lists the eight fonts provided in compressed format in the `fonts` directory.

Table 6. File names for additional Asian fonts

Font typeface	Font file name	Country or Region
Times New Roman WT J	tnrwt_j.zip	Japan and other countries or regions
Times New Roman WT K	tnrwt_k.zip	Korea
Times New Roman WT SC	tnrwt_s.zip	China (Simplified Chinese)
Times New Roman WT TC	tnrwt_t.zip	Taiwan (Traditional Chinese)

Table 6. File names for additional Asian fonts (continued)

Font typeface	Font file name	Country or Region
Monotype Sans Duospace WT J	mtsansdj.zip	Japan and other countries or regions
Monotype Sans Duospace WT K	mtsansdk.zip	Korea
Monotype Sans Duospace WT SC	mtsansds.zip	China (Simplified Chinese)
Monotype Sans Duospace WT TC	mtsansdt.zip	Taiwan (Traditional Chinese)

Note: These fonts do not replace the system fonts. These fonts are to be used in conjunction with or for use with DB2 Universal Database. You cannot engage in the general or unrestricted sale or distribution of these fonts.

Procedure:

To install an additional Asian font:

1. Unzip the font package.
2. Copy the font package to the `/opt/IBMJava2-141/jre/lib/fonts` directory. You need to create the directory if it does not already exist.
3. Enter the following command:

   ```
   export JAVA_FONTS=/opt/IBMJava2-141/jre/lib/fonts
   ```

As a minimum, you need to install one font of each typeface for your country or region. If you are in China, Korea, or Taiwan, use the country-specific or region-specific versions; otherwise, use the Japanese version of the fonts. If you have space on your system, install all eight fonts.

Configuring the Development Center to use Java Development Kits (Linux)

In some cases, DB2 Universal Database does not install a Java Development Kit on the client operating system. To use the Development Center to develop Java stored procedures on these clients, you must point the Development Center to the location of an installed Java Development Kit .

Follow these steps to identify the location of a Java Development Kit:

1. In Development Center, select the **Project** -> **Environment Settings** menu item.
2. In the Environment Settings notebook, select the **Process** node.
3. In the **Java Home** section of the Process page, select the **JDK level** that will be used to build and run Java stored procedures.
4. In the **Directory** field, specify a directory path that exists or is accessible on the client where the selected JDK is installed.
5. If the client computer is used to develop Java stored procedures on multiple DB2 servers, you might need to select additional JDK levels and specify their installed locations, depending on which JDK levels are used by these servers.

On the DB2 server, the Java Development Kit installation might not have linked some Java libraries into the system `/usr/lib` subdirectory. These links are needed to build and run Java stored procedures.

The following example shows how to create the links to a Java Development Kit on a Linux client:

```
 # Setup the links to the .so files
cd /usr/lib
ln -s /opt/IBMJava2-131/jre/bin/libhpi.so
ln -s /opt/IBMJava2-131/jre/bin/libjava.so
ln -s /opt/IBMJava2-131/jre/bin/libjitc.so
ln -s /opt/IBMJava2-131/jre/bin/libxhpi.so
ln -s /opt/IBMJava2-131/jre/bin/libzip.so
ln -s /opt/IBMJava2-131/jre/bin/classic/libjvm.so
```

Creating group and user IDs on United Linux 1.0 and SuSE Linux distributions

To create group and user IDs for a DB2 UDB Version 8.2 installation on United Linux 1.0 and SuSE Linux distributions, use the `groupadd` and `useradd` commands, respectively. The Version 8.2 *Installation and Configuration Supplement* incorrectly documents the `mkgroup` and `mkuser` commands to create group and user IDs, respectively.

Help system daemon does not start after installing with the db2_install command (Linux)

If you are installing the DB2 Information Center on Linux platforms using the db2_install command, the help system daemon (db2icd) does not start following the installation. You need to manually start the help system daemon or restart your computer to access the documentation.

See the topic titled "DB2 Information Center daemon" in the **Documentation updates | Installation and configuration** section of the *Release Notes*.

Installation image reduction tool (Windows)

The db2iprune command line utility can reduce the size of your DB2 Universal Database (UDB) Windows product installation image. The tool is particularly useful for large scale deployments of DB2 UDB. The tool is also useful for embedding DB2 UDB within an application.

The db2iprune tool consists of an input file and an executable file. The input file (.prn), contains a full list of removable components and is used to indicate which features and languages you would like to remove from the installation image. The db2iprune executable file (db2iprune.exe) then removes the cabinet files associated with those features and languages. The result is a new, smaller DB2 UDB image that can be installed using the regular installation methods. Fixpaks are also applied using the regular methods. When the fixpak is installed it detects and updates only the components that were installed using db2iprune.

The db2iprune tool is located in the `\db2\windows\utilities\db2iprune` directory on your DB2 UDB product installation CD. This directory also contains a Readme file. Refer to the db2iprune Readme for detailed instructions on how to use db2iprune.

DB2 Universal Database, Version 8 HTML documentation installation restriction (Windows)

On Windows, do not install the DB2 Universal Database (UDB), Version 8 HTML documentation on a workstation or server where a DB2 UDB, Version 7 (or earlier) product is already installed. The installer detects the earlier version and removes the earlier product.

Previous installations updated to the latest level (Windows)

If you have a DB2 product that is installed at an earlier Version 8 level, the installation image detects this and updates the product to the latest level.

System requirements for the DB2 .NET Data Provider (Windows)

Before using the DB2 Universal Database (UDB) installation program to install the DB2 .NET Data Provider, you must have the .NET Framework already installed on the computer. If the .NET Framework is not installed, the DB2 UDB installation program will not register the DB2 .NET Data Provider.

If DB2 UDB is installed and the .NET Framework is not installed, the DB2 .NET Data Provider is not registered. If the .NET Framework is installed at a later date, you can run the db2nmpreg executable to register the provider. This executable resides in the sqllib\bin directory. There are no parameters.

To register DB2 .NET Data Provider, enter db2nmpreg from any command window.

Installing DB2 Version 8 Clients and DB2 Connect PE as a non-Administrator (Windows)

When installing a DB2 Administration client, DB2 Application Development client, or DB2 Connect Personal Edition, you must update your Windows TCP/IP services file if the following conditions apply:

- The data warehousing feature is selected in the installation setup
- The user ID performing the installation is not a member of the Administrators group on the target computer
- The product is being installed on any one of the following operating systems: Windows NT®, Windows 2000, Windows XP, or Windows Server 2003.

If all of the these conditions apply, the following entries need to be added to the Windows TCP/IP services file:

Table 7. Entries required in Windows TCP/IP services file

Port name	Port number
vwkernel	11000/tcp
vwd	11001/tcp
vwlogger	11002/tcp

Without these entries, the Data Warehouse Center will not function properly.

Migration notes

Migrating DB2 Universal Database (Windows)

The following steps show the correct order for the migration prerequisites for DB2 Universal Database (UDB) on Windows.

Prerequisites:

Before migrating:

1. Review the migration recommendations, restrictions, and space recommendations.
2. Record configuration settings before DB2 UDB migration.
3. Change the diagnostic error level.
4. Verify that databases are ready for DB2 UDB migration.
5. Back up your databases.
6. If you are using replication, you must archive all of the DB2 UDB log files.
7. You must have SYSADM authority.
8. Take the DB2 server offline for DB2 UDB migration.

Note: As of Version 8.1 the DB2_HASH_JOIN registry variable will be ON by default. See the "Administration: Performance" subsection in the "Documentation updates" section for details.

Migrating DB2 Universal Database when using DataJoiner or replication

If you want to migrate an instance of DataJoiner® or DB2 Universal Database (UDB) for Linux, and Windows on which you are running the Capture or Apply programs for DB2 UDB replication, you must prepare to migrate your replication environment before you migrate the DB2 UDB or DataJoiner instance. Detailed instructions for carrying out the required preparation are included in the migration documentation for DB2 DataPropagator™, Version 8. You can find migration documentation for DB2 DataPropagator, Version 8 at http://www.ibm.com/software/data/dpropr/.

Migrating a DB2 Version 8 Windows 32–bit database to Windows 64-bit

This topic lists the steps for migrating your 32–bit DB2 Version 8 database on a 32–bit computer to a 64–bit database on a 64–bit Windows operating system.

Prerequisites:

- A 64-bit version of DB2 Version 8 must be installed on your 64-bit computer.
- Ensure that your 32-bit Windows system is running DB2 Version 8.

Procedure:

To migrate to DB2 Version 8 on Windows 64-bit:

1. Backup your DB2 Version 8 databases on your 32-bit Windows system.
2. Restore your DB2 Version 8 backup (created in step #1) on your 64-bit Windows system.

Note: In addition to migrating DB2 UDB from 32-bit systems to 64-bit systems, the following migrations are also possible:

- Migrating between versions of Windows
- Migrating between versions of DB2 UDB
- Migrating everything at once
- Migrating back to 32–bit

Detailed information is available in the following IBM Redbook: Scaling DB2 UDB on Windows Server 2003. This Redbook can be found at the following URL:

http://publib-b.boulder.ibm.com/Redbooks.nsf/RedbookAbstracts/sg247019.html

Migrating XML Extender from previous versions

If you are using an earlier version of DB2 XML Extender, you must migrate each database that is enabled for XML Extender before you use an existing XML-enabled database with the updated XML Extender release. Each new fix pack contains all of the previous fix packs' updates.

Before running the migration program, create a backup of the database.

To migrate an XML enabled database and XML enabled columns, complete the following steps.

1. From the DB2 Command Line, enter:

   ```
   db2 connect to database_name
   db2 bind dxxinstall/bnd/@dxxMigv.lst
   db2 bind dxxinstall/bnd/@dxxbind.lst
   ```

   where *dxxinstall* is the directory path in which you installed DB2 Universal Database.

2. From the DB2 Command Line, enter:

   ```
   dxxMigv database_name
   ```

Uninstallation information

Uninstalling DB2 UDB silently (Windows)

To remove DB2 products silently using the **msiexec** command:

```
msiexec /x <product_code> /qn
```

where <product code> represents the code for the product you want to remove.

Here is a list of the DB2 product codes:

DB2 Universal Database Enterprise Server Edition (ESE)
{D8F53726-C7AD-11D4-9155-00203586D551}

DB2 Universal Database Workgroup Server Edition (WSE)
{7A28F948-4945-4BD1-ACC2-ADC081C24830}

DB2 Universal Database Express Edition (EXP)
{58169F10-CA30-4F40-8C6D-C6DA8CE47E16}

DB2 Universal Database Personal Edition (PE)
{C0AA883A-72AE-495F-9601-49F2EB154E93}

DB2 Warehouse Manager (WM)
{84AF5B86-19F9-4396-8D99-11CD91E81724}

DB2 Data Links Manager (DLM)
{1D16CA65-F7D9-47E5-BB26-C623A44832A3}

Relational Connect (RCON)
{273F8AB8-C84B-4EE6-85E7-D7C5270A6D08}

DB2 Connect Enterprise Edition (CEE)
{9C8DFB63-66DE-4299-AC6B-37D799A728A2}

DB2 Connect Personal Edition (CPE)
{F1912044-6E08-431E-9B6D-90ED10C0B739}

DB2 Administration Client (ADMCL)
{ABD23811-AA8F-416B-9EF6-E54D62F21A49}

DB2 Application Development Client (ADCL)
{68A40485-7F7F-4A91-9AB6-D67836E15CF2}

DB2 Run-Time Client (RTCL)
{63F6DCD6-0D5C-4A07-B27C-3AE3E809D6E0}

DB2 Run-Time Client Lite (RTLITE)
{07C9CEE7-4529-4E60-95D3-6B6EF6AC0E81}

DB2 Eclipse Documentation (DOCE)
{FE2D4758-041C-4E4E-95B3-529E4E1EAF3E}

DB2 Query Patroller (QP)
{7A8BE511-8DF3-4F22-B61A-AF0B8755E354}

Life Sciences Data Connect (LSDC)
{DD30AEB3-4323-40D7-AB39-735A0523DEF3}

DB2 Cube Views (CUBE)
{C8FEDF8F-84E8-442F-A084-0A0F6A772B52}

© Copyright IBM Corp. 2002 - 2004

DB2 Spatial Extender (SE)
{F6846BF9-F4B5-4BB2-946D-3926795D5749}

Example:

If you want to remove DB2 UDB Enterprise Edition, enter the following command

```
msiexec /x <product_code> /qn
```

The following DB2 product codes are no longer supported in DB2 UDB Version 8.2:

- WMC {5FEA5040-22E2-4760-A88C-73DE82BE4B6E}
- DOC {73D99978-A255-4150-B4FD-194ECF4B3D7C}

Product code for DB2 Information Center removal using a silent uninstall (Windows)

When removing the DB2 Information Center using a silent uninstall on Windows, use the following product code:

{FE2D4758-041C-4E4E-95B3-529E4E1EAF3E}

Limitations

IMPORT REPLACE does not honor the Not Logged Initially clause

The IMPORT command's REPLACE option does not honor the CREATE TABLE statement's NOT LOGGED INITIALLY (NLI) clause or the ALTER TABLE statement's ACTIVATE NOT LOGGED INITIALLY clause.

If an import with the REPLACE action is performed within the same transaction as a CREATE TABLE or ALTER TABLE statement where the NLI clause is invoked, the import will not honor the NLI clause. All inserts will be logged.

Workaround 1

Delete the contents of the table using the DELETE statement, then invoke the import with INSERT statement.

Workaround 2

Drop the table and recreate it, then invoke the import with INSERT statement.

This limitation applies to DB2 UDB Version 7 and DB2 UDB Version 8.

Data export with ODBC to file warehouse program

The Data export with ODBC to file warehouse program does not support the following Sybase data types:

- BIT
- BINARY
- VARBINARY

Data types that are not supported in the Development Center's integrated SQL debugger

The following data types are not supported in the SQL Debugger that is integrated into Development Center:

- For DB2 Universal Database (UDB) for Linux and Windows, the CHAR FOR BIT DATA,VARCHAR FOR BIT DATA, and GRAPHIC data types are not supported.
- For DB2 UDB for z/OS, the GRAPHIC data type is not supported and the BLOB and CLOB data types, used as parameters, are not supported. BLOB and CLOB used as local variables are supported.

Structured types in the Development Center

The Development Center no longer supports the creation of structured types.

Development Center limitations for 64–bit operating systems

Debugging of Java stored procedures against a 64-bit server is not supported by the Development Center. Debugging SQL stored procedures is supported on 64-bit Windows operating systems. Developing OLE DB or XML User Defined Functions is not supported for 64-bit servers.

© Copyright IBM Corp. 2002 - 2004

Development Center (Linux)

You cannot use the Development Center to debug Java stored procedures running on any of the Linux distributions (32–bit, 64–bit, Intel™, zSeries®, or iSeries™).

Debugging stored procedures with double quotation marks

The Development Center does not support debugging for any stored procedure with double quotation marks (") in the stored procedure name, schema, or specific name.

Path settings needed to enable Java routines to compile in the Development Center

The Development Center cannot compile Java routines unless it knows where your developer kit versions are installed. Default directories for your developer kit versions are written to your `$HOME/.ibm/db2/dc/DB2DC.settings` file when the Development Center starts for the first time. You can copy these directories into your `$USER.settings` file and modify them with a Unicode editor, or you can create symbolic links to your developer kit directories in the default locations.

Development Center limitations to run and debug Java stored procedures simultaneously

The Development Center does not support running and debugging Java stored procedures simultaneously. In the Development Center, you can either run multiple Java stored procedures at the same time or debug a single Java stored procedure; you cannot run a Java stored procedure while debugging another Java stored procedure. By default, the KEEPFENCED database manager configuration keyword is set to KEEPFENCED=YES as required for debugging SQL stored procedures. When the keyword KEEPFENCED has the default value YES, the routine process is kept active and JVM port conflicts will occur. Java stored procedures will cause a JVM exception and will fail in any of the following circumstances when the default KEEPFENCED=YES database manager configuration setting is used:

- Building a Java stored procedure in the Development Center then debugging it
- One user running a Java stored procedure and another user debugging a Java stored procedure while the initial Java stored procedure is still running
- One user debugging a Java stored procedure and another user running a Java stored procedure while the initial Java stored procedure is being debugged

To work around this limitation, ensure that the KEEPFENCED database manager configuration keyword is set to KEEPFENCED=NO by running the following commands:

```
db2 update dbm cfg using KEEPFENCED NO
db2stop
db2start
```

When KEEPFENCED is set to NO, the db2fmp process will shut down when a Java stored procedure call is finished, and DB2 Universal Database will start a new db2fmp process to handle the next routine call. This ensures that there will be no existing JVM in debug mode when you start to debug a Java stored procedure.

The KEEPFENCED=YES setting is required to build SQL stored procedures for debugging and to debug SQL stored procedures. When KEEPFENCED=NO, you can still build and execute SQL stored procedures but you cannot debug them.

Cursors in PHP applications

When the PHP interpreter creates a cursor on behalf of an application, it is created by default as a Scrollable Keyset-driven cursor. In certain cases, this might cause unexpected results to be returned. In order to avoid this situation, explicitly specify the "FOR READ ONLY" clause for all SELECT statements that are used to update data. Other alternatives include setting the CLI Configuration parameters "Patch2=6", "Patch2=42", or "DisableKeysetCursor=1". However, each of these might have other consequences. Refer to the *CLI Guide and Reference* documentation for details regarding these configuration keywords.

Four unsupported SQL administrative routines

The following SQL administrative routines are not supported in this release:

- APP procedure
- INSTALLAPP procedure
- SERVER procedure
- UNINSTALLAPP procedure

Bind option limitations for CLI packages

Some bind options might not take effect when binding CLI packages with any of the following list files: `db2cli.lst`, `ddcsmvs.lst`, `ddcs400.lst`, `ddcsvm.lst`, or `ddcsvse.lst`. Because CLI packages are used by CLI, ODBC, JDBC, OLE DB, .NET, and ADO applications, any changes made to the CLI packages affect all applications of these types. Only a subset of bind options are therefore supported by default when binding CLI packages. The supported options are: ACTION, COLLECTION, CLIPKG, OWNER, and REPLVER. All other bind options that impact CLI packages are ignored.

To create CLI packages with bind options that are not supported by default, specify the COLLECTION bind option with a collection ID that is different from the default collection ID, NULLID. Any bind options specified are then accepted. For example, to create CLI packages with the KEEPDYNAMIC YES bind option, which is not supported by default, issue the following command:

```
db2 bind @db2cli.lst collection newcolid keepdynamic yes
```

In order for CLI/ODBC applications to access the CLI packages created in the new collection, set the CurrentPackageSet CLI/ODBC keyword in the `db2cli.ini` initialization file to the new collection ID.

To overwrite CLI packages that already exist under a particular collection ID, perform either of the following actions:

- Drop the existing CLI package before issuing the bind command for this collection ID
- Specify the ACTION REPLACE bind option when issuing the bind command

CLI LOAD restriction for specifying column names

If the INSERT statement supplied to the CLI LOAD utility includes a VALUES clause, target columns cannot be specified. For example, the following statement is supported by CLI LOAD:

```
INSERT into tableA VALUES (?, ?, ?)
```

This statement, however, that specifies target columns is not supported for use with the CLI LOAD utility:

```
INSERT into tableA (col1, col2, col3) VALUES (?, ?, ?)
```

Tools catalog database creation not supported (Linux for AMD64)

The creation of the tools catalog database under a 64-bit DB2 Universal Database (UDB) instance on Linux (AMD64) is not supported. Do not attempt to create the tools catalog under a 64-bit instance using any of the following methods:

- Installing DB2 UDB
- Updating a 64-bit instance using the **db2isetup** command
- Issuing the **CREATE TOOLS CATALOG** CLP command after the installation is finished.

The creation of the tools catalog database under a 32-bit instance on Linux (AMD64) is supported as of Version 8.1.4.

Memory limitations for DB2 UDB Express and DB2 Workgroup Server Edition V8.2

There are memory limitations for the following products:

- IBM DB2 Universal Database (UDB) Express Edition (Named user option). The maximum memory per server is 4GB.
- IBM DB2 UDB Express Edition (CPU option). The maximum memory per server is 4GB.
- IBM DB2 UDB Workgroup Server Edition. The maximum memory per server is 16GB.
- IBM DB2 UDB Workgroup Server Unlimited Edition. The maximum memory per server is 16GB.

Scheduling a warehouse process to run at intervals

When scheduling a warehouse process to run at intervals, you must determine the longest time it takes to run all the production steps in the process and schedule the intervals accordingly. If a process exceeds the scheduled time interval, all subsequently scheduled occurrences of that process will not run and will not be rescheduled.

Load and Import Columns page does not support DBCS characters in IXF files

If you use the Load wizard or Import notebook to set up a load or import from an IXF input file containing DBCS characters, the Columns page will not correctly display the column names contained in the file.

Security plug-in problem and restriction for DB2 UDB clients (Windows)

When developing security plug-ins that will be deployed in DB2 clients on Windows operating systems, do not unload any auxiliary libraries in the plug-in termination function. This restriction applies to all types of client security plug-ins, including group, user ID and password, Kerberos, and GSS-API plug-ins.

This restriction is caused by DB2 Universal Database's problem unloading security plug-ins in DB2 clients on Windows operating systems.

Two-part user ID not supported (Windows ME)

The two-part user ID for the CONNECT statement and ATTACH command, such as `domainname\username`, is not supported on Windows ME.

Minimum display settings for GUI tools

For the GUI tools such as the Control Center to work properly, you must have a screen resolution of at least 800 by 600 and use a display palette of at least 32 colors.

Improper display of GB18030 characters in the title bar of a window

If you have characters from the GB18030 Chinese character encoding standard in the title bar of a window, they might appear as questions marks or squares.

Health indicator restrictions

The health monitor is unable to execute actions for the *db2.db2_op_status* health indicator if the indicator enters the down state. This state can be caused, for example, when an instance that the indicator is monitoring becomes inactive because of an explicit stop request or an abnormal termination. If you want to have the instance restart automatically after any abnormal termination, you must configure the fault monitor to keep the instance highly available.

Known problems and workarounds

The db2nkill tool is not shipped in DB2 UDB Express Edition

Problem:

The db2nkill tool is not shipped in DB2 Universal Database (UDB) Express Edition Version 8. This may cause various problems, including failure while using the "db2gcf -k" command.

For example, Issuing the "db2gcf -k -i psustr -p 0 -L" command causes the following response:

```
Instance  : psustr
DB2 Kill : Failure
Partition 0 : Failure
```

The corresponding entry in the db2diag.log file is as follows:

```
2003-07-06-22.11.40.241991 pid:26366 tid:1 level:2
Common - Generic Control Facility - gcf_kill() probe:220
impact 0x900002C1 DB2 kill service failed
data #1 (36 bytes)
 'psustr', 0 : could not be killed.
```

Workaround:

The same scenario works without any failures on DB2 UDB Enterprise Server Edition. Use the db2nkill tool that ships with DB2 UDB Enterprise Server Edition Version 8.

The "db2gcf –k" command fails on DB2 UDB Express Edition

Problem:

The db2gcf command starts, stops, or monitors a DB2 Universal Database (UDB) instance, usually from an automated script, such as in an HA (high availability) cluster.

The db2nkill tool is not shipped with DB2 UDB Express Version 8. This will result in failure when using db2gcf with the –k command parameter.

Workaround:

The "db2gcf –k" command works without any failures on Enterprise Server Edition (ESE) as db2nkill is included with DB2 UDB ESE Version 8.

Hotkeys not working in Microsoft Visual Studio .NET Framework 1.1

If your hotkeys are not working in Microsoft Visual Studio .NET Framework 1.1, you can download a hotfix from the Microsoft Web site. You can find the hotfix in the Microsoft Knowledge Base, article Q836745.

© Copyright IBM Corp. 2002 - 2004

The simplified Chinese locale (Red Hat Linux)

Red Hat Version 8 and later (including Red Hat Enterprise Linux [RHEL] versions 2.1 and 3) have changed the default code set for simplified Chinese from GBK (code page 1386) to GB18030 (code page 5488 or 1392).

Since DB2 Universal Database (UDB) for Linux supports the GBK code set natively and the GB18030 code set via Unicode, DB2 UDB will default its code set to ISO 8859-1 (code page 819), and in some operations will also default its territory to the United States (US).

To work around this limitation, you have two options:

- You can override the Red Hat default code set from GB18030 to GBK and the territory from US to China (whose territory ID is CN and territory code is 86).
- You can use a different simplified Chinese locale.

If you choose to use the first option, issue the following commands:

```
db2set DB2CODEPAGE=1386
db2set DB2TERRITORY=86
db2 terminate
db2stop
db2start
```

If you choose to use the second option, issue any one of the following commands:

```
export LANG=zh_CN.gbk
export LANG=zh_CN
export LANG=zh_CN.utf8
```

where the code set associated with zh_CN is eucCN or code page 1383, and with zh_CN.utf8 is code page 1208.

SQLFLAG (STD) precompiler option error

If the SQLFLAG(STD) precompile option is enabled, it will cause the following error: `Abend C6 occurred while running Precompile program DSNHPC`

Remove the SQLFLAG (STD) precompile option when using the Development Center to create SQL stored procedures to run on DB2 Universal Database for z/OS, Version 8.

DB2 Connect Custom Advisor

Despite being documented in the *DB2 Connect User's Guide*, the DB2 Connect Custom Advisor is no longer supported in Version 8.2.

Displaying Indic characters in the DB2 GUI tools

If you have problems displaying Indic characters when using the DB2 GUI tools, you might not have the required fonts installed on your system.

DB2 Universal Database (UDB) has packaged the following IBM TrueType and OpenType proportional Indic language fonts for your use. You can find these fonts in the `font` directory on any of the following CDs:

- *IBM Developer Kit, Java Technology Edition (64-bit) for AIX 5L*
- *DB2 Embedded Application Server and applications (XML registry, Web Administration tools and Java distributed debugger)* for your operating system

These fonts are to be used only in conjunction with DB2 UDB. You cannot engage in the general or unrestricted sale or distribution of these fonts:

Table 8. Indic fonts packaged with DB2 UDB

Typeface	Weight	Font File Name
Devanagari MT for IBM	Medium	devamt.ttf
Devanagari MT for IBM	Bold	devamtb.ttf
Tamil	Medium	TamilMT.ttf
Tamil	Bold	TamilMTB.ttf
Telugu	Medium	TeluguMT.ttf
Telugu	Bold	TeleguMTB.ttf

Detailed instructions on how to install the fonts and modify the `font.properties` file can be found in the Internationalization section of the IBM Development Kit for Java documentation.

In addition, the following Microsoft products come with Indic fonts that can be used with DB2 GUI tools:

- Microsoft Windows 2000 operating system
- Microsoft Windows XP operating system
- Microsoft Publisher
- Microsoft Office

Enclose DB2 Information Center search terms within quotation marks if they contain numbers

To get accurate search results in the DB2 Information Center you must enclose search terms that include numbers within quotation marks.

For example, if you search for the following term you will receive no results:

```
1.4.1
```

However, if you enclose the term within quotation marks, you receive the appropriate results:

```
"1.4.1"
```

A search for the following term will return extra topics:

```
DB20000I
```

But a search on the following term works properly:

```
"DB20000I"
```

Secure environments (Windows)

You might experience file permission problems if you are using DB2 Universal Database (UDB) on Windows and are not an administrator on the Windows system. If you receive an SQL1035N, SQL1652N, or SQL5005C error message, possible causes and workarounds are shown in the following information:

User does not have sufficient authority on the sqllib directory:

Problem

User receives an SQL1035N or SQL1652N error when trying to open the

DB2 CLP or command window. The DB2 UDB code (core files) are installed into a directory structure where write privileges are limited, but some DB2 UDB tools need to write and create files in the DB2INSTPROF directory.

Workaround

Create a new directory where you can grant users, at minimum, the MODIFY permission and use either **db2set -g db2tempdir** to point to the new directory, or set the db2tempdir variable in the Windows system environment.

User does not have sufficient authority to write to the sqllib\<instance_dir> directory even though user belongs to SYSADM_GROUP:

Problem

User receives an SQL5005C system error when trying to update the database manager configuration file (update dbm cfg). The user does not have the required NTFS permissions to write to the sqllib*instance_dir* directory even though you have added this user to the SYSADM_GROUP.

First workaround

Grant the users, at minimum, the MODIFY permission on the *instance_dir* directory at the file system level.

Second workaround

Create a new directory where you can grant the user, at minimum, the MODIFY permission. Use the **db2set db2instprof** command to point to the new directory. You will need either to recreate the instance so that the information is stored under the new instance directory specified by db2instprof, or you will need to move the old instance directory to the new directory.

XML Extender sample programs renamed

Some XML Extender sample programs might have the same name as other installed programs. Accidentally invoking another program with the same name as an XML Extender sample program can damage your XML files. The following list shows the old XML Extender sample program names as well as new replacement program names that are less likely to cause conflicts. Make sure you use the new sample program names instead of the old ones to prevent damaging your XML files.

Table 9. Replacement sample programs for XML Extender (Windows)

Old Program (Do not use)	New Program (Use)
insertx.exe	dxxisrt.exe
retrieve.exe	dxxretr.exe
retrieve2.exe	dxxretr2.exe
retrievec.exe	dxxretrc.exe
shred.exe	dxxshrd.exe
tests2x.exe	dxxgenx.exe
tests2xb.exe	dxxgenxb.exe
tests2xc.exe	dxxgenxc.exe

Table 10. Replacement sample programs for XML Extender (Linux)

Old Program (Do not use)	New Program (Use)
insertx	dxxisrt
retrieve	dxxretr
retrieve2	dxxretr2
retrievec	dxxretrc
shred	dxxshrd
tests2x	dxxgenx
tests2xb	dxxgenxb
tests2xc	dxxgenxc

Using the new sample programs with the sample sqx files

The source code (.sqx files) for the executables listed previously are located in the `samples\db2xml\c` directory of your installation. The source files are still labeled with their old names. If you make changes to the source code, copy your newly compiled executables (with the old names) to the `sqllib\bin` directory.

On Windows platforms, you must make an additional copy, rename it with its new name above, and copy it to the bin directory. Both copies replace the existing files in the bin directory. For example, after compiling your new version of shred.exe, you need to make two copies and replace the files in the bin directory: one labeled shred.exe and the other renamed dxxshrd.exe.

On Linux platforms, you need only to replace the file with the old name with your newly compiled version. If you create new executable files from these samples, you must copy the new files from the `\SQLLIB\samples\db2xml\c\` directory into the `\SQLLIB\bin\` directory, and then make an additional copy, renaming them according to the previous table.

Decomposing documents in XML Extender that contain non unique attribute and element names

You can now decompose documents that contain non-unique attributes or non-unique element names that map to different columns (of the same or different tables) without receiving the DXXQ045E error. The following is an example of an XML document with non-unique attributes and non-unique element names:

```
<Order ID="0001-6789">
       <!-- Note: attribute name ID is non-unique –>
       <Customer ID="1111">
                    <Name>John Smith</Name>
       </Customer>
       <!-- Note: element name Name is non_unique –>
       <Salesperson ID="1234">
              <Name>Jane Doe</Name>
       </Salesperson>
       <OrderDetail>
              <ItemNo>xxxx-xxxx</ItemNo>
              <Quantity>2</Quantity>
              <UnitPrice>12.50</UnitPrice>
       </OrderDetail>
       <OrderDetail>
              <ItemNo>yyyy-yyyy</ItemNo>
```

```
            <Quantity>4</Quantity>
            <UnitPrice>24.99</UnitPrice>
      </OrderDetail>
</Order>
```

The accompanying DAD, which maps the duplicate elements and attributes to different columns, looks like this:

```
<element_node name="Order">
  <RDB_node>
    <table name="order_tab" key="order_id"/>
    <table name="detail_tab"/>
    <condition>
      order_tab.order_id=detail_tab.order_id
    </condition>
  </RDB_node>

  <!--attribute ID duplicated below, but mapped to a different col-->
  <attribute_node name="ID">
    <RDB_node>
      <table name="order_tab" />
      <column name="order_id" type="char(9)"/>
    </RDB_node>
  </attribute_node>

  <element_node name="Customer">
    <!--attribute ID duplicated above, but mapped to a different col-->
    <attribute_node name="ID">
      <RDB_node>
        <table name="order_tab" />
        <column name="cust_id" type="integer"/>
      </RDB_node>
    </attribute_node>

    <!--element name duplicated below, but mapped to a different col-->
    <element_node name="Name">
      <text_node>
        <RDB_node>
          <table name="order_tab" />
          <column name="cust_name" type="char(20)" />
        </RDB_node>
      </text_node>
    </element_node>
  </element_node>

  <element_node name="Salesperson">
    <!--attribute ID duplicated above, but mapped to a different col-->
    <attribute_node name="ID">
      <RDB_node>
        <table name="order_tab" />
        <column name="salesp_id" type="integer"/>
      </RDB_node>
    </attribute_node>

    <!--element name duplicated above, but mapped to a different col-->
    <element_node name="Name">
      <text_node>
        <RDB_node>
          <table name="order_tab" />
          <column name="salesp_name" type="char(20)" />
        </RDB_node>
      </text_node>
    </element_node>
  </element_node>

  <element_node name="OrderDetail" multi_occurrence="YES">
    <element_node name="ItemNo">
```

```
      <text_node>
        <RDB_node>
          <table name="detail_tab" />
          <column name="itemno" type="char(9)"/>
        </RDB_node>
      </text_node>
    </element_node>
    <element_node name="Quantity">
      <text_node>
        <RDB_node>
          <table name="detail_tab" />
          <column name="quantity" type="integer"/>
        </RDB_node>
      </text_node>
    </element_node>
    <element_node name="UnitPrice">
      <text_node>
        <RDB_node>detail_tab" />
          <table name="detail_tab" />
          <column name="unit_price" type="decimal(7,2)"/>
        </RDB_node>
      </text_node>
    </element_node>
  </element_node>
</element_node>
```

The contents of the tables would look like the following sample after the preceding document is decomposed:

```
ORDER _TAB:

ORDER_ID       CUST_ID       CUST_NAME       SALESP_ID       SALESP_NAME
0001-6789      1111          John Smith      1234            Jane Doe

DETAIL_TAB:

ORDER_ID          ITEMNO         QUANTITY          UNIT_PRICE
0001-6789         xxxx-xxxx      2                 12.50
0001-6789         yyyy-yyyy      4                 24.99
```

Note: To map multiple elements and attributes to the same column of the same table, define an alias for the table and use the alias in the DAD <table> element of one of the mappings.

Documentation updates

Administration: Implementation

Authentication considerations for remote clients

The authentication type DATA_ENCRYPT_CMP is designed to allow clients from a previous release that do not support data encryption to connect to a server using SERVER_ENCRYPT authentication instead of DATA_ENCRYPT. This authentication does not work when the following three statements are true:

- The client level is Version 7.2.
- The gateway level is Version 8 FixPak 7 or later.
- The server is Version 8 FixPak 7 or later.

In this case, the client cannot connect to the server. To allow the connection, you must either upgrade your client to Version 8, or have your gateway level at Version 8 FixPak 6 or earlier.

Direct I/O (DIO) and concurrent I/O (CIO) support

Direct I/O (DIO) improves memory performance because it bypasses caching at the file system level. This process reduces CPU overhead and makes more memory available to the database instance.

Concurrent I/O (CIO) includes the advantages of DIO and also relieves the serialization of write accesses.

DB2 Universal Database (UDB) supports DIO and CIO on AIX®; and DIO on HP-UX, Solaris Operating Environment, Linux, and Windows.

The keywords NO FILE SYSTEM CACHING and FILE SYSTEM CACHING are part of the CREATE and ALTER TABLESPACE SQL statements to allow you to specify whether DIO or CIO is to be used with each table space. When NO FILE SYSTEM CACHING is in effect, DB2 UDB attempts to use CIO wherever possible. In cases, where CIO is not supported (for example, if JFS is used), DIO is used instead.

For more information, refer to the article “Improve database performance on file system containers in IBM DB2 UDB Stinger using Concurrent I/O on AIX” located at the following URL:

http://www.ibm.com/developerworks/db2/library/techarticle/dm-0408lee/

Distributor technology and automatic client rerouting

The following information is part of the *Administration Guide: Implementation* Appendix B “Using automatic client rerouting”:

The DB2 Universal Database for Linux and Windows automatic client reroute feature allows client applications to recover from a loss of communication with the server by automatically reestablishing the database connection from the client to the server, so that the application can continue to work with minimal interruption.

© Copyright IBM Corp. 2002 - 2004

When a client to server connection fails, the client's requests for reconnection are distributed to a defined set of systems by a distributor or dispatcher, such as WebSphere® EdgeServer

You may be using Distributor Technology in an environment similar to the following:

Client —> Distributor Technology —> (DB2 Connect Server 1 or DB2 Connect Server 2) —> DB2 z/OS

where:

- The Distributor Technology component has a TCP/IP host name of DThostname
- The DB2 Connect Server 1 has a TCP/IP host name of GWYhostname1
- The DB2 Connect Server 2 has a TCP/IP host name of GWYhostname2
- The DB2 z/OS server has a TCP/IP host name of zOShostname

The client is catalogued using DThostname in order to utilize the distributor technology to access either of the DB2 Connect Servers. The intervening distributor technology makes the decision to use GWYhostname1 or GWYhostname2. Once the decision is made, the client has a direct socket connection to one of these two DB2 Connect gateways. Once the socket connectivity is established to the chosen DB2 Connect server, you have a typical client to DB2 Connect server to DB2 z/OS connectivity.

For example, assume the distributor chooses GWYhostname2. This produces the following environment:

Client —> DB2 Connect Server 2 —> DB2 z/OS

The distributor does not retry any of the connections if there is any communication failure. If you want to enable the Automatic Client Reroute feature for a database in such an environment, the alternate server for the associated database or databases in the DB2 Connect Server (DB2 Connect Server 1 or DB2 Connect Server 2) should be set up to be the distributor (DThostname). Then, if DB2 Connect Server 1 locks up for any reason, Automatic Client Reroute is triggered and client connection is retried with the distributor as both primary and alternate server. This option allows you to combine and maintain the distributor capabilities with the DB2 Automatic Client Reroute feature. Setting the alternate server to a host other than the distributor host name will still provide the clients with the Automatic Client Reroute feature. However, the clients will establish direct connections to the defined alternate server and bypass the distributor technology, which eliminates the distributor and the value that it brings.

Automatic Client Reroute will intercept the following sqlcodes:

- sqlcode -20157
- sqlcode -1768 (reason code = 7)

Local system account support (Windows)

Applications running under the context of the local system account (LSA) are supported on all Windows platforms, except Windows ME.

Two-part user ID support

The CONNECT statement and ATTACH command support two-part user IDs. The qualifier of the SAM-compatible user ID is the NetBIOS style name which has a maximum length of 15 characters. This feature is not supported on Windows ME.

Additional information for Kerberos support

Linux prerequisites:

The prerequisites for Linux Kerberos support are inaccurately reported in the documentation. The provided DB2 Kerberos security plug-in is supported with Red Hat Enterprise Linux Advanced Server 3 with the IBM Network Authentication Service (NAS) 1.4 client.

zSeries and iSeries compatibility:

For connections to zSeries and iSeries, the database must be cataloged with the AUTHENTICATION KERBEROS parameter and the TARGET PRINCIPAL parameter name must be explicitly specified.

Neither zSeries nor iSeries support mutual authentication.

Windows issues:

- Due to the manner in which Windows detects and reports some errors, the following conditions result in an unexpected client security plug-in error (SQL30082N, rc=36):
  - Expired account
  - Invalid password
  - Expired password
  - Password change forced by administrator
  - Disabled account

  Furthermore, in all cases, the DB2 administration log or db2diag.log will indicate ″Logon failed″ or ″Logon denied.″
- If a domain account name is also defined locally, connections explicitly specifying the domain name and password will fail with the following error:

  ```
  The Local Security Authority cannot be contacted
  ```

  The error is a result of Windows locating the local user first. The solution is to fully qualify the user in the connection string. For example:

  ```
  name@DOMAIN.IBM.COM
  ```
- Windows accounts cannot include the @ character in their name because the character is assumed to be the domain separator by the DB2 Kerberos plug-in.
- When interoperating with a non-Windows platform, ensure that all Windows domain server accounts and all Windows client accounts are configured to use DES encryption. If the account used to start the DB2 service is not configured to use DES encryption, the DB2 server will fail to accept Kerberos contexts. In particular, DB2 Universal Database will fail with an unexpected server plug-in error, and will log that the AcceptSecurityContext API returned SEC_I_CONTINUTE_NEEDED (0x00090312L).

  To determine if Windows accounts are configured to use DES encryption, look under **Account properties** in the **Active Directory**. A restart might be required if the account properties are changed.

- If the client and server are both on Windows, then the DB2 service can be started under the local system account. However, if the client and server are in different domains, the connection might fail with an invalid target principal name error. The workaround is to explicitly catalog the target principal name on the client using the fully qualified server host name and the fully qualified domain name, in the following format:

  host/<*server hostname*>@<*server domain name*>

  For example:

  host/myhost.domain.ibm.com@DOMAIN.IBM.COM

  Otherwise, you must start the DB2 service under a valid domain account.

Administration: Performance

New system environment variables (Linux)

The DB2_MAPPED_BASE and DB2DBMSADDR system environment variables have been added at FixPak 8.

Use of these registry variables is only recommended for advanced users.

DB2_MAPPED_BASE:

Variable name
DB2_MAPPED_BASE

Values
0 OR (hex) virtual address in the 31-bit and 32-bit address range OR NULL (not set)

Operating systems
Linux on x86 and Linux on zSeries (31-bit)

Description
The DB2_MAPPED_BASE registry variable can be used to increase the amount of contiguous virtual address space available to a DB2 Universal Database (UDB) process by relocating the attachment address of the shared libraries for the specific process. The contiguous virtual address space is important to maximize the amount of database shared memory available to DB2 UDB. This variable is only effective on distributions that include the mapped_base file in the process identification directory in the proc file system.

DB2 UDB will attempt to relocate the shared libraries to the virtual address 0x20000000 if this variable is not set.

The registry variable can also be set to any virtual address (in hex) in the range of the 31 and 32-bit address space if the new address places the shared libraries lower in the address space.

Note: An incorrect address can cause severe issues with DB2 UDB, ranging from an inability to start DB2 UDB to an inability to connect to the database. An incorrect address is one that collides with an area in memory that is already in use or is predestined to be used for something else. To address this problem, reset the DB2_MAPPED_BASE variable to NULL by using the following command:

```
db2set DB2_MAPPED_BASE=
```

The following message may appear multiple times in the db2diag.log file because this change is required once per logical node:

```
ADM0506I  DB2 has automatically updated the "mapped_base"
kernel parameter from "0x40000000(hex) 1073741824(dec)" to
the recommended value "0x20000000(hex) 536870912(dec)".
```

This message will only appear if setting of the registry variable successful, and it will include the address that the shared libraries are relocated to.

DB2DBMSADDR:

Variable name
DB2DBMSADDR

Values
Virtual addresses in the range 0x09000000 to 0xB0000000 in increments of 0x10000

Operating systems
Linux on x86 and Linux on zSeries (31-bit)

Description
Specifies the default database shared memory address in hexadecimal format.

Note: An incorrect address can cause severe issues with DB2 UDB, ranging from an inability to start DB2 UDB, to an inability to connect to the database. An example of an incorrect address is one that collides with an area in memory that is already in use or predestined to be used for something else. To address this problem, reset the DB2DBMSADDR variable to NULL by using the following command:

```
db2set DB2DBMSADDR=
```

This variable can be set in conjunction with DB2_MAPPED_BASE or alone to fine tune the address space layout of DB2 UDB processes. This variable changes the location of the instance shared memory from it's current location at virtual address 0x10000000 to the new value given.

New communication registry variable

The DB2TCP_CLIENT_RCVTIMEOUT registry variable has been added in Version 8.2.

Table 11. Communications variables

Variable name	Operating systems	Values
Description		
DB2TCP_CLIENT_RCVTIMEOUT	All	Default=0 (not set) Values: 0 to 32767 seconds

Table 11. Communications variables (continued)

Variable name	Operating systems	Values
Description		
Specifies the number of seconds a client waits for data on a TCP/IP receive. There is no timeout if the registry variable is not set or is set to 0. If the TCP/IP receive returns with data before the timeout value has expired, the application proceeds as usual. If the timeout value expires before data is returned, the connection closes. **Note:** This registry variable is applicable to the DB2 Client and the client side of the DB2 Gateway only. It is not applicable to the DB2 Server.		

SQL compiler variables

The following update applies to the topic "SQL compiler variables" in Appendix A "DB2 registry and environment variables" of the *Administration Guide: Performance*:

When either or both of the DB2 compiler variables DB2_MINIMIZE_LISTPREFETCH and DB2_INLIST_TO_NLJN, are set to ON, they remain active even if REOPT(ONCE) is specified.

Configuration parameter updates

Following are the updates to the configuration parameter documentation:

authentication – Authentication type

The Authentication type (authentication) database manager configuration parameter also accepts the following values:

- DATA_ENCRYPT

  The server accepts encrypted SERVER authentication schemes and the encryption of user data. The authentication works exactly the same way as SERVER_ENCRYPT.

  The following user data are encrypted when using this authentication type:

  – SQL statements.
  – SQL program variable data.
  – Output data from the server processing an SQL statement and including a description of the data.
  – Some or all of the answer set data resulting from a query.
  – Large object (LOB) data streaming.
  – SQLDA descriptors.

- DATA_ENCRYPT_CMP

  The server accepts encrypted SERVER authentication schemes and the encryption of user data. In addition, this authentication type allows compatibility with earlier products that do not support DATA_ENCRYPT authentication type. These products are permitted to connect with the SERVER_ENCRYPT authentication type and without encrypting user data. Products supporting the new authentication type must use it. This authentication type is only valid in the server's database manager configuration file and is not valid when used on the CATALOG DATABASE command.

util_impact_lim – Instance impact policy

Starting with DB2 Universal Database Version 8.2, the default value of the **Instance impact policy** (*util_impact_lim*) database manager configuration parameter changes from 100 to 10.

sysadm_group, sysmaint_group, sysctrl_group, sysmon_group

The following database manager configuration parameters can all accept group names of 30 bytes (or less) on all platforms:

- System administration authority group name (*sysadm_group*)
- System maintenance authority group name (*sysmaint_group*)
- System control group authority name (*sysctrl_group*)
- System monitor authority group name (*sysmon_group*)

The table in the topic "Database manager configuration parameter summary" contains incorrect data types for these database manager configuration parameters. The correct value in all cases is char(30).

estore_seg_sz – Extended storage memory segment size

The maximum size for the **Extended storage memory segment size database** (*estore_seg_size*) configuration parameter on Windows based platforms is 16 777 216.

hadr_timeout – HADR timeout value

The correct upper limit of the **HADR timeout value** (*hadr_timeout*) database configuration parameter is 4 294 967 295.

locklist – Maximum storage for lock list

The documentation for the **Maximum storage for locklist** (*locklist*) database configuration parameter states that the maximum value for Windows 64-bit and 32-bit servers that service only local clients is 60 000. This value is incorrect, and should be 524 288.

num_db_backups – Number of database backups

The range of values for the **Number of database backups** (*num_db_backups* database configuration parameter is incorrect. The correct range is 0 – 32 767.

SQLDBCONF database configuration parameter file

After migrating to DB2 Universal Database (UDB) Version 8.2 from Version 8.1, DB2 UDB uses a new 16 KB database configuration parameter file named SQLDBCONF. (In Version 8.1, the database configuration parameter file was only 4 KB and named SQLDBCON).

Change to the DB2_HASH_JOIN default value

As of Version 8.1 the registry variable DB2_HASH_JOIN is set to ON by default.

The hash-join variable should be used, but it needs to be tuned to get the best performance.

Hash-join performance is best if you can avoid hash loops and overflow to disk. To tune hash-join performance, estimate the maximum amount of memory available for the **sheapthres** parameter , then tune the **sortheap** parameter. Increase its value until you avoid as many hash loops and disk overflows as possible, but do not reach the limit specified by the **sheapthres** parameter.

For more information, see the "Join methods" topic in the *Administration Guide: Performance* manual.

DB2NTNOCACHE registry variable is deprecated

Functionality previously achieved through DB2NTNOCACHE can be achieved at the table space level by specifying the NO FILE SYSTEM CACHING clause on the CREATE TABLESPACE or the ALTER TABLESPACE statement. Refer to *SQL Reference* for details on usage. The DB2NTNOCACHE registry variable will be removed in a future release.

Explain tables and organization of explain information

Explain tables might be common to more than one user. However, the explain tables can be defined for one user, and aliases can be defined for each additional user using the same name to point to the defined tables. Alternatively, the explain tables can be defined under the SYSTOOLS schema. The Explain facility will default to the SYSTOOLS schema if no other explain tables or aliases are found under the user's session ID for dynamic SQL, or the statement authorization ID for static SQL. Each user sharing the common explain tables must have insert permission on those tables. Read permission for the common explain tables should also be limited, typically to users who analyze the explain information.

Guidelines for capturing explain information

Explain data is captured if you request it when an SQL statement is compiled. Consider how you expect to use the captured information when you request explain data.

Capturing information in the explain tables:

- Dynamic SQL statements:

  Explain table information is captured in any of the following cases:

  – The CURRENT EXPLAIN MODE special register is set to:
    - YES: The SQL compiler captures explain data and executes the SQL statement.
    - EXPLAIN: The SQL compiler captures explain data but does not execute the SQL statement.
    - RECOMMEND INDEXES: The SQL compiler captures explain data and the recommended indexes are placed in the ADVISE_INDEX table, but the SQL statement is not executed.
    - EVALUATE INDEXES: The SQL compiler uses indexes placed by the user in the ADVISE_INDEX table for evaluation. In EVALUATE INDEXES mode, all dynamic statements are explained as if these virtual indexes were available. The SQL compiler then chooses to use the virtual indexes if they improve the performance of the statements. Otherwise, the indexes are ignored. To find out if proposed indexes are useful, review the EXPLAIN results.
    - REOPT: The SQL compiler captures Explain data for static or dynamic SQL statements during statement reoptimization at execution time, when actual values for the host variables, special registers, or parameter markers are available.

  – The EXPLAIN ALL option has been specified on the BIND or PREP command. The SQL compiler captures explain data for dynamic SQL at run time, even if the CURRENT EXPLAIN MODE special register is set to NO. The SQL statement also executes and returns the results of the query.

Additional return codes from db2CfgGet API, collate_info parameter

The collating information parameter can only be displayed using the db2CfgGet API. It **cannot** be displayed through the command line processor or the Control Center.

Configuration Type Database

Parameter Type Informational

This parameter provides 260 bytes of database collating information. The first 256 bytes specify the database collating sequence, where byte “n” contains the sort weight of the code point whose underlying decimal representation is “n” in the code page of the database.

The last 4 bytes contain internal information about the type of the collating sequence. The last 4 bytes of collate_info is an integer. The integer is sensitive to the endian order of the platform. The possible values are:

- **0** – The sequence contains non-unique weights
- **1** – The sequence contains all unique weights
- **2** – The sequence is the identity sequence, for which strings are compared byte for byte.
- **3** – The sequence is NLSCHAR, used for sorting characters in a TIS620-1 (code page 874) Thai database.
- **4** – The sequence is IDENTITY_16BIT, which implements the "CESU-8 Compatibility Encoding Scheme for UTF-16: 8-Bit" algorithm as specified in the Unicode Technical Report #26 available at the Unicode Technical Consortium Web site at http://www.unicode.org.
- **X'8001'** – The sequence is UCA400_NO, which implements the UCA (Unicode Collation Algorithm) based on the Unicode Standard version 4.00, with normalization implicitly set to ON.
- **X'8002'** – The sequence is UCA400_LTH, which implements the UCA (Unicode Collation Algorithm) based on the Unicode Standard version 4.00, and sorts all Thai characters as per the Royal Thai Dictionary order.
- **X'8003'** – The sequence is UCA400_LSK. which implements the UCA (Unicode Collation Algorithm) based on the Unicode Standard version 4.00, and sorts all Slovakian characters properly.

If you use this internal type information, you need to consider byte reversal when retrieving information for a database on a different platform.

You can specify the collating sequence at database creation time.

Automatic setting of default prefetch size and update defaults

Starting with DB2 Universal Database (UDB) Version 8.2, you can use AUTOMATIC prefetch size for a table space. DB2 UDB automatically updates the prefetch size when the number of containers changes for the table space.

The syntax of the DB2_PARALLEL_IO registry variable is expanded to recognize containers with different I/O parallelism characteristics. Through the expanded syntax, containers for different table spaces can have different I/O parallelism characteristics. The I/O parallelism characteristic of each table space is used when a prefetch size of AUTOMATIC is specified for the table space. If the DB2_PARALLEL_IO registry variable is enabled but the expanded syntax

identifying specific I/O parallelism characteristics for table spaces is not used, a default level of parallelism is assumed. The default level is RAID 5 (6+1).

The prefetch size information used by the optimizer is refreshed only when an ALTER TABLESPACE statement that changes the prefetch size of a table space or changes the number of containers (using ADD/DROP/BEGIN NEW STRIPE SET/ADD TO NEW STRIPE SET) is issued. If the number of physical disks per container registry settings changes, an **ALTER TABLESPACE <table space name> PREFETCHSIZE AUTOMATIC** statement should be issued to refresh the optimizer information (unless an ALTER TABLESPACE statement that refreshes the optimizer information is already issued).

If a table space is redirected or restored to use a different number of containers, refresh the optimizer information by issuing an **ALTER TABLESPACE <table space name> PREFETCHSIZE AUTOMATIC** statement. If there are multiple stripe sets within a table space, the maximum number of containers among the stripe sets is used to calculated the prefetch size. If the calculated prefetch size exceeds the maximum size (32 767 pages), the largest multiple of the number of containers that is smaller than the maximum is used as the prefetch size.

In a DB2 UDB Enterprise Server Edition environment, if a table space uses an AUTOMATIC prefetch size, the prefetch size might be different on different database partitions. This situation can exist because different database partitions can have different numbers of containers used for calculating the prefetch size. To generate the query access plan, the optimizer uses the prefetch size from the first partition in a database partition group.

Administration: Planning

Supported territory codes and code pages

In Appendix B, "National language support (NLS)" in *Administration Guide: Planning*, the topic called "Supported territory codes and code pages" has tables for each territory. Two tables require updates:

China (PRC), territory identifier: CN

The code page for the Linux GBK row in the "China (PRC), territory identifier: CN" table should be changed from 1383 to 1386.

That is, the row should now read:

```
1386     D-4     GBK     86     zh_CN.GBK     Linux
```

Japan, territory identifier: JP

The table for " Japan, territory identifier: JP" has been revised.

The following locale name should be removed:

```
954      D-1      eucJP   81       japanese      Solaris
```

Here is the revised table:

Table 12. Japan, territory identifier: JP

Code page	Group	Code set	Territory code	Locale	Operating system
932	D-1	IBM-932	81	Ja_JP	AIX
943	D-1	IBM-943	81	Ja_JP	AIX

Table 12. Japan, territory identifier: JP (continued)

Code page	Group	Code set	Territory code	Locale	Operating system
954	D-1	IBM-eucJP	81	ja_JP	AIX
1208	N-1	UTF-8	81	JA_JP	AIX
930	D-1	IBM-930	81	-	Host
939	D-1	IBM-939	81	-	Host
5026	D-1	IBM-5026	81	-	Host
5035	D-1	IBM-5035	81	-	Host
1390	D-1		81	-	Host
1399	D-1		81	-	Host
954	D-1	eucJP	81	ja_JP.eucJP	HP-UX
5039	D-1	SJIS	81	ja_JP.SJIS	HP-UX
954	D-1	EUC-JP	81	ja_JP	Linux
932	D-1	IBM-932	81	-	OS/2®
942	D-1	IBM-942	81	-	OS/2
943	D-1	IBM-943	81	-	OS/2
954	D-1	eucJP	81	ja	SCO
954	D-1	eucJP	81	ja_JP	SCO
954	D-1	eucJP	81	ja_JP.EUC	SCO
954	D-1	eucJP	81	ja_JP.eucJP	SCO
943	D-1	IBM-943	81	ja_JP.PCK	Solaris
954	D-1	eucJP	81	ja	Solaris
1208	N-1	UTF-8	81	ja_JP.UTF-8	Solaris
943	D-1	IBM-943	81	-	Windows
1394	D-1		81	-	

XA function supported by DB2 Universal Database

DB2 Universal Database (UDB) supports the XA91 specification defined in *X/Open CAE Specification Distributed Transaction Processing: The XA Specification*, with the following exceptions:

- Asynchronous services

  The XA specification allows the interface to use asynchronous services, so that the result of a request can be checked at a later time. The database manager requires that the requests be invoked in synchronous mode.

- Registration

  The XA interface allows two ways to register an RM: static registration and dynamic registration. DB2 UDB supports both dynamic and static registration. DB2 UDB provides two switches:

  – *db2xa_switch* for dynamic registration

  – *db2xa_switch_static* for static registration

- Association migration

  DB2 UDB does not support transaction migration between threads of control.

XA switch usage and location

As required by the XA interface, the database manager provides a *db2xa_switch* and a *db2xa_switch_static* external C variable of type xa_switch_t to return the XA switch structure to the TM. Other than the addresses of various XA functions, the following fields are returned:

Field **Value**

name The product name of the database manager. For example, DB2 UDB for AIX.

flags For *db2xa_switch* TMREGISTER | TMNOMIGRATE is set

Explicitly states that DB2 UDB uses dynamic registration, and that the TM should not use association migration. Implicitly states that asynchronous operation is not supported.

For *db2xa_switch_static* TMNOMIGRATE is set

Explicitly states that DB2 UDB uses dynamic registration, and that the TM should not use association migration. Implicitly states that asynchronous operation is not supported.

version Must be zero.

Using the DB2 Universal Database XA switch

The XA architecture requires that a Resource Manager (RM) provide a *switch* that gives the XA Transaction Manager (TM) access to the RM's **xa_** routines. An RM switch uses a structure called xa_switch_t. The switch contains the RM's name, non-NULL pointers to the RM's XA entry points, a flag, and a version number.

Linux–based systems: The switch for DB2 Universal Database (UDB) can be obtained through either of the following two ways:

- Through one additional level of indirection. In a C program, this can be accomplished by defining the macro:

  ```
  #define db2xa_switch (*db2xa_switch)
  #define db2xa_switch_static (*db2xa_switch)
  ```

  prior to using *db2xa_switch* or *db2xa_switch_static*.
- By calling **db2xacic** or **db2xacicst**

  DB2 UDB provides these APIs, which return the address of the *db2xa_switch* or the *db2xa_switch_static* structure. This function is prototyped as:

  ```
  struct xa_switch_t * SQL_API_FN  db2xacic( )
  struct xa_switch_t * SQL_API_FN  db2xacicst( )
  ```

With either method, you must link your application with `libdb2`.

Windows NT: The pointer to the *xa_switch* structure, *db2xa_switch*, or *db2xa_switch_static* is exported as DLL data. This implies that a Windows NT application using this structure must reference it in one of three ways:

- Through one additional level of indirection. In a C program, this can be accomplished by defining the macro:

  ```
  #define db2xa_switch (*db2xa_switch)
  #define db2xa_switch_static (*db2xa_switch)
  ```

  prior to using *db2xa_switch* or *db2xa_switch_static*.
- If using the Microsoft Visual C++ compiler, *db2xa_switch* or *db2xa_switch_static* can be defined as:

  ```
  extern __declspec(dllimport) struct xa_switch_t db2xa_switch
  extern __declspec(dllimport) struct xa_switch_t db2xa_switch_static
  ```
- By calling **db2xacic** or **db2xacicst**

  DB2 UDB provides this API, which returns the address of the *db2xa_switch* or *db2xa_switch_static* structure. This function is prototyped as:

  ```
  struct xa_switch_t * SQL_API_FN  db2xacic( )
  struct xa_switch_t * SQL_API_FN  db2xacicst( )
  ```

With any of these methods, you must link your application with `db2api.lib`.

Example C Code: The following code illustrates the different ways in which the *db2xa_switch* or *db2xa_switch_static* can be accessed via a C program on any DB2 UDB platform. Be sure to link your application with the appropriate library.

```
   #include <stdio.h>
   #include <xa.h>

   struct xa_switch_t * SQL_API_FN  db2xacic( );

   #ifdef DECLSPEC_DEFN
   extern __declspec(dllimport) struct xa_switch_t db2xa_switch;
   #else
   #define db2xa_switch (*db2xa_switch)
   extern struct xa_switch_t db2xa_switch;
   #endif
main( )
   {
      struct xa_switch_t *foo;
      printf ( "%s \n", db2xa_switch.name );
      foo = db2xacic();
      printf ( "%s \n", foo->name );
      return ;
   }
```

Updated TPM and tp_mon_name values for xa_open string formats

The Internal settings column in the following table has been updated to reflect settings for the thread of control (TOC).

TOC is the entity to which all DB2 UDB XA connections are bound:

- When set to a value of T, all DB2 UDB XA connections formed under a particular OS thread are unique to that thread. Multiple threads cannot share DB2 UDB connections. Each OS thread has to form its own set of DB2 UDB XA connections.
- When set to a value of P, all DB2 UDB XA connections are unique to the OS Process and all XA connections can be shared between OS threads.

Table 13. Valid values for TPM and tp_mon_name

TPM value	TP monitor product	Internal settings
CICS®	IBM TxSeries CICS	AXLIB=libEncServer (for Windows) =/usr/lpp/encina/lib/libEncServer (for Linux– systems) HOLD_CURSOR=T CHAIN_END=T SUSPEND_CURSOR=F TOC=T
ENCINA	IBM TxSeries Encina® monitor	AXLIB=libEncServer (for Windows) =/usr/lpp/encina/lib/libEncServer (for Linux– systems) HOLD_CURSOR=F CHAIN_END=T SUSPEND_CURSOR=F TOC=T

Table 13. Valid values for TPM and tp_mon_name (continued)

TPM value	TP monitor product	Internal settings
MQ	IBM MQSeries®	AXLIB=mqmax (for Windows) =/usr/mqm/lib/libmqmax_r.a (for AIX threaded applications) =/usr/mqm/lib/libmqmax.a (for AIX non-threaded applications) =/opt/mqm/lib/libmqmax.so (for Solaris) =/opt/mqm/lib/libmqmax_r.sl (for HP threaded applications) =/opt/mqm/lib/libmqmax.sl (for HP non-threaded applications) =/opt/mqm/lib/libmqmax_r.so (for Linux threaded applications) =/opt/mqm/lib/libmqmax.so (for Linux non-threaded applications) HOLD_CURSOR=F CHAIN_END=F SUSPEND_CURSOR=F TOC=P
CB	IBM Component Broker	AXLIB=somtrx1i (for Windows) =libsomtrx1 (for Linux– systems) HOLD_CURSOR=F CHAIN_END=T SUSPEND_CURSOR=F TOC=T
SF	IBM San Francisco	AXLIB=ibmsfDB2 HOLD_CURSOR=F CHAIN_END=T SUSPEND_CURSOR=F TOC=T
TUXEDO	BEA Tuxedo	AXLIB=libtux HOLD_CURSOR=F CHAIN_END=F SUSPEND_CURSOR=F TOC=T
MTS	Microsoft Transaction Server	It is not necessary to configure DB2 UDB for MTS. MTS is automatically detected by DB2 UDB's ODBC driver.
JTA	Java Transaction API	It is not necessary to configure DB2 UDB for Enterprise Java Servers (EJS) such as IBM WebSphere. DB2 UDB's JDBC driver automatically detects this environment. Therefore this TPM value is ignored.

Activating conversion tables for code pages 923 and 924

The following table contains a list of all the code page conversion table files that are associated with code pages 923 and 924. Each file is of the form XXXXYYYY.cnv or ibmZZZZZ.ucs, where XXXXX is the source code page number and YYYY is the target code page number. The file ibmZZZZZ.ucs supports conversion between code page ZZZZZ and Unicode.

Procedure:

To activate a particular code page conversion table, rename or copy that conversion table file to its new name as shown in the second column.

For example, to support the euro symbol when connecting a 8859-1/15 (Latin 1/9) client to a Windows 1252 database, you need to rename or copy the following code page conversion table files in the sqllib/conv/ directory:

- 09231252.cnv to 08191252.cnv
- 12520923.cnv to 12520819.cnv
- ibm00923.ucs to ibm00819.ucs

Table 14. Conversion table files for code pages 923 and 924

923 and 924 conversion table files in the sqllib/conv/ directory	**New name**
04370923.cnv	04370819.cnv
08500923.cnv	08500819.cnv
08600923.cnv	08600819.cnv
08630923.cnv	08630819.cnv
09230437.cnv	08190437.cnv
09230850.cnv	08190850.cnv
09230860.cnv	08190860.cnv
09231043.cnv	08191043.cnv
09231051.cnv	08191051.cnv
09231114.cnv	08191114.cnv
09231252.cnv	08191252.cnv
09231275.cnv	08191275.cnv
09241252.cnv	10471252.cnv
10430923.cnv	10430819.cnv
10510923.cnv	10510819.cnv
11140923.cnv	11140819.cnv
12520923.cnv	12520819.cnv
12750923.cnv	12750819.cnv
ibm00923.ucs	ibm00819.ucs

Conversion table files for euro-enabled code pages

The following tables list the conversion tables that have been enhanced to support the euro currency symbol. If you want to disable euro symbol support, download the conversion table file indicated in the column titled ″Conversion table files″.

Arabic:

Database server CCSIDs/CPGIDs	**Database client CCSIDs/CPGIDs**	**Conversion table files**
864, 17248	1046, 9238	08641046.cnv, 10460864.cnv, IBM00864.ucs
864, 17248	1256, 5352	08641256.cnv, 12560864.cnv, IBM00864.ucs
864, 17248	1200, 1208, 13488, 17584	IBM00864.ucs

Database server CCSIDs/CPGIDs	Database client CCSIDs/CPGIDs	Conversion table files
1046, 9238	864, 17248	10460864.cnv, 08641046.cnv, IBM01046.ucs
1046, 9238	1089	10461089.cnv, 10891046.cnv, IBM01046.ucs
1046, 9238	1256, 5352	10461256.cnv, 12561046.cnv, IBM01046.ucs
1046, 9238	1200, 1208, 13488, 17584	IBM01046.ucs
1089	1046, 9238	10891046.cnv, 10461089.cnv
1256, 5352	864, 17248	12560864.cnv, 08641256.cnv, IBM01256.ucs
1256, 5352	1046, 9238	12561046.cnv, 10461256.cnv, IBM01256.ucs
1256, 5352	1200, 1208, 13488, 17584	IBM01256.ucs

Baltic:

Database server CCSIDs/CPGIDs	Database client CCSIDs/CPGIDs	Conversion table files
921, 901	1257	09211257.cnv, 12570921.cnv, IBM00921.ucs
921, 901	1200, 1208, 13488, 17584	IBM00921.ucs
1257, 5353	921, 901	12570921.cnv, 09211257.cnv, IBM01257.ucs
1257, 5353	922, 902	12570922.cnv, 09221257.cnv, IBM01257.ucs
1257, 5353	1200, 1208, 13488, 17584	IBM01257.ucs

Belarus:

Database server CCSIDs/CPGIDs	Database client CCSIDs/CPGIDs	Conversion table files
1131, 849	1251, 5347	11311251.cnv, 12511131.cnv
1131, 849	1283	11311283.cnv

Cyrillic:

Database server CCSIDs/CPGIDs	Database client CCSIDs/CPGIDs	Conversion table files
855, 872	866, 808	08550866.cnv, 08660855.cnv
855, 872	1251, 5347	08551251.cnv, 12510855.cnv
866, 808	855, 872	08660855.cnv, 08550866.cnv
866, 808	1251, 5347	08661251.cnv, 12510866.cnv
1251, 5347	855, 872	12510855.cnv, 08551251.cnv, IBM01251.ucs
1251, 5347	866, 808	12510866.cnv, 08661251.cnv, IBM01251.ucs

Database server CCSIDs/CPGIDs	Database client CCSIDs/CPGIDs	Conversion table files
1251, 5347	1124	12511124.cnv, 11241251.cnv, IBM01251.ucs
1251, 5347	1125, 848	12511125.cnv, 11251251.cnv, IBM01251.ucs
1251, 5347	1131, 849	12511131.cnv, 11311251.cnv, IBM01251.ucs
1251, 5347	1200, 1208, 13488, 17584	IBM01251.ucs

Estonia:

Database server CCSIDs/CPGIDs	Database client CCSIDs/CPGIDs	Conversion table files
922, 902	1257	09221257.cnv, 12570922.cnv, IBM00922.ucs
922, 902	1200, 1208, 13488, 17584	IBM00922.ucs

Greek:

Database server CCSIDs/CPGIDs	Database client CCSIDs/CPGIDs	Conversion table files
813, 4909	869, 9061	08130869.cnv, 08690813.cnv, IBM00813.ucs
813, 4909	1253, 5349	08131253.cnv, 12530813.cnv, IBM00813.ucs
813, 4909	1200, 1208, 13488, 17584	IBM00813.ucs
869, 9061	813, 4909	08690813.cnv, 08130869.cnv
869, 9061	1253, 5349	08691253.cnv, 12530869.cnv
1253, 5349	813, 4909	12530813.cnv, 08131253.cnv, IBM01253.ucs
1253, 5349	869, 9061	12530869.cnv, 08691253.cnv, IBM01253.ucs
1253, 5349	1200, 1208, 13488, 17584	IBM01253.ucs

Hebrew:

Database server CCSIDs/CPGIDs	Database client CCSIDs/CPGIDs	Conversion table files
856, 9048	862, 867	08560862.cnv, 08620856.cnv, IBM0856.ucs
856, 9048	916	08560916.cnv, 09160856.cnv, IBM0856.ucs
856, 9048	1255, 5351	08561255.cnv, 12550856.cnv, IBM0856.ucs
856, 9048	1200, 1208, 13488, 17584	IBM0856.ucs
862, 867	856, 9048	08620856.cnv, 08560862.cnv, IBM00862.ucs

Database server CCSIDs/CPGIDs	Database client CCSIDs/CPGIDs	Conversion table files
862, 867	916	08620916.cnv, 09160862.cnv, IBM00862.ucs
862, 867	1255, 5351	08621255.cnv, 12550862.cnv, IBM00862.ucs
862, 867	1200, 1208, 13488, 17584	IBM00862.ucs
916	856, 9048	09160856.cnv, 08560916.cnv
916	862, 867	09160862.cnv, 08620916.cnv
1255, 5351	856, 9048	12550856.cnv, 08561255.cnv, IBM01255.ucs
1255, 5351	862, 867	12550862.cnv, 08621255.cnv, IBM01255.ucs
1255, 5351	1200, 1208, 13488, 17584	IBM01255.ucs

Latin-1:

Database server CCSIDs/CPGIDs	Database client CCSIDs/CPGIDs	Conversion table files
437	850, 858	04370850.cnv, 08500437.cnv
850, 858	437	08500437.cnv, 04370850.cnv
850, 858	860	08500860.cnv, 08600850.cnv
850, 858	1114, 5210	08501114.cnv, 11140850.cnv
850, 858	1275	08501275.cnv, 12750850.cnv
860	850, 858	08600850.cnv, 08500860.cnv
1275	850, 858	12750850.cnv, 08501275.cnv

Latin-2:

Database server CCSIDs/CPGIDs	Database client CCSIDs/CPGIDs	Conversion table files
852, 9044	1250, 5346	08521250.cnv, 12500852.cnv
1250, 5346	852, 9044	12500852.cnv, 08521250.cnv, IBM01250.ucs
1250, 5346	1200, 1208, 13488, 17584	IBM01250.ucs

Simplified Chinese:

Database server CCSIDs/CPGIDs	Database client CCSIDs/CPGIDs	Conversion table files
837, 935, 1388	1200, 1208, 13488, 17584	1388ucs2.cnv
1386	1200, 1208, 13488, 17584	1386ucs2.cnv, ucs21386.cnv

Traditional Chinese:

Database server CCSIDs/CPGIDs	Database client CCSIDs/CPGIDs	Conversion table files
937, 835, 1371	950, 1370	09370950.cnv, 0937ucs2.cnv
937, 835, 1371	1200, 1208, 13488, 17584	0937ucs2.cnv
1114, 5210	850, 858	11140850.cnv, 08501114.cnv

Thailand:

Database server CCSIDs/CPGIDs	Database client CCSIDs/CPGIDs	Conversion table files
874, 1161	1200, 1208, 13488, 17584	IBM00874.ucs

Turkish:

Database server CCSIDs/CPGIDs	Database client CCSIDs/CPGIDs	Conversion table files
857, 9049	1254, 5350	08571254.cnv, 12540857.cnv
1254, 5350	857, 9049	12540857.cnv, 08571254.cnv, IBM01254.ucs
1254, 5350	1200, 1208, 13488, 17584	IBM01254.ucs

Ukraine:

Database server CCSIDs/CPGIDs	Database client CCSIDs/CPGIDs	Conversion table files
1124	1251, 5347	11241251.cnv, 12511124.cnv
1125, 848	1251, 5347	11251251.cnv, 12511125.cnv

Unicode:

Database server CCSIDs/CPGIDs	Database client CCSIDs/CPGIDs	Conversion table files
1200, 1208, 13488, 17584	813, 4909	IBM00813.ucs
1200, 1208, 13488, 17584	862, 867	IBM00862.ucs
1200, 1208, 13488, 17584	864, 17248	IBM00864.ucs
1200, 1208, 13488, 17584	874, 1161	IBM00874.ucs
1200, 1208, 13488, 17584	921, 901	IBM00921.ucs
1200, 1208, 13488, 17584	922, 902	IBM00922.ucs
1200, 1208, 13488, 17584	1046, 9238	IBM01046.ucs
1200, 1208, 13488, 17584	1250, 5346	IBM01250.ucs
1200, 1208, 13488, 17584	1251, 5347	IBM01251.ucs
1200, 1208, 13488, 17584	1253, 5349	IBM01253.ucs
1200, 1208, 13488, 17584	1254, 5350	IBM01254.ucs
1200, 1208, 13488, 17584	1255, 5351	IBM01255.ucs
1200, 1208, 13488, 17584	1256, 5352	IBM01256.ucs

Database server CCSIDs/CPGIDs	Database client CCSIDs/CPGIDs	Conversion table files
1200, 1208, 13488, 17584	1386	ucs21386.cnv, 1386ucs2.cnv

Vietnamese:

Database server CCSIDs/CPGIDs	Database client CCSIDs/CPGIDs	Conversion table files
1258, 5354	1129, 1163	12581129.cnv

API Reference

SYNCPOINT option support

The SYNCPOINT option for the sqlesetc, sqleqryc, and sqlaprep APIs is ignored as of Version 8; it is available only for backward compatibility.

New field for SQLEDBDESC structure

In the sqlecrea API, a new field has been added to support Direct I/O.

Field name
Unsigned char sqlfscaching

Description
File system caching

Values

0 File system caching is ON for the current table space

1 File system caching is OFF for the current table space

other File system caching is ON for the current table space

Correction to new field in the SQLB-TBSPQRY-DATA structure

A new field, *unsigned char fsCaching*, has been added in the SQLB-TBSPQRY-DATA structure. This new field supports Direct I/O. Although the size of the reserved bit is documented as 32–bit, the correct size is 31–bit.

Application development: Building and Running Applications

Customizing precompile and bind options for SQL procedures

The precompile and bind options for SQL procedures can be customized by setting the instance-wide DB2 registry variable, DB2_SQLROUTINE_PREPOPTS with the command:

```
db2set DB2_SQLROUTINE_PREPOPTS=<options>
```

In addition to the options documented at Version 8.2, the REOPT option is allowed:

```
BLOCKING {UNAMBIG | ALL | NO}
DATETIME {DEF | USA | EUR | ISO | JIS | LOC}
DEGREE {1 | degree-of-parallelism | ANY}
DYNAMICRULES {BIND | RUN}
EXPLAIN {NO | YES | ALL}
EXPLSNAP {NO | YES | ALL}
```

```
FEDERATED {NO | YES}
INSERT {DEF | BUF}
ISOLATION {CS | RR | UR | RS | NC}
QUERYOPT optimization-level
REOPT {ALWAYS | NONE | ONCE}
VALIDATE {RUN | BIND}
```

C/C++ compile option required (Linux on PowerPC 64-bit)

The compile option "-m64" is required for a 64-bit instance on DB2 Universal Database for Linux on PowerPC® for building DB2 C/C++ applications and routines.

Setting environment variables for Micro Focus COBOL stored procedures (Windows)

In order to run Micro Focus COBOL external routines on Windows, it is necessary to ensure that the Micro Focus COBOL environment variables are permanently set as system variables.

Procedure:

To set environment variables to system variables:

1. Open the Control Panel
2. Select **System**
3. Select the **Advanced** tab
4. Click **Environment Variables**
5. Add the variables to the **System variables** list

Setting the environment variables in the **User variables** list, at a command prompt, or in a script is insufficient.

Application development: Call Level Interface (CLI)

MapBigintCDefault CLI/ODBC configuration keyword

Keyword description:
Specify the default C type of BIGINT columns and parameter markers.

db2cli.ini keyword syntax:
MapBigintCDefault = **0** | **1** | **2**

Default setting:
The default C type representation for BIGINT data is SQL_C_BIGINT.

Usage notes:

MapBigintCDefault controls the C type that is used when SQL_C_DEFAULT is specified for BIGINT columns and parameter markers. This keyword should be used primarily with Microsoft applications, such as Microsoft Access, which cannot handle 8-byte integers. Set MapBigintCDefault as follows:

- 0 - for the default SQL_C_BIGINT C type representation
- 1 - for an SQL_C_CHAR C type representation
- 2 - for an SQL_C_WCHAR C type representation

This keyword affects the behavior of CLI functions where SQL_C_DEFAULT might be specified as a C type, such as `SQLBindParameter()`, `SQLBindCol()`, and `SQLGetData()`.

DescribeOutputLevel CLI/ODBC configuration keyword

Keyword description:

Set the level of output column describe information that is requested by the CLI driver during prepare or describe requests.

db2cli.ini keyword syntax:

DescribeOutputLevel = **0** | **1** | **2** | **3**

Default setting:

Request the describe information listed in level 2 of Table 15 on page 63.

Usage notes:

This keyword controls the amount of information the CLI driver requests on a prepare or describe request. By default, when the server receives a describe request, it returns the information contained in level 2 of Table 15 on page 63 for the result set columns. An application, however, might not need all of this information or might need additional information.

Setting the DescribeOutputLevel keyword to a level that suits the needs of the client application might improve performance because the describe data transferred between the client and server is limited to the minimum amount that the application requires. If the DescribeOutputLevel setting is set too low, it might impact the functionality of the application (depending on the application's requirements). The CLI functions to retrieve the describe information might not fail in this case, but the information returned might be incomplete.

Supported settings for DescribeOutputLevel are:

- 0 - no describe information is returned to the client application
- 1 - describe information categorized in level 1 (see Table 15 on page 63) is returned to the client application
- 2 - (default) describe information categorized in level 2 (see Table 15 on page 63) is returned to the client application
- 3 - describe information categorized in level 3 (see Table 15 on page 63) is returned to the client application

The following table lists the fields that form the describe information that the server returns when it receives a prepare or describe request. These fields are grouped into levels, and the DescribeOutputLevel CLI/ODBC configuration keyword controls which levels of describe information the CLI driver requests.

Note: Not all levels of describe information are supported by all DB2 servers. All levels of describe information are supported on the following DB2 servers: DB2 Universal Database (UDB) for Linux, and Windows Version 8 and later, DB2 UDB for z/OS Version 8 and later, and DB2 UDB for iSeries Version 5 Release 3 and later. All other DB2 servers support only the 2 or 0 setting for DescribeOutputLevel.

Table 15. Levels of describe information

Level 1	Level 2	Level 3
SQL_DESC_COUNT SQL_COLUMN_COUNT SQL_DESC_TYPE SQL_DESC_CONCISE_TYPE SQL_COLUMN_LENGTH SQL_DESC_OCTET_LENGTH SQL_DESC_LENGTH SQL_DESC_PRECISION SQL_COLUMN_PRECISION SQL_DESC_SCALE SQL_COLUMN_SCALE SQL_DESC_DISPLAY_SIZE SQL_DESC_NULLABLE SQL_COLUMN_NULLABLE SQL_DESC_UNSIGNED SQL_DESC_SEARCHABLE SQL_DESC_LITERAL_SUFFIX SQL_DESC_LITERAL_PREFIX SQL_DESC_CASE_SENSITIVE SQL_DESC_FIXED_PREC_SCALE	all fields of level 1 and: SQL_DESC_NAME SQL_DESC_LABEL SQL_COLUMN_NAME SQL_DESC_UNNAMED SQL_DESC_TYPE_NAME SQL_DESC_DISTINCT_TYPE SQL_DESC_REFERENCE_TYPE SQL_DESC_STRUCTURED_TYPE SQL_DESC_USER_TYPE SQL_DESC_LOCAL_TYPE_NAME SQL_DESC_USER_DEFINED_ TYPE_CODE	all fields of levels 1 and 2 and: SQL_DESC_BASE_COLUMN_NAME SQL_DESC_UPDATABLE SQL_DESC_AUTO_UNIQUE_VALUE SQL_DESC_SCHEMA_NAME SQL_DESC_CATALOG_NAME SQL_DESC_TABLE_NAME SQL_DESC_BASE_TABLE_NAME

Application development: Programming Client Applications

OleDbReportIsLongForLongTypes CLI/ODBC configuration keyword

Keyword description:

Makes OLE DB flag LONG data types with DBCOLUMNFLAGS_ISLONG.

db2cli.ini keyword syntax:

OleDbReportIsLongForLongTypes = **0** | **1**

Equivalent statement attribute:

SQL_ATTR_REPORT_ISLONG_FOR_LONGTYPES_OLEDB

Default setting:

LONG types (LONG VARCHAR, LONG VARCHAR FOR BIT DATA, LONG VARGRAPHIC and LONG VARGRAPHIC FOR BIT DATA) do not have the DBCOLUMNFLAGS_ISLONG flag set, which may cause the columns to be used in the WHERE clause.

Usage notes:

OLE DB's client cursor engine and OLE DB .NET Data Provider's CommandBuilder generate update and delete statements based on column information provided by the IBM DB2 OLE DB Provider. If the generated statement contains a LONG type in the WHERE clause, the statement will fail because LONG types cannot be used in a search with an equality operator. Setting the keyword OleDbReportIsLongForLongTypes to 1 will make the IBM DB2 OLE DB Provider report LONG types (LONG VARCHAR, LONG VARCHAR FOR BIT DATA, LONG VARGRAPHIC and LONG VARGRAPHIC FOR BIT DATA) with the DBCOLUMNFLAGS_ISLONG flag set. This will prevent the long columns from being used in the WHERE clause.

OleDbSQLColumnsSortByOrdinal CLI/ODBC configuration keyword

Keyword description:
Makes OLE DB's IDBSchemaRowset::GetRowset(DBSCHEMA_COLUMNS) return a row set sorted by the ORDINAL_POSITION column.

db2cli.ini keyword syntax:
OleDbSQLColumnsSortByOrdinal = **0** | **1**

Equivalent statement attribute:
SQL_ATTR_SQLCOLUMNS_SORT_BY_ORDINAL_OLEDB

Default setting:
IDBSchemaRowset::GetRowset(DBSCHEMA_COLUMNS) returns the row set sorted by the columns TABLE_CATALOG, TABLE_SCHEMA, TABLE_NAME, COLUMN_NAME.

Usage notes:

The Microsoft OLE DB specification requires that IDBSchemaRowset::GetRowset(DBSCHEMA_COLUMNS) returns the row set sorted by the columns TABLE_CATALOG, TABLE_SCHEMA, TABLE_NAME, COLUMN_NAME. The IBM DB2 OLE DB Provider conforms to the specification. However, applications that use the Microsoft ODBC Bridge provider (MSDASQL) have been typically coded to get the row set sorted by ORDINAL_POSITION. Setting the OleDbSQLColumnsSortByOrdinal keyword to 1 will make the provider return a row set sorted by ORDINAL_POSITION.

DB2 Data Source property group for the IBM DB2 OLE DB Provider

The IBM DB2 OLE DB Provider has added a new property group: DB2 Data Source. The property set for DB2 Data Source is DBPROPSET_DB2DATASOURCE.

The GUID for the property set is
`{0x8a80412a,0x7d94,0x4fec,{0x87,0x3e,0x6c,0xd1,0xcd,0x42,0x0d,0xcd}}`

DBPROPSET_DB2DATASOURCE has three properties:

- DB2PROP_REPORTISLONGFORLONGTYPES
- DB2PROP_RETURNCHARASWCHAR
- DB2PROP_SORTBYORDINAL

DB2PROP_REPORTISLONGFORLONGTYPES:

```
#define DB2PROP_REPORTISLONGFORLONGTYPES 4
Property group: DB2 Data Source
Property set: DB2PROPSET_DATASOURCE
Type: VT_BOOL
Typical R/W: R/W
Description: Report IsLong for Long Types
```

OLE DB's client cursor engine and OLE DB .NET Data Provider's CommandBuilder generate update and delete statements based on column information provided by the IBM DB2 OLE DB Provider. If the generated statement contains a LONG type in the WHERE clause, the statement will fail because LONG types cannot be used in a search with an equality operator.

Table 16. DB2PROP_REPORTISLONGFORLONGTYPES values

Values	Meaning
VARIANT_TRUE	Will make the IBM DB2 OLE DB Provider report LONG types (LONG VARCHAR, LONG VARCHAR FOR BIT DATA, LONG VARGRAPHIC, and LONG VARGRAPHIC FOR BIT DATA) with the DBCOLUMNFLAGS_ISLONG flag set. This will prevent the long columns from being used in the WHERE clause.
VARIANT_FALSE	DBCOLUMNFLAGS_ISLONG is not set for LONG VARCHAR, LONG VARCHAR FOR BIT DATA, LONG VARGRAPHIC and LONG VARGRAPHIC FOR BIT DATA. This is the default.

DB2PROP_RETURNCHARASWCHAR:

```
#define DB2PROP_RETURNCHARASWCHAR 2
Property group: DB2 Data Source
Property set: DB2PROPSET_DATASOURCE
Type: VT_BOOL
Typical R/W: R/W
Description: Return Char as WChar
```

Table 17. DB2PROP_RETURNCHARASWCHAR values

Values	Meaning
VARIANT_TRUE	OLE DB describes columns of type CHAR, VARCHAR, LONG VARCHAR, or CLOB as DBTYPE_WSTR. The code page of data implied in ISequentialStream will be UCS-2. This is the default.
VARIANT_FALSE	OLE DB describes columns of type CHAR, VARCHAR, LONG VARCHAR, or CLOB as DBTYPE_STR. The code page of data implied in ISequentialStream will be the local code page of the client.

DB2PROP_SORTBYORDINAL:

```
#define DB2PROP_SORTBYORDINAL 3
Property group: DB2 Data Source
Property set: DB2PROPSET_DATASOURCE
Type: VT_BOOL
Typical R/W: R/W
Description: Sort By Ordinal
```

The Microsoft OLE DB specification requires that IDBSchemaRowset::GetRowset(DBSCHEMA_COLUMNS) returns the row set sorted by the columns TABLE_CATALOG, TABLE_SCHEMA, TABLE_NAME, COLUMN_NAME. The IBM DB2 OLE DB Provider conforms to the specification. However, applications that use the Microsoft ODBC Bridge provider (MSDASQL) have been typically coded to get the row set sorted by ORDINAL_POSITION.

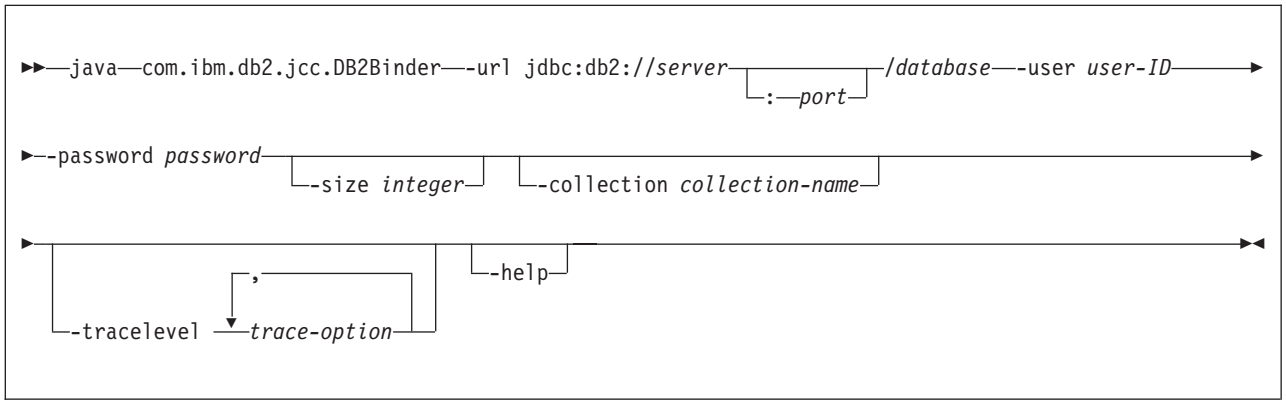
Table 18. DB2PROP_SORTBYORDINAL values

Values	Meaning
VARIANT_TRUE	Will make the provider return a row set sorted by ORDINAL_POSITION.
VARIANT_FALSE	Will make the provider return a row set sorted by TABLE_CATALOG, TABLE_SCHEMA, TABLE_NAME, COLUMN_NAME. This is the default.

Incorrect URL syntax in the DB2Binder syntax diagram

In the topic "Installing the DB2 Universal JDBC Driver", the DB2Binder syntax diagram incorrectly defines the URL syntax for the DB2 Universal JDBC Driver. The correct representation of the URL syntax for DB2Binder is shown in the following diagram:

DB2Binder syntax:



Rerouting DB2 Universal JDBC driver clients

The automatic client reroute feature in DB2 Universal Database (UDB) for Linux and Windows allows client applications to recover from a loss of communication with the server so that they can continue to work with minimal interruption.

Whenever a server locks up, each client that is connected to that server receives a communication error, which terminates the connection and results in an application error. When availability is important, you should have a redundant setup or failover support. (Failover is the ability of a server to take over operations when another server fails.) In either case, the DB2 Universal JDBC driver client attempts to reestablish the connection to a new server, or to the original server, which might be running on a failover node. When the connection is reestablished, the application receives an SQLException that informs it of the transaction failure, but the application can continue with the next transaction.

Restrictions:

- DB2 Universal JDBC driver client reroute support is available only for connections that use the javax.sql.DataSource interface.
- Before a client application can recover from a loss of communication, an alternate server location must be specified at the server. The database administrator specifies the alternate server with the UPDATE ALTERNATE SERVER FOR DATABASE command.

Procedure:

After the database administrator specifies the alternate server location on a particular database at the server instance, the primary and alternate server locations are returned back to the client at connect time. The DB2 Universal JDBC driver creates an instance of Referenceable object DB2ClientRerouteServerList and stores that instance in its transient memory. If communication is lost, the DB2 Universal JDBC driver tries to reestablish the connection using the server information that is returned from the server.

The clientRerouteServerListJNDIName DataSource property provides additional client reroute support at the client; clientRerouteServerListJNDIName has two functions:

- It allows alternate server information to persist across JVMs
- It provides an alternate server location in case the first connection to the database server fails

The clientRerouteServerListJNDIName identifies a JNDI reference to a DB2ClientRerouteServerList instance in a JNDI repository for alternate server information. After a successful connection to the primary server, the alternate server information that is provided by clientRerouteServerListJNDIName is overwritten by the information from the server. The DB2 Universal JDBC driver will attempt to propagate the updated information to the JNDI store after a failover if the clientRerouteServerListJNDIName property is defined. If clientRerouteServerListJNDIName is specified, primary server information specified in DB2ClientRerouteServerList will be used for connection. If the primary server is not specified, serverName information specified on the data source will be used.

DB2ClientRerouteServerList is a serializable Java bean with four properties:

- alternateServerName
- alternatePortNumber
- primaryServerName
- primaryPortNumber

Getter and setter methods for accessing these properties are provided. The definition of the DB2ClientRerouteServerList class is as follows:

```
package com.ibm.db2.jcc;
public class DB2ClientRerouteServerList
  implements java.io.Serializable,
  javax.naming.Referenceable
{
  public String[] alternateServerName;
  public synchronized void
    setAlternateServerName(String[] alternateServer);
  public String[] getAlternateServerName();
  public int[] alternatePortNumber;
  public synchronized void
    setAlternatePortNumber(int[] alternatePortNumberList);
  public int[] getAlternatePortNumber();

  public synchronized void
    setPrimaryServerName (String primaryServerName);
  public String getPrimaryServerName ();
  public synchronized void setPrimaryPortNumber (int primaryPortNumber)
  public int getPrimaryPortNumber ();
}
```

A newly established failover connection is configured with the original data source properties, except for the server name and port number. In addition, any DB2 UDB special registers that were modified during the original connection are reestablished in the failover connection by DB2 Universal Driver JDBC driver.

When a communication failure occurs, the DB2 Universal JDBC driver first attempts recovery to the primary server. If this fails, the driver attempts to connect to the alternate location (failover). After a connection is reestablished, the driver throws a java.sql.SQLException to the application with SQLCODE -4498, to

indicate to the application that the connection has been automatically reestablished to the alternate server. The application can then retry its transaction.

Procedure for make DB2ClientRerouteServerList persistent:

To setup storage to make DB2ClientRerouteServerList persistent, follow these steps:

1. Create an instance of DB2ClientRerouteServerList and bind that instance to the JNDI registry. For example:

```
// Create a starting context for naming operations
InitialContext registry = new InitialContext();
// Create a DB2ClientRerouteServerList object
DB2ClientRerouteServerList address=new DB2ClientRerouteServerList();

// Set the port number and server name for the primary server
address.setPrimaryPortNumber(50000);
address.setPrimaryServerName("mvs1.sj.ibm.com");

// Set the  port number and server name for the alternate server
int[] port = {50002};
String[] server = {"mvs3.sj.ibm.com"};
address.setAlternatePortNumber(port);
address.setAlternateServerName(server);

registry.rebind("serverList", address);
```

2. Assign the JNDI name of the DB2ClientRerouteServerList object to DataSource property clientRerouteServerListJNDIName. For example:

```
datasource.setClientRerouteServerListJNDIName("serverList");
```

Customizing the DB2 Universal JDBC driver configuration properties

The DB2 Universal JDBC driver configuration properties let you set property values that have driver-wide scope. Those settings apply across applications and DataSource instances. You can change the settings without having to change application source code or DataSource characteristics.

Each DB2 Universal JDBC driver configuration property setting is of the following form:

property=value

If the configuration property begins with db2.jcc.override, the configuration property is applicable to all connections and overrides any Connection or DataSource property with the same property name. If the configuration property begins with db2.jcc or db2.jcc.default, the configuration property value is a default. Connection or DataSource property settings override that value.

Procedure:

To set configuration properties:

- Set the configuration properties as Java system properties. Those settings override any other settings.

  For standalone Java applications, you can set the configuration properties as Java system properties by specifying `-Dproperty=value` for each configuration property when you execute the **java** command.
- Set the configuration properties in a resource whose name you specify in the db2.jcc.propertiesFile Java system property. For example, you can specify an absolute path name for the db2.jcc.propertiesFile value.

For standalone Java applications, you can set the configuration properties by specifying the `-Ddb2.jcc.propertiesFile=path` option when you execute the **java** command.

- Set the configuration properties in a resource named DB2JccConfiguration.properties. A standard Java resource search is used to find DB2JccConfiguration.properties. The DB2 Universal JDBC driver searches for this resource only if you have not set the db2.jcc.propertiesFile Java system property.

  DB2JccConfiguration.properties can be a standalone file, or it can be included in a JAR file.

  If DB2JccConfiguration.properties is a standalone file, the path for DB2JccConfiguration.properties must be in the CLASSPATH concatenation.

  If DB2JccConfiguration.properties is in a JAR file, the JAR file must be in the CLASSPATH concatenation.

You can set the following DB2 Universal JDBC driver configuration properties. All properties are optional.

db2.jcc.override.traceFile
Enables the DB2 Universal JDBC driver trace for Java driver code, and specifies the name on which the trace file names are based.

Specify a fully qualified file name for the db2.jcc.override.traceFile property value.

The db2.jcc.override.traceFile property overrides the traceFile property for a Connection or DataSource object.

For example, specifying the following setting for db2.jcc.override.traceFile enables tracing of the DB2 Universal JDBC Driver Java code to a file named `/SYSTEM/tmp/jdbctrace`:

```
db2.jcc.override.traceFile=/SYSTEM/tmp/jdbctrace
```

You should set the trace properties under the direction of IBM Software Support.

db2.jcc.sqljUncustomizedWarningOrException
Specifies the action that the DB2 Universal JDBC driver takes when an uncustomized SQLJ application runs. db2.jcc.sqljUncustomizedWarningOrException can have the following values:

0 The DB2 Universal JDBC driver does not generate a Warning or Exception when an uncustomized SQLJ application runs. This is the default.

1 The DB2 Universal JDBC driver generates a Warning when an uncustomized SQLJ application runs.

2 The DB2 Universal JDBC driver generates an Exception when an uncustomized SQLJ application runs.

db2secFreeToken function removed

The db2secFreeToken function (Free memory held by token) is no longer part of the db2secGssapiServerAuthFunctions_1 user authentication plug-in API.

Deploy custom security plug-ins carefully

The integrity of your DB2 Universal Database (UDB) installation can be compromised if the deployment of security plug-ins are not adequately coded,

reviewed, and tested. DB2 UDB takes precautions against many common types of failures, but it cannot guarantee complete integrity when user-written security plug-ins are deployed.

Security plug–ins

If you are using your own customized security plug–in, you can use a user ID of up to 255 characters on a connect statement issued through the CLP or a dynamic SQL statement.

Security plug–in APIs

For the db2secGetGroupsForUser, db2secValidatePassword, and db2secGetAuthIDs APIs, the input parameter, *dbname*, can be null and its corresponding length input parameter, *dbnamelen*, will be set to 0.

Security plug–in naming conventions (Linux)

.so is now accepted as a file name extension for user-written security plug–in libraries on all Linux platforms.

On AIX, security plug–in libraries can have an extension of *.a* or *.so*. If both versions of the plug–in library exist, the *.a* version is used.

For HP–UX on PA–RISC, security plug–in libraries can have an extension of *.sl* or *.so*. If both versions of the plug–in library exist, the *.sl* version is used.

On all other Linux platforms, *.so* is the only supported file name extension for security plug–in libraries.

GSS-API security plug-ins do not support multiple-flow authentication

GSS-API authentication is limited to flowing one token from the client to the server and one token from the server to the client. These tokens are obtained from `gss_init_sec_context()` on the client and from `gss_accept_sec_context()` on the server. GSS-API plug-ins attempting additional flows will generate a security plug-in unexpected error, causing the connection to fail.

GSS-API security plug-ins do not support message encryption and signing

Message encryption and signing is not available in GSS-API security plug-ins.

Implicit ending of transactions in stand-alone applications

All application terminations (normal and abnormal) implicitly roll back outstanding units of work, regardless of operating system.

Distributed transaction support

In the What's new documentation for DB2 Universal Database (UDB) Version 8.2, the Distributed transaction support information for the DB2 Universal JDBC driver improvements section has incorrect information. The last sentence of this section is incorrect. The correct information is:

As of Version 8.2, DB2 UDB provides support for distributed transaction processing that conforms to the XA specification. This support implements the Java 2 Platform, Enterprise Edition (J2EE) Java Transaction Service (JTS) and Java Transaction API (JTA) specifications.

Application development: Programming Server Applications

Common language run time (CLR) routine execution control modes (EXECUTION CONTROL clause)

As a database administrator or application developer, you might want to protect the assemblies associated with your DB2 Universal Database (UDB) external routines from unwelcome tampering by restricting the actions of routines at run time. DB2 .NET CLR routines support the specification of an execution control mode that identifies what types of actions a routine is allowed to perform at run time. At run time, DB2 UDB can detect if the routine attempts to perform actions beyond the scope of its specified execution control mode, which can be helpful when determining whether an assembly has been compromised.

To set the execution control mode of a CLR routine, specify the optional EXECUTION CONTROL clause in the CREATE statement for the routine. Valid modes are:

- SAFE
- FILEREAD
- FILEWRITE
- NETWORK
- UNSAFE

To modify the execution control mode in an existing CLR routine, execute the ALTER PROCEDURE or ALTER FUNCTION statement.

If the EXECUTION CONTROL clause is not specified for a CLR routine, by default the CLR routine runs using the most restrictive execution control mode, SAFE. Routines that are created with this execution control mode can only access resources that are controlled by the database manager. Less restrictive execution control modes allow a routine to access files on the local file system (FILEREAD or FILEWRITE) or on the network. The execution control mode UNSAFE specifies that no restrictions are to be placed on the behavior of the routine. Routines defined with UNSAFE execution control mode can execute binary code.

These control modes represent a hierarchy of allowable actions, and a higher-level mode includes the actions that are allowed below it in the hierarchy. For example, execution control mode NETWORK allows a routine to access files on the network, files on the local file system, and resources that are controlled by the database manager. Use the most restrictive execution control mode possible and avoid using the UNSAFE mode.

If DB2 UDB detects at run time that a CLR routine is attempting an action outside of the scope of its execution control mode, DB2 UDB returns an error (SQLSTATE 38501).

The EXECUTION CONTROL clause can only be specified for LANGUAGE CLR routines. The scope of applicability of the EXECUTION CONTROL clause is limited to the .NET CLR routine itself, and does not extend to any other routines that it might call.

Maximum decimal precision and scale in common language run time (CLR) routines

The DECIMAL data type in DB2 Universal Database (UDB) is represented with a precision of 31 digits and a scale of 28 digits. The .NET CLR System.Decimal data type is limited to a precision of 29 digits and a scale of 28 digits. Therefore, DB2 UDB external CLR routines must not assign a value greater than (2^96)−1, the highest value that can be represented using a 29 digit precision and a 28 digit scale, to a System.Decimal data type variable. DB2 UDB raises a run time error (SQLSTATE 22003, SQLCODE -413) if such an assignment occurs.

When a routine CREATE statement is executed, if a DECIMAL data type parameter is defined with a scale greater than 28, DB2 UDB raises an error (SQLSTATE 42611, SQLCODE -604).

Command Reference

db2inidb - Initialize a mirrored database command

Do not issue the `db2 connect to` *`database`* command prior to issuing the `db2inidb` *`database`* `as mirror` command.

Attempting to connect to a split mirror database prior to initializing it erases the log files needed for roll forward recovery.

The connect sets your database back to the state it was in when you suspended the database. If the database is marked as consistent at the time of the suspend, DB2 Universal Database concludes there is no need for crash recovery and empties the logs for future use. If this situation occurs, attempting to rollforward causes a SQL4970 error.

Usage note for the db2iupdt command

Starting with Version 8.2, when you update a DB2 Universal Database instance with the **db2iupdt** command, you must first stop any DB2 processes running against that instance.

db2pd - Monitor and troubleshoot DB2 command

Additional parameters for the **db2pd** command include:

-hadr Reports High Availability Disaster Recovery information. Descriptions of each reported element can be found in the High availability disaster recovery section of the *System Monitor Guide and Reference* .

-utilities
Reports Utility information. Descriptions of each reported element can be found in the Utilities section of the *System Monitor Guide and Reference* .

New parameter for the db2sqljcustomize command

The db2sqljcustomize command has a new parameter.

db2sqljcustomize - DB2 SQLJ Profile Customizer command:

-storebindoptions
Stores the value of the `-bindoptions` and `-staticpositioned` values in the serialized profile. If these values are not specified when invoking the dbsqljbind tool, the values stored in the serialized profile are used. When the Customizer is invoked with `.grp` file, the values are stored in each individual `.ser` file. The stored values can be viewed using db2sqljprint tool.

New parameter for the sqlj command

The sqlj command has a new parameter.

sqlj - DB2 SQLJ Translator command:

-db2optimize
Specifies that the SQLJ translator generates code for a connection context class that is optimized for DB2 Universal Database. This option optimizes the code for the user defined context but not for the default context. When you run the SQLJ translator with this option, the DB2 Universal JDBC driver file `db2jcc.jar` must be in the CLASSPATH for compiling the generated Java application.

Update to the SET CLIENT command

The SET™ CLIENT command specifies connection settings for the back end process.

The command parameter SYNCPOINT for this command is ignored as of Version 8. SYNCPOINT continues to be included for backward compatibility.

Update to the PRECOMPILE command

The PRECOMPILE command processes an application program source file containing embedded SQL statements. A modified source file is produced, containing host language calls for the SQL and, by default, a package is created in the database.

The command parameter SYNCPOINT for this command is ignored as of Version 8. SYNCPOINT is continued to be included for backward compatibility.

Update to the UPDATE HISTORY FILE command

Updates the location, device type, or comment in a history file entry.

The command parameter STATUS specifies a new status for an entry.

Previous documentation incorrectly states that the STATUS command parameter can have a value of “I” to mark the entry as inactive. Valid values are:

A Marks the entry as active.

E Marks the entry as expired.

Updates to the EXPORT and IMPORT commands

The complete "Required connection " subsection for the EXPORT and IMPORT commands is as follows:

Required connection:

Database. If implicit connect is enabled, a connection to the default database is established. Utility access to Linux or Windows database servers from Linux or Windows clients must be a direct connection through the engine and not through a DB2 Connect gateway or loop back.

Update to the LOAD command

The complete information for the INDEXING MODE parameter's AUTOSELECT value is as follows:

INDEXING MODE:

AUTOSELECT

The load utility will automatically decide between REBUILD or INCREMENTAL mode. The decision is based on the amount of data being loaded and the depth of the index tree. Information relating to the depth of the index tree is stored in the index object. RUNSTATS is not required to populate this information. AUTOSELECT is the default indexing mode.

File type modifiers for the load utility

The `SET INTEGRITY` command in the description for the "generatedoverride" modifier has been updated.

The description for the "usedefaults" modifier has also been updated.

The updates are as follows:

Table 19. Valid file type modifiers for load: All file formats

Modifier	Description

generatedoverride	This modifier instructs the load utility to accept user-supplied data for all generated columns in the table (contrary to the normal rules for these types of columns). This is useful when migrating data from another database system, or when loading a table from data that was recovered using the RECOVER DROPPED TABLE option on the ROLLFORWARD DATABASE command. When this modifier is used, any rows with no data or NULL data for a non-nullable generated column will be rejected (SQL3116W).
	Note: When this modifier is used, the table will be placed in CHECK PENDING state. To take the table out of CHECK PENDING state without verifying the user-supplied values, issue the following command after the load operation:
	`SET INTEGRITY FOR < table-name > GENERATED COLUMN IMMEDIATE UNCHECKED`
	To take the table out of CHECK PENDING state and force verification of the user-supplied values, issue the following command after the load operation:
	`SET INTEGRITY FOR < table-name > IMMEDIATE CHECKED.`
	This modifier cannot be used with either the `generatedmissing` or the `generatedignore` modifier.

Table 19. Valid file type modifiers for load: All file formats (continued)

Modifier	Description
usedefaults	If a source column for a target table column has been specified, but it contains no data for one or more row instances, default values are loaded. Examples of missing data are: • For DEL files: two adjacent column delimiters (",,") or two adjacent column delimiters separated by an arbitrary number of spaces (", ,") are specified for a column value. • For DEL/ASC/WSF files: A row that does not have enough columns, or is not long enough for the original specification. **Note:** For ASC files, NULL column values are not considered explicitly missing, and a default will not be substituted for NULL column values. NULL column values are represented by all space characters for numeric, date, time, and /timestamp columns, or by using the NULL INDICATOR for a column of any type to indicate the column is NULL. Without this option, if a source column contains no data for a row instance, one of the following occurs: • For DEL/ASC/WSF files: If the column is nullable, a NULL is loaded. If the column is not nullable, the utility rejects the row.

File type modifiers for the import utility

The description for the "usedefaults" and "codepage=x" modifiers have been updated as follows:

Table 20. Valid file type modifiers for import: All file formats

Modifier	Description
usedefaults	If a source column for a target table column has been specified, but it contains no data for one or more row instances, default values are loaded. Examples of missing data are: • For DEL files: two adjacent column delimiters (",,") or two adjacent column delimiters separated by an arbitrary number of spaces (", ,") are specified for a column value. • For DEL/ASC/WSF files: A row that does not have enough columns, or is not long enough for the original specification. **Note:** For ASC files, NULL column values are not considered explicitly missing, and a default will not be substituted for NULL column values. NULL column values are represented by all space characters for numeric, date, time, and /timestamp columns, or by using the NULL INDICATOR for a column of any type to indicate the column is NULL. Without this option, if a source column contains no data for a row instance, one of the following occurs: • For DEL/ASC/WSF files: If the column is nullable, a NULL is loaded. If the column is not nullable, the utility rejects the row.

Table 21. Valid file type modifiers for import: ASCII file formats (ASC/DEL)

Modifier	Description
codepage=*x*	*x* is an ASCII character string. The value is interpreted as the code page of the data in the output data set. Converts character data from this code page to the application code page during the import operation. The following rules apply: • For pure DBCS (graphic) mixed DBCS, and EUC, delimiters are restricted to the range of x00 to x3F, inclusive. • `nullindchar` must specify symbols included in the standard ASCII set between code points x20 and x7F, inclusive. This refers to ASCII symbols and code points. **Notes:** 1. The `codepage` modifier cannot be used with the `lobsinfile` modifier. 2. If data expansion occurs when the code page is converted from the application code page to the database code page, the data might be truncated and loss of data can occur.

ATTACH command

The USER parameter of the ATTACH command specifies the authentication identifier. When attaching to a DB2 Universal Database instance on a Windows operating system, the user name can be specified in a format compatible with Microsoft Windows NT Security Account Manager (SAM). The qualifier must be a NetBIOS style name, which has a maximum length of 15 characters. For example, `domainname\username`.

RECOVER DATABASE command

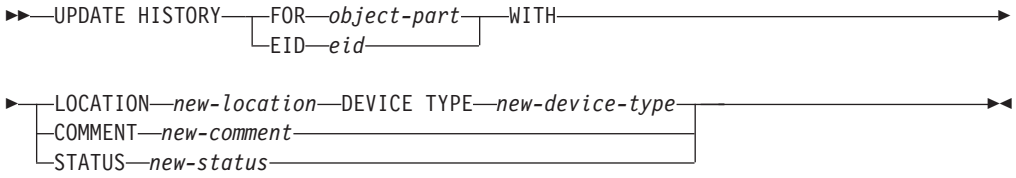
In the Examples section of the RECOVER DATABASE Command for Version 8.2 documentation, timestamps are incorrectly formatted as `yyyy:mm:dd:hh:mm:ss`.

The correct format is `yyyy-mm-dd-hh.mm.ss`

UPDATE HISTORY FILE command

The UPDATE HISTORY FILE command updates the location, device type, comment, or status in a history file entry.

Updated command syntax:



Updated command parameters:

FOR *object-part*
Specifies the identifier for the history entry to be updated. It is a timestamp with an optional sequence number from `001` to `999`.

Note: Cannot be used to update entry status. To update the entry status, specify an EID instead.

STATUS *new-status*

Specifies a new status for an entry. Only backup entries can have their status updated. Valid values are:

A Active. Most entries are active.

I Inactive. Backup images that are no longer on the active log chain become inactive.

E Expired. Backup images that are no longer required because there are more than NUM_DB_BACKUPS active images are flagged as expired.

D Backup images that are no longer available for recovery should be marked as having been deleted.

db2updv8 - Update database to version 8 current level command

This command updates the system catalogs in a database to support the current level in the following ways:

- Introduces new routines shipped with the current database manager
- Catalogs the ODBC/CLI/JDBC schema procedures as trusted procedures to improve performance and scalability
- Fixes two typographical errors in the SYSPROC.SNAPSHOT_QUIESCERS table function's return parameters:
  - OUIESCER_TBS_ID corrected to QUIESCER_TBS_ID
  - OUIESCER_STATE corrected to QUIESCER_STATE
- Fixes LOB columns' column descriptors such that "logged" bit is consistent with the catalog tables
- Creates SYSIBM.SYSREVTYPEMAPPINGS view
- Updates the view definitions for SYSSTAT.COLUMNS and SYSSTAT.TABLES
- Updates the SYSCOLDIST.DISTCOUNT column to be updateable
- Updates the SYSINDEXES.TBSPACEID column such that column flags are not set
- Fixes parameter lengths of 17 table functions. Refer to the *Column expansion for snapshot UDFs* topic in the **Documentation updates | SQL Administrative Routines** section of the *Version 8.2 Release Notes* for a complete listing of functions
- Alters 33 SYSPROC snapshot UDFs to THREADSAFE:
  - SNAPSHOT_DBM
  - SNAPSHOT_FCM
  - SNAPSHOT_FCMNODE
  - SNAPSHOT_SWITCHES
  - SNAPSHOT_APPL_INFO
  - SNAPSHOT_APPL
  - SNAPSHOT_STATEMENT
  - SNAPSHOT_LOCKWAIT
  - SNAPSHOT_AGENT
  - SNAPSHOT_SUBSECT
  - SNAPSHOT_DATABASE
  - SNAPSHOT_BP

- SNAPSHOT_LOCK
- SNAPSHOT_TABLE
- SNAPSHOT_DYN_SQL
- SNAPSHOT_TBS
- SNAPSHOT_TBS_CFG
- SNAPSHOT_QUIESCERS
- SNAPSHOT_CONTAINER
- SNAPSHOT_RANGES
- SNAPSHOT_TBREORG
- HEALTH_DBM_INFO
- HEALTH_DBM_HI
- HEALTH_DBM_HI_HIS
- HEALTH_DB_INFO
- HEALTH_DB_HI
- HEALTH_DB_HI_HIS
- HEALTH_TBS_INFO
- HEALTH_TBS_HI
- HEALTH_TBS_HI_HIS
- HEALTH_CONT_INFO
- HEALTH_CONT_HI
- HEALTH_CONT_HI_HIS

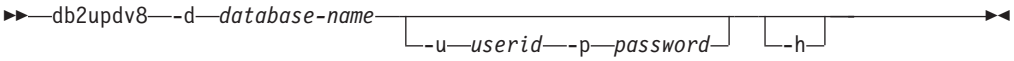
Authorization:

sysadm

Required connection:

Database. This command automatically establishes a connection to the specified database.

Command syntax:



Command parameters:

-d database-name
Specifies the name of the database to be updated.

-u userid
Specifies the user ID.

-p password
Specifies the password for the user.

-h Displays help information. When this option is specified, all other options are ignored, and only the help information is displayed.

Example:

After installing the current level (a FixPak or a new version), update the system catalog in the sample database by issuing the following command:

```
db2updv8 -d sample
```

Usage notes:

1. This command can be used only on a database running DB2 Universal Database Version 8.1.2 or later. If the command is issued more than once, no errors are reported and each of the catalog updates is applied only once.
2. To enable the new built-in functions, all applications must disconnect from the database and the database must be deactivated if it has been activated.

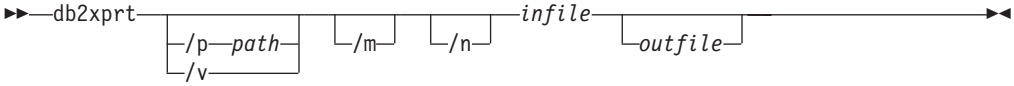
Formatting trap files (Windows)

A new tool, db2xprt.exe, is available to let you format trap files (*.TRP). This tool formats DB2 Universal Database's binary trap files into a human readable ASCII file. Trap files are located in the instance directory (DB2INSTPROF) by default or in the diagnostic data directory path if the DIAGPATH database manager configuration parameter is set.

Authorization:

You must have access to the DIAGPATH directory.

Command syntax:



Command parameters:

/p *path* A semicolon (;) separated path that points to the location or locations where the binary files and PDB files are located.

/v Displays version information.

/m Formats a memory dump along with the rest of the trap file.

/n Format data without regard to line number information.

infile Specifies the input file.

outfile Specifies the output file.

Data Movement

Index information lost when the import utility is used

You can use the import utility to recreate a table that was saved through the export utility.

The *Data Movement* topic "Using import to recreate an exported table" outlines attributes of the original table are *not* retained. In addition to the attributes which have already been documented, the following attributes are *not* retained:

- Index information:
  - Include columns (if any)
  - Index name, if the index is a primary-key index

– Descending order, if the index is a primary-key index (Ascending is the default)

Data Recovery and High Availability

High availability disaster recovery overview

When performing the START HADR, STOP HADR, or TAKEOVER HADR commands, the corresponding error codes might be generated: SQL01767N, SQL01769N, or SQL01770N with a reason code of 98. The reason code indicates that there is no installed license for HADR on the server where the command was run. To correct the problem, install a valid HADR license using **db2licm** or install a version of the server that contains a valid HADR license as part of its distribution.

Cross-platform backup and restore support

DB2 Universal Database (UDB) supports cross-platform backup and restore operations. You can restore databases created on a DB2 UDB Version 8 32-bit Windows platform to a DB2 UDB Version 8 64-bit Windows platform, or the reverse. You can restore databases created on a DB2 UDB Version 8 32-bit Linux x86 platform to a DB2 UDB Version 8 64-bit Linux x86-64 or IA64 platform, or the reverse. You can restore databases created on DB2 UDB Version 8 AIX, HP-UX, Linux PPC, Linux zSeries, or the Solaris Operating Environment platforms, in 32-bit or 64-bit, to DB2 UDB Version 8 AIX, HP-UX, Linux PPC, Linux zSeries, or Solaris Operating Environment platforms (32-bit or 64-bit).

Backing up to tape (Linux)

The maximum block size limit for 3480 and 3490 tape devices on Linux is 61 440 bytes

Table 22. Maximum block size limit for 3480 and 3490 tape devices on Linux

Device	Attachment	Block Size Limit	DB2 buffer size limit (in 4-KB pages)
3480	s370	61 440	15
3490	s370	61 440	15

Tivoli Storage Manager

When calling the BACKUP DATABASE or RESTORE DATABASE commands, you can specify that you want to use the Tivoli® Storage Manager (TSM) product to manage database or table space backup or restore operation. The minimum required level of TSM client API is Version 4.2.0, except on the following:

- 64-bit Solaris systems which require TSM client API Version 4.2.1.
- 64-bit Windows NT operating systems which require TSM client API Version 5.1.
- 32-bit Linux for iSeries and pSeries® which requires at minimum TSM client API Version 5.1.5
- 64-bit Linux for iSeries and pSeries which requires at minimum TSM client API Version 5.2.2
- 64-bit Linux on AMD Opteron systems which require a minimum TSM client API Version 5.2.0.
- Linux for zSeries 64-bit which requires a minimum TSM client API Version 5.2.2.

Value restrictions for the HADR local host and local service parameters

When specifying values for the high availability disaster recovery (HADR) local host and local service parameters (HADR_LOCAL_SVC and HADR_REMOTE_SVC) while preparing an **update database configuration** command, the values must be ports that are not in use for any other service. If the parameters are being configured using the Linux command line, the values should be also set in the `/etc/services` file.

Additional system requirements for high availability disaster recovery

If you create a table space on the primary database and log replay fails on the standby database because the containers are not available, the primary database does not receive an error message stating that the log replay failed.

To check for log replay errors, you must monitor the db2diag.log and the administration log on the standby database when you are creating new table spaces.

If a takeover operation occurs, the new table space that you created is not available on the new primary database. To recover from this situation, restore the table space on the new primary database from a backup image.

In the following example, table space MY_TABLESPACE is restored on database MY_DATABASE before it is used as the new primary database:

1. `db2 connect to my_database`
2. `db2 list tablespaces show detail`

   Note: Run the **db2 list tablespaces show detail** command to show the status of all table spaces and to obtain the table space ID number required for Step 5.
3. `db2 stop hadr on database my_database`
4. `db2 "restore database my_database tablespace (my_tablespace) online redirect"`
5. `db2 "set tablespace containers for my_tablespace_ID_# ignore rollforward container operations using (path '/my_new_container_path/')"`
6. `db2 "restore database my_database continue"`
7. `db2 rollforward database my_database to end of logs and stop tablespace "(my_tablespace)"`
8. `db2 start hadr on database my_database as primary`

Non-replicated operations for high availability disaster recovery

Version 8.2 documentation states:

BLOBs and CLOBs are not replicated; however, the space for them will be allocated on the standby database.

The statement should read as follows:

Non-logged BLOBs and CLOBs are not replicated; however, the space for them will be allocated on the standby database.

HADR does not support raw logs

High availability disaster recovery (HADR) does not support the use of raw I/O (direct disk access) for database log files. If HADR is started with the START HADR command, or if the database is restarted with HADR configured, and raw logs are detected, the associated command will fail with SQL1768N reason code "9".

Data Warehouse Center

Defining DB2 warehouse sources and targets

The following update affects two Data Warehouse Center topics:

- Defining DB2 warehouse sources
- Defining a warehouse target

By default, when you define a warehouse source or a warehouse target, the number of tables that are returned is 250. However, you can use the new environment variable VWS_MAX_TABLELIST to set the number of tables that are returned. The maximum number of tables that can be returned is 40 000. This number might be less depending on the size of the table names in the list. It is recommended that you specify a number much smaller than 40 000.

Updates to the Business Intelligence tutorial

Verifying that the DWCTBC and TBC_MD databases are registered with ODBC:

In Version 8, the control database, TBC_MD that is used in the tutorial, does not need to be a system ODBC data source. However, the target database or database source DWCTBC must be a system ODBC data source.

Opening the Define Warehouse Source notebook:

The procedure for opening the Define Warehouse Source notebook for the Tutorial Relational Source has changed.

Procedure:

To open the Define Warehouse Source notebook for the Tutorial Relational Source:

1. From the Data Warehouse Center window, right-click the **Warehouse Sources** folder.
2. Click **Define** —> **ODBC** —> **DB2** —> **DB2 Family**.

The Define Warehouse Source notebook opens.

Opening the Define Warehouse Target notebook:

The procedure for opening the Define Warehouse Target notebook has changed.

Procedure:

To open the Define Warehouse Target notebook:

1. From the Data Warehouse Center window, right-click the **Warehouse Targets** folder.
2. Click **Define** —> **ODBC** —> **DB2** —> **DB2 Family**.

The Define Warehouse Target notebook opens.

Setting the purge limit for warehouse log files

The log file holds records until a designated count limit is reached. The default count limit is 1000 records. Typically, each job that you run creates 12 to 15 log records. Set the purge limit to a number that meets your needs by updating the **Purge log when total records equal** field on the Server tab of the Warehouse Properties page.

Data Warehouse Center support for CURSOR load

The DB2 Universal Database Load step now allows a view or a table to be used as the source to the step, resulting in a LOAD FROM CURSOR.

In order to map columns in the wizard for CURSOR load, the **Map columns based on column positions found in the input file** radio button must be selected.

Unicode warehouse control database migration and limitations

Starting with Version 8.2 of the Data Warehouse Center, the warehouse control database must be a Unicode database. If you have a Unicode warehouse control database from a version of the Data Warehouse Center that is before Version 8.2, you still must create a new Unicode control database by using the Warehouse Control Database Management tool.

When you migrate a warehouse control database from a version of the Data Warehouse Center that is before Version 8.2, the Data Warehouse Center Control Database Management tool runs the **db2move** command to move the data to a new Unicode control database. During this process, windows appear that show the progress of the **db2move** command. This migration path only occurs once.

The Data Warehouse Center does not support Unicode on Sybase servers.

Change in date format for Modified column

In the details view of the main Data Warehouse Center window, the format of the date in the **Modified** column has been updated. The date in the **Modified** column is displayed in the format for your locale and includes the time. This change in the date format ensures that sorting objects on the **Modified** column functions properly. This update applies to most lists of Data Warehouse Center objects that are shown in the Navigator and Details views, such as:

- Subjects
- Processes
- Warehouse schemas
- Warehouse agent sites
- Programs
- Steps
- Data resources
- Users

- User groups

Defining statistical transformers in the Data Warehouse Center

To perform a statistical transformation of your data, define the statistical transformer that you want to use.

Procedure:

To define statistical transformers:

1. Open the Process Model window.
2. Click the transformer icon and select a transformer from the list of available transformers.
3. Link the transformer that you selected to a warehouse source and warehouse target as required by the rules for that transformer.

Each transformer has specific rules for how it must be linked to a warehouse source and warehouse target. See the documentation for each transformer for more information.

Prerequisite for the iSeries warehouse agent

To use an iSeries warehouse agent for DB2 Warehouse Manager on V5R2 and V5R3 systems, the following PTF is required:

```
PTF SI13558
```

This database PTF enables the CLI on iSeries to handle Unicode data.

DB2 .NET Data Provider

DB2Connection.ConnectionString property

There is an additional keyword for the DB2Connection.ConnectionString property:

CurrentSchema
The schema to be used after a successful connection. Upon a successful connection, a SET CURRENT SCHEMA statement is sent to the DB2 server. This allows the application to name SQL objects without having to qualify them by a schema name.

Development Center

Restrictions to DB2 Universal type 2 and type 4 drivers

In Version 8.2, support was added to allow users to connect to a DB2 Universal Database (UDB) database from within the Development Center using the DB2 Universal Type 2 and Type 4 drivers. However, if you attempt to use one of these drivers to connect to an iSeries server, or to a DB2 UDB server that is Version 8.1 or earlier, you will see the following error message:

```
Connection to <database> failed.
IBM DB2 Universal driver (JCC) not found.
```

Refer to the topic titled ″JDBC drivers″ in the DB2 Information Center for additional information on which drivers to use in order to avoid this error.

GUI Tools

Modifying the statement termination character

Starting with DB2 Universal Database (UDB) Version 8.2 FixPak 1 (equivalent to Version 8.1 FixPak 8), you can modify the statement termination character within a script that is being run in the command line processor (CLP) or the Command Editor. This on the fly modification is similar to the method currently available in DB2 UDB for OS/390®. The following example shows how the termination character can be changed after each statement:

```
connect to gilroy user newton using password;
select * from newton.department;
—#SET TERMINATOR :
select * from newton.employee:
—#SET TERMINATOR @
select * from newton.department@
—#SET TERMINATOR ;
select * from newton.department;
—#SET TERMINATOR &
terminate&
```

The ability to change the termination character is important when a script contains compound statements. In the following example, DB2 UDB assumes that the first ; that it encounters in the compound CREATE TRIGGER statement is the termination character for the whole CREATE TRIGGER statement. However, this is not the case. It is only meant to be the termination character for one of the statements inside the compound CREATE TRIGGER statement.

```
CONNECT TO SAMPLE;
DROP TRIGGER newton.NWTTRIGGER;
CREATE TRIGGER newton.NWTTRIGGER AFTER DELETE
ON newton.NWTTABLE FOR EACH ROW MODE DB2SQL
BEGIN ATOMIC
   insert into newton.nwttable values(0,'0');
   insert into newton.nwttable values( -1, '-1');
END;
CONNECT RESET;
TERMINATE;
```

The following example shows how the statement termination character can be modified within the script to achieve the desired results.

```
CONNECT TO SAMPLE;
DROP TRIGGER newton.NWTTRIGGER;
—#SET TERMINATOR @
CREATE TRIGGER newton.NWTTRIGGER AFTER DELETE
ON newton.NWTTABLE FOR EACH ROW MODE DB2SQL
BEGIN ATOMIC
   insert into newton.nwttable values(0,'0');
   insert into newton.nwttable values( -1, '-1');
END@
—#SET TERMINATOR ;
CONNECT RESET;
```

If you do not need your scripts to run locally on DB2 for OS/390 or your DB2 UDB scripts do not connect to OS/390, then using —#SET TERMINATOR is not the recommended method for modifying statement termination characters. Instead you should use the existing –tdX or ;— options.

The –tdX option allows you to specify the termination character when calling a script using a CLP command. The 'X' represents the character being used as the statement termination character. For example, in the command:

```
db2 –tvf test.txt –td&
```

The & will be used as the statement termination character when running the script in the test.txt file. If that script contained the compound CREATE TRIGGER statement, it would be written as:

```
CONNECT TO SAMPLE&
DROP TRIGGER newton.NWTTRIGGER&
CREATE TRIGGER newton.NWTTRIGGER AFTER DELETE
ON newton.NWTTABLE FOR EACH ROW MODE DB2SQL
BEGIN ATOMIC
   insert into newton.nwttable values(0,'0');
   insert into newton.nwttable values( -1, '-1');
END&
CONNECT RESET&
TERMINATE&
```

Note: The –tdX option is not available in the Command Editor.

The script containing the compound CREATE TRIGGER statement can also be rewritten using the ;— option as follows:

```
CONNECT TO SAMPLE;
DROP TRIGGER newton.NWTTRIGGER;
CREATE TRIGGER newton.NWTTRIGGER AFTER DELETE
ON newton.NWTTABLE FOR EACH ROW MODE DB2SQL
BEGIN ATOMIC
   insert into newton.nwttable values(0,'0');—
   insert into newton.nwttable values( -1, '-1');—
END;
CONNECT RESET;
TERMINATE;
```

Database unavailable status in the database details pane of the Control Center

You can use the Control Center's details pane to view information about your databases. Selecting a database in the object tree or contents pane displays a summary of its state. In certain situations database information might be unavailable. Some reasons for this unavailability are described in the following table.

Table 23. Reasons for a database status of unavailable

Database status element	Possible reasons for unavailable status
Last backup	• No backups have been performed for the database. • User does not have the required authority to access this information.
Size	• Database is pre-Version 8.2. • User does not have the required authority to access this information.
Capacity	• Database is pre-Version 8.2. • Database has multiple partitions. • User does not have the required authority to access this information.

Table 23. Reasons for a database status of unavailable (continued)

Database status element	Possible reasons for unavailable status
Health	• Health monitor is not turned on. • Timing delay. There is approximately a 5 minute delay from the time a database is activated until its health status is available.
Maintenance	• Database is pre-Version 8.2.

Default write to table output generation (Create Event Monitor)

A **Generate** button has been added to the Output options dialog, which is launched from the Create Event Monitor window. Clicking the Generate button generates the default *write to table output* option. This output is equivalent to the syntax generated by the **db2evtbl** command.

The generated option shows the user which tables and data elements will be included when the event monitor is created. Users can modify the command to suit their needs.

The generated syntax is based on the event monitor name and event types specified in the Create Event Monitor window. Specify the event monitor name and event types before generating the output option syntax.

If the event monitor name or event types change after the generation of the output option, a message displays to remind the user to regenerate the output option before creating the event monitor. If the output option is not regenerated, event tables will be generated based on the event monitor name that was previously specified.

Installation and Configuration Supplement

Application server for DB2 UDB

The application server for DB2 Universal Database (UDB) no longer supports remote administration or stored procedures.

The updated topics are as follows:

Enabling the application server for DB2 UDB

Enabling the database does the following:

- Connects to a specified database
- Creates and populates metadata tables
- Updates the DBM CFG parameters, JDK_PATH, and JAVA_HEAP_SZ
- Installs the DB2 Web Services Application.

Prerequisite:

On Linux, after installing the application server and before enabling the application server, you must set up the Linux Java environment. Refer to your Application Development Guide: Building and Running Applications book for details on setting up the Linux Java environment.

Procedure:

To enable the application server for DB2 UDB, perform the following:

1. Log on to the DB2 server as **root** on Linux operating systems, or as a user with **Administrator** privileges on Windows operating systems.
2. For Linux–based operating systems run the following command:

   ```
   . /db2instance_path/sqllib/db2profile
   ```

   where *db2instance_path* is where the DB2 UDB instance was created.
3. Run one the following commands:
   - For Linux operating systems:

     ```
     AppServer_install_path/bin/enable.sh
           -db db_alias
           -user db_user
           -password db_password
           -db2path path_to_sqllib
           -instance instance_name
           -easpath path_to_eas
           -fencedid fenced_userid
     ```

   - For Windows operating systems:

     ```
     AppServer_install_path\bin\enable
           -db db_alias
           -user db_user
           -password db_password
           -db2path path_to_sqllib
           -instance instance_name
           -easpath path_to_eas
     ```

   where:
   - *db_alias* is the alias of the database to be enabled.
   - *db_user* is the user id to use when connecting to the database.
   - *db_password* is the password to use with the user id connecting to the database.
   - *path_to_sqllib* is the path to the DB2 UDB instance SQLLIB directory. This path is used to update DB2EAS with the required JAR files.
   - *instance_name* is the name of a DB2 UDB instance.
   - *path_to_eas* is the path to the embedded application server.
   - *fenced_userid* is the user id for the fenced user.

Once the application server for DB2 UDB is enabled, the application server is started automatically.

Starting the application server for DB2 UDB locally

The application server should be started using the fenced user ID for systems creating web services in a .NET environment or running XML Metadata Registry (XMR) only.

Starting the application server for DB2 UDB remotely

This section is removed. The application server for DB2 UDB no longer supports remote administration.

Stopping the application server for DB2 UDB locally

The application server should be stopped using the fenced user ID for systems creating web services in a .NET environment or running XML Metadata Registry (XMR) only.

Stopping the application server for DB2 UDB remotely

This section is removed. The application server for DB2 UDB no longer supports remote administration.

Uninstalling the application server for DB2 UDB

This section is removed. The application server for DB2 UDB no longer supports remote administration.

DB2 Embedded Application Server enablement

The enabled database of a DB2 Embedded Application Server must be located in a 32–bit instance Any databases accessed from the DB2 Embedded Application Server can be located in 32–bit or 64–bit instances.

Deploying DB2 Web Tools

Application servers using JDK 1.4 no longer require customization of the CLASSPATH variable during DB2 Web Tools deployment. All dependencies, including those for XML parser and transformer, are now deployed with the web module and are expected to be loaded from the `WEB-INF\lib` directory according to the J2EE specification. This change affects two information topics:

- Deploying DB2 Web Tools on WebLogic application servers
- Deploying DB2 Web Tools on other application servers

The updated topics are as follows:

Deploying DB2 Web Tools on WebLogic application servers

This task describes how to deploy and configure DB2 Web Tools (including the Web Command Center and the Web Health Center) on BEA WebLogic 7.0. These tools run as Web applications on a Web server to provide access to DB2 servers through Web browsers.

Prerequisites:

Before you install DB2 Web Tools on WebSphere, ensure that you have:

- BEA WebLogic 7.0 application server.
- IBM DB2 Administration Client Version 8.
- A Web browser that is compliant with HTML 4.0.

  Note: DB2 Web Tools were tested using Netscape 4.x, Netscape 6.x, Netscape 7.x, Mozilla 1.x, Internet Explorer 5.x, Opera 6.x, Konqueror 3.x (Linux) and EudoraWeb 2.x (Palm OS). Use of certain Web browsers that have not been tested might require an explicit reference to be added in the servlet configuration.

Restrictions:

The following restrictions apply to the DB2 Web Tools deployment:

- Multiple language and code page conversions between the middle tier and DB2 servers are not supported. Although the language for the server is what is used for display, some characters might appear incorrectly.
- In order to see the health alerts for databases, table spaces, and table space containers in the Web Health Center, you need to ensure that the databases are cataloged on the Web application server.

- The use of the Web browser buttons (**Stop**, **Back**, **History**) are not supported while using DB2 Web Tools.
- If you are using Netscape Navigator 4 with DB2 Web Tools, your browser display might not refresh properly . If you experience this problem, you can refresh your display by minimizing the window and then maximizing it. You can also refresh the display by hiding the browser window under another window, and then bringing it to the foreground again.
- To assign your own alias to any DB2 Universal Database (UDB) system, instance node, or database, you must explicitly catalog it on the application server using the DB2 Configuration Assistant or the DB2 Control Center.
- During their first startup, the DB2 Web Tools require a significantly longer time to initialize than for subsequent startups. Most of this wait is due to the automatic catalog process. If you do not want to use the automatic catalog functionality, you can shorten the time you must wait by turning the automatic cataloging off through the servlet configuration.

  Note: The servlet configuration parameters are available in the deployment descriptor file `web.xml`. The parameter names and default values are subject to change with each release. Some application servers might allow changes to these parameters, either through their interface or by editing the `web.xml` file directly.
- The output (results) buffer has an absolute maximum size of 1MB when using a desktop or laptop browser, even if it is configured for larger capacity. In the case of PDA Web browsers, the limit is 1KB.
- On Linux and Windows operating systems, the DB2 Web Tools automatically discover and catalog any systems on the same TCP/IP net as the application server. Systems on the same TCP/IP net have the same first three digits in their IP address. The tools try to catalog the DB2 UDB system nodes using the original remote TCP/IP host name. If there is name duplication, the tools assign a unique random name. You must explicitly catalog any other DB2 administration servers on the application server if you want them to be accessible. This includes any servers using TCP/IP that are not on the same TCP/IP net as the application server, as well as any servers that do not use TCP/IP.
- On Linux and Windows operating systems, the DB2 Web Tools attempt to automatically discover and catalog any DB2 UDB instance nodes and databases that reside on cataloged DB2 UDB systems. It is possible to configure a remote instance for multiple communication protocols, therefore, the catalog will contain a separate node entry for each protocol supported by an automatically cataloged instance. If there is name duplication, the tools assign a unique random name.

Procedure:

To install DB2 Web Tools on WebLogic application servers:

1. Deploy the DB2 Web Tools through the WebLogic administrative console by completing the following:
   a. Start the WebLogic administrative console.
   b. Click **domain** –> **deployments** –> **Web Applications** in the left pane of the window.
   c. Click the **Configure a new Web Application** link to install DB2 Web Tools Web application.
   d. Browse the listing of the file system to locate `Sqllib\tools\web\db2wa.war`.
   e. Click on **select** beside the `db2wa.war` file name.

f. Choose a server from the list of available servers to house DB2 Web Tools, select the server and click the arrow to move the server to target servers.

   Note: Preserving the original name **db2wa** is mandatory, as DB2 Web Tools has it hardcoded.

g. Click **Configure and Deploy**.

h. Wait until the application server refreshes the deployment status of the Web application on the selected server. If successful, it should show **Deployed=true**

2. Invoke the DB2 Web Tools Web application, which is located at:

   `http://server_name:app_server_port_number/db2wa`

   For example, `http://server_name:7001/db2wa`.

Deploying DB2 Web Tools on other application servers

This task describes how to deploy and configure DB2 Web Tools (including the Web Command Center and the Web Health Center) on other application servers such as Tomcat 4.0 and Macromedia JRun 4.0. These tools run as Web applications on a Web server to provide access to DB2 servers through Web browsers.

Prerequisites:

Before you install DB2 Web Tools, ensure that you have:

- An application server, such as:
  - Tomcat 4.0 Servlet/JSP Container (http://jakarta.apache.org/tomcat/)
  - Macromedia JRun 4.0
- IBM DB2 Administration Client Version 8.
- A Web browser that is compliant with HTML 4.0.

Restrictions:

The following restrictions apply to the DB2 Web Tools deployment:

- Multiple language and code page conversions between the middle tier and DB2 servers are not supported. Although the language for the server is what is displayed, some characters might appear incorrectly.
- In order to see the health alerts for databases, table spaces, and table space containers in the Web Health Center, you need to ensure that the databases are cataloged on the Web application server.
- The use of the Web browser buttons (**Stop**, **Back**, **History**) are not supported while using DB2 Web Tools.
- If you are using Netscape Navigator 4 with DB2 Web Tools, your browser display might not refresh properly. If you experience this problem, you can refresh your display by minimizing the window and then bringing it back. You can also refresh the display by hiding the browser window under another window, and then bringing it to the foreground again.
- To assign your own alias to any DB2 UDB system, instance node, or database, you must explicitly catalog it on the application server using the DB2 Configuration Assistant or the DB2 Control Center.
- During their first startup, the DB2 Web Tools require a significantly longer time to initialize than for subsequent startups. Most of this wait is due to the automatic catalog process. If you do not want to use the automatic catalog

functionality, you can shorten the time you must wait by turning the automatic cataloging off through the servlet configuration.

Note: The servlet configuration parameters are available in the deployment descriptor file `web.xml`. The parameter names and default values are subject to change with each release. Some application servers might allow changes to these parameters, either through their interface or by editing the `web.xml` file directly.

- The output (results) buffer has an absolute maximum size of 1MB when using a desktop or laptop browser, even if it is configured for more. In the case of PDA Web browsers, the limit is 1KB.
- On Linux and Windows operating systems, the DB2 Web Tools automatically discover and catalog any systems on the same TCP/IP net as the application server. Systems on the same TCP/IP net have the same first three digits in their IP address. The tools try to catalog the DB2 UDB system nodes using the original remote TCP/IP host name. If there is name duplication, the tools assign a unique random name. You must explicitly catalog any other DB2 administration servers on the application server if you want them to be accessible. This includes any servers using TCP/IP that are not on the same TCP/IP net as the application server, as well as any servers that do not use TCP/IP.
- On Linux and Windows operating systems, the DB2 Web Tools attempt to automatically discover and catalog any DB2 UDB instance nodes and databases that reside on cataloged DB2 UDB systems. It is possible to configure a remote instance for multiple communication protocols, therefore, the catalog will contain a separate node entry for each protocol supported by an automatically cataloged instance. If there is name duplication, the tools assign a unique random name.

Procedure:

The following are the procedures for installing DB2 Web Tools using application servers such as Tomcat 4.0 or Macromedia JRun 4.0:

Tomcat 4.0

1. Prepare the Tomcat 4.0 configuration file (CLASSPATH) by completing the following:
   a. Create a new environment/system variable **CATALINA_HOME** to contain the path (root directory) to Tomcat 4.0. For example, `D:\jakarta-tomcat-4.0.3`.
   b. Confirm that the Tomcat Servlet/JSP Container is functional:
      1) Start Tomcat by running **startup.bat** from Tomcat's `bin` directory.
      2) Access the main Web page **http://localhost:8080/** through a Web browser.
2. Deploy the DB2 Web Tools into the Tomcat Servlet/JSP Container by locating the DB2 Web Tools install path (i.e. `Sqllib\tools\web\db2wa.war`) and copying **db2wa.war** into Tomcat's deployment directory (i.e. Tomcat's webapps directory).
3. Invoke DB2 Web Tools on Tomcat Servlet/JSP Container by completing the following:
   a. Open a DB2 Command Window and change the directory to Tomcat's `bin` directory.
   b. Start Tomcat using **startup.bat** and confirm that a new directory (**db2wa**) has been added into the `webapps` directory.

Note: Running **startup.bat** from a command prompt window would not set DB2PATH. In order to enable DB2PATH being set, the CLASSPATH line needs to be changed to explicitly reference the DB2 UDB installation path rather than the %DB2PATH% environment variable.

c. The DB2 Web Tools enterprise application is located at **http://localhost:8080/db2wa** and can be accessed with an HTML 4.0 compliant Web browser.

JRun

1. Prepare a new application server for DB2 Web Tools by completing the following tasks:

   Recommendation:

   Creating a new application server is recommended, but not mandatory. For testing purposes, the default server can be used, and only the configuration of the JVM classpath and the deployment is required.

   a. Start the JRun Management Console and login as the administrator of the application server.
   b. Create a new application server using **Create New Server** located at the top right of the main page. Do not change the host name selection from `localhost`.
   c. Enter the new server name (**DB2WebToolsServer**) and click the JRun Server Directory. The value is automatically filled in.
   d. Click the **Create Server** button.
   e. Record the generated values or enter new values for:
      - JNDI Provider URL
      - Web Server Port Number. This would be the value to be used in the URL for the DB2 Web Tools (i.e. `http://localhost:`*`web_server_port_numer`*`/db2wa`)
      - Web Connector Proxy Port Number
   f. Click **update port numbers** if necessary and close the window.

2. Deploy DB2 Web Tools on the JRun application server by completing the following tasks:

   a. Start the application server selected to host DB2 Web Tools Web application (DB2WebToolsServer, default or any other except admin).
   b. Click **Web Applications** and then click **Add**.
   c. Browse the **Deployment File** section to select the `Sqllib\tools\web\db2wa.war` file in the DB2 UDB installation path.
   d. Click **Deploy** and confirm that the context path is `/db2wa`.
   e. Select the application server and confirm that the DB2 Web Tools application appears in the **Web Applications** section. Do *not* click **Apply** on this page.
   f. Select the **Home** link from the upper left panel of the main page.
   g. Restart the application server from the **Home** view that contains the DB2 Web Tools (**DB2WebToolsServer**).

The DB2 Web Tools enterprise application is located at http://localhost:*your_web_server_port_numer*/db2wa and can be accessed with an HTML 4.0 compliant Web browser.

Direct I/O on block devices (Linux)

Direct I/O is now supported on both file systems and block devices for distributions of Linux with a 2.6 kernel. Direct I/O on block devices is an alternative way to specify device containers for direct disk access or for raw I/O. The performance of Direct I/O is equivalent to the raw character device method. DB2 Universal Database (UDB) enables Direct I/O while opening the table space when the CREATE TABLESPACE statement specifies a block device name for the container path. Previously, the same performance was achieved using the raw I/O method, which required binding the block device to a character device using the **raw** utility.

Table 24. Comparison of direct I/O and raw I/O

Direct I/O (new method)	Raw I/O (old method)

`CREATE TABLESPACE dms1`
`MANAGED BY DATABASE`
`USING (DEVICE '/dev/sda5' 11170736)`
	`CREATE TABLESPACE dms1`
	`MANAGED BY DATABASE`
	`USING (DEVICE '/dev/raw/raw1' 11170736)`

Although the raw I/O method is still supported by DB2 UDB, it has been deprecated and support for it might be removed from future kernels.

Recommendation:

If you want to exploit direct disk access, create your DMS device containers using Direct I/O to avoid future migration issues.

Note: Direct I/O is not supported by DB2 UDB on Linux/390.

DB2 Information Center daemon (Linux)

The DB2 Information Center daemon is responsible for the control of the DB2 documentation server. The daemon, which is part of the DB2 Information Center installation, is composed of two files:

- db2icd – the initialization script
- db2ic.conf – the configuration file

These files are installed in the following locations:

AIX /etc/db2icd

/var/db2/v81/db2ic.conf

HP /sbin/init.d/db2icd

/var/opt/db2/v81/db2ic.conf

Solaris Operating Environment
/etc/init.d/db2icd

/var/db2/v81/db2ic.conf

Linux /etc/init.d/db2icd

/var/db2/v81/db2ic.conf

Starting or stopping the Information Center daemon (AIX, Solaris Operating Environment, HP, Linux)

The only time you should need to start or stop the daemon manually is when you want to change the configuration variables for the daemon. Normally, the daemon

is started at system startup, according to the run levels created during the installation of the DB2 Information Center.

Procedure:

To stop and start the Information Center daemon:

1. Halt the daemon if it is already running. At a command line, enter:
   INIT_DIR/db2icd stop

   where *INIT_DIR* is the installation directory of the db2icd file listed previously.
2. Change any of the variables for the daemon by editing the db2ic.conf file. Currently, you can modify the TCP port number where the documentation is available, and the location of the temporary workspace used by the daemon while it is running.
3. Start the daemon. At a command line, enter:
   INIT_DIR/db2icd start

   where *INIT_DIR* is the installation directory of the db2icd file listed previously.

When the daemon starts, it uses the new environment variables.

There is also an option to shut down and restart the daemon immediately. At a command line, enter:

INIT_DIR/db2icd restart

where *INIT_DIR* is the installation directory of the db2icd file listed previously.

You can check the status of the daemon at any time. At a command line, enter:

INIT_DIR/db2icd status

where *INIT_DIR* is the installation directory of the db2icd file listed previously. The daemon returns the current status, and displays the ID of the daemon process or processes if it is active.

Response file keywords and sample files

To install the DB2 Information Center Version 8.2 using a response file, use the following information:

Response file keyword (PROD)
INFORMATION_CENTER

Response file name
db2doce.rsp

Response file installation error codes

The following error code is for Windows only and is not applicable to Linux operating systems.

3010 The installation is successful, however a system restart is required to complete the installation.

Required user accounts for installation of DB2 servers (Windows)

Increase quotas

The *Increase quotas* user right has been changed to *Adjust memory quotas for a process* on the Windows XP and Windows Server 2003 operating systems.

User rights granted by the DB2 installer - Debug Programs

The DB2 installation program does not grant the Debug Programs user right. The DB2 installer grants the following user rights:

- Act as part of the operating system
- Create token object
- Lock pages in memory
- Log on as a service
- Increase quotas
- Replace a process level token

Asynchronous I/O support (Linux)

Asynchronous I/O (AIO) support is now available on Linux (2.6 and some 2.4 kernels) for raw devices and O_DIRECT file systems. AIO improves page cleaner performance. You can enable or disable AIO on Linux by issuing the **db2set** command.

To use AIO, users must install libaio-0.3.98 or later and have a kernel that supports AIO. Users must also run the **db2set DB2LINUXAIO=true** command and restart DB2 Universal Database.

db2ln command modified to create DB2 UDB 64-bit library links (Linux)

In previous levels of DB2 Universal Database (UDB) Version 8, the **db2ln** command created certain DB2 links under /usr/lib and /usr/include. On platforms where both 32-bit and 64-bit DB2 UDB instances are supported, these links point to library files or include files under DB2DIR/lib64 or DB2DIR/include64 by default, where DB2DIR is the directory where DB2 UDB Version 8 is installed. If the default is not desired, you can specify the bit width by running the **db2ln** command with the -w flag:

```
db2ln -w 32|64
```

This prevents DB2 UDB 32-bit instances from coexisting with 64-bit instances on some platforms.

Starting with DB2 UDB Version 8.2, the **db2ln** command creates DB2 64-bit library links on those platforms in appropriate directories. In this case, the -w flag is only used for populating /usr/include. When the **db2ln** command creates the links for DB2 UDB library files, both 32-bit and 64-bit links are created on supported platforms. This allows both 32-bit instances and 64-bit instances to exist and run at the same time.

On some Linux distributions, the libc RPM command comes with the /usr/lib/libdb2.so or /usr/lib64/libdb2.so library. This library is used for Sleepycat Software's Berkeley DB implementation and is not associated with IBM's DB2 UDB. However, this file prevents the **db2ln** command and the **db2rmln** command from functioning. The **db2ln** command does not overwrite the file and

the **db2rmln** command does not remove the file. In this case, to compile applications using DB2 UDB, compiling and linking processes need to provide a full path to DB2 UDB's headers and libraries, respectively. This is the recommended method because it allows compiling and linking against multiple releases of DB2 UDB on the same computer.

See the DB2 UDB Version 8.2 *Installation and Configuration* manual for details about restrictions using the **db2ln** command.

Message Reference

ADM message updates

ADM12504E

The description for message ADM12504E incorrectly implies that the instance name on the primary database should match the instance name on the standby database.

ADM12504E

Unable to establish HADR primary-standby connection because the DB2 instance names do not match. Correct the HADR_REMOTE_INST configuration parameter.

It is not a HADR requirement for the instance name on the primary to match the instance name on the standby.

CLP message updates

DB21015E

The Command Line Processor backend process request queue or input queue was not created within the timeout period.

Explanation

Either the values for the DB2BQTRY and DB2BQTIME environment variables need to be increased or the command line processor back-end program ″db2bp″ cannot be started. The ″db2bp″ program must reside in the correct database manager install path and users must have execute permission on the file.

On Linux platforms, ensure that the file system has enough file blocks and inodes.

User Response

Correct the error and resubmit the command.

DBI message updates

DBI1060E

Invalid package name <″pkg-name″>

Explanation:

An incorrect name has been entered. The package either does not exist or the name has been entered incorrectly.

User Response:

Check to see if the name of the given package exists on the distribution media. If so, examine the name to see if it has been misspelled. All package names should be in lowercase.

SQL message additions and updates

SQL0121N

The target name ″<name>″ is specified more than once for assignment in the same SQL statement.

Explanation

The same target name ″<name>″ is specified more than once as an OUT or INOUT argument of a CALL statement, or in the list of columns of an INSERT statement, the left hand side of assignments in the SET clause of an UPDATE statement, or the left hand side of the assignment statement. The target name identifies a column, SQL parameter, SQL variable, or new transition variable.

Note that this error may occur when updating or inserting into a view where more than one column of the view is based on the same column of a base table.

The statement cannot be processed.

User Response

Correct the syntax of the statement so each name is specified only once.

sqlcode : -121

SQL0270N

Function not supported (Reason code = "<reason-code>")

Explanation

74 Updating a status field in the recovery history file by timestamp is not allowed.

User Response

74 Update the status field in the recovery history file by EID only.

Explanation

75 Automatic statistics collection is not supported on a multiple database partition system, on a system where SMP is enabled, or on a federated system.

User Response

75 Disable automatic statistics collection for this database by setting the database configuration parameters AUTO_STATS_PROF and AUTO_PROF_UPD to OFF.

Or change the system to one that is on a single database partition, where SMP is not enabled, and that is not federated.

SQL0494W

The number of result sets is greater than the number of locators.

Explanation

The number of result set locators specified on the ASSOCIATE LOCATORS statement is less than the number of result sets returned by the stored procedure. The first ″n″ result set locator values are returned, where ″n″ is the number of result set locator variables specified on the SQL statement.

The SQL statement is successful.

User Response

Increase the number of result set locator variables specified on the SQL statement.

sqlcode : +494

sqlstate : 01614

SQL1227N

Explanation

Reason code 4

Buffer size entries of PAGE_FETCH_PAIRS must be ascending in value. Also, for 32-bit instances, any buffer size value in a PAGE_FETCH_PAIRS entry cannot be greater than the number of pages in the table or 524287, whichever is lower. For 64-bit instances, any buffer size value in a PAGE_FETCH_PAIRS entry cannot be greater than the number of pages in the table or 2147483674.

SQL1271W

SQL1271W Database "<name>" is recovered but one or more table spaces are offline on node(s) "<node-list>".

Explanation

This message can occur after crash recovery, database level roll-forward recovery, or table space level roll-forward recovery. For database level roll-forward recovery, the database is available for use if the STOP option was specified. One or more table spaces on the specified nodes are not available. This situation could occur under any of the following situations:

- One or more table spaces receive I/O errors during recovery.
- One or more table spaces may require storage definition following recovery.
- One or more table spaces are unusable and must be dropped.
- One or more table spaces have been taken offline.
- Additional table spaces are restored while a table space level roll-forward recovery is already in progress.
- The database is rolled forward past a previous point-in-time table space roll-forward recovery, causing all table spaces involved to be placed in restore-pending state.

The table spaces that are offline can be identified on the specified node or nodes by the LIST TABLESPACES command or the **db2dart** utility. The administration notification log will provide more information on specific table spaces.

If ",..." is displayed at the end of the node list, see the administration notification log for the complete list of nodes.

Note: If you are using a partitioned database server, the node number or numbers indicate which node or nodes the error occurred on. Otherwise, the node number is not pertinent and should be ignored.

User Response

Repair or restore the table spaces if necessary and run roll-forward recovery. If the same error occurs, try running table space level roll-forward recovery offline.

SQL1768N

Unable to start HADR. Reason code = "<reason-code>"

Explanation:

9 The database is configured to use raw logs.

User Response:

9 Reconfigure the database to use only file system storage for the log files, instead of using raw i/o (direct disk access) devices. Refer to the discussion of the *logpath* and *newlogpath* database configuration parameters in *Administration Guide: Performance*.

SQL1790W

A default table space could not be found with a page size of at least ″<page size>″.

Explanation

The procedure NNSTAT was not able to create the SYSPROC.FED_STATS table to keep a history of the statement that was run. The table space with sufficient page size (at least ″<pagesize>″) could not be found.

User Response

Ensure that a table space with a page size of at least ″<pagesize>″ exists.

sqlcode: +1790

sqlstate: 01670

SQL1791N

The specified server definition, schema, or nickname object-name does not exist.

Explanation

The procedure NNSTAT accepts a server definition, schema, and nickname as input and one or more of these objects, including object-name, could not be found.

User Response

Specify an existing server definition, schema, or nickname and resubmit the statement.

sqlcode: -1791

sqlstate: 42704

SQL2316W

The Runstats command string for the statistics profile has exceeded the maximum size. The Runstats command string will be truncated to the maximum size and stored in the catalog table SYSIBM.SYSTABLE.

Explanation

The maximum size of the STATISTICS_PROFILE column is 32768 bytes. If the size of the Runstats command string for the statistics profile is greater than this amount, the Runstats command string is truncated to the maximum size.

The utility continues processing.

User Response

Refer to the STATISTICS_PROFILE column in the catalog table SYSIBM.SYSTABLES to review the statistics profile. If the existing statistics profile is not desired, reissue the RUNSTATS utility and specify the UPDATE PROFILE or the UPDATE PROFILE ONLY option to modify the profile. Refer to the RUNSTATS utility documentation for information about the utility options.

SQL3705N

The buffer size parameter specified is not valid. The buffer size must be specified as 0 or be between 8 and 250000 inclusive. For multiple buffers, the total buffer size must not exceed 250000.

SQL5099N

There is a new reason code for SQL5099N:

17 - Cannot disable log archiving when database is in roll forward pending state.

SQL20290N

The SQL statement references routine ″<routine-name>″ (specific name ″<specific-name>″) which cannot be run on partition ″<partition-number>″.

Explanation

The routine ″<routine-name>″ (specific name ″<specific-name>″) was called with an invalid partition number ″<partition-number>″.

User Response

Specify -1 for the partition number parameter to run the routine on the current partition.

sqlcode: -20290

sqlstate: 560CA

SQL22025N

An invalid input argument was specified for the reorgchk stored procedure.

Explanation

Only 'T' and 'S' are supported for the first argument of the stored procedure. If 'T' is specified as the first argument, the second argument to the stored procedure must be a fully qualified table name as <schema.table-name>.

User Response

Resubmit with valid input arguments.

SQL27994W

The length of a special register default column is smaller than the target column length. Load may truncate the values corresponding to this column (<column-num>).

Explanation

The clause SESSION_USER, CURRENT_USER, SYSTEM_USER, or CURRENT_SCHEMA has been specified for column ″<column-num>″, but this column was either defined with a length that is less than 128 bytes (SESSION_USER only) or the column-value inserted into this column may exceed the target length after code page conversion. Load may truncate the column-value.

User Response

If your system standards would not allow a user ID (SESSION_USER) to exceed the length of the column, then this warning may be ignored. To prevent this warning from occurring, the length of the column must be at least 128 bytes. If code page conversion resulted in the growth of this register default column, increase the length of the column to accommodate the growth.

Quick Beginnings

Verifying that your databases are ready for migration

The following prerequisites section is documented in the 8.2 version of the topic that explains how to verify that your databases are ready for migration:

Prerequisites

Ensure that the migration.log file, found in the instance owner's home directory, contains the following text: Version of DB2CKMIG being run: VERSION 8.

This prerequisite is in fact a post-migration step performed at the end of the procedure.

Common Criteria certification for DB2 UDB

The authoritative information for DB2 UDB configurations that have been certified for Common Criteria can be found at http://niap.nist.gov/cc-scheme

Spatial Extender

Verifying the Spatial Extender installation

The runGseDemo sample program can be used to become familiar with application programming for DB2 Spatial Extender. For a description of the steps that the sample program takes to create a spatially-enabled database and perform spatial analysis on data in that database, refer to the topic titled "The DB2 Spatial Extender sample program". This topic is in the Information Center and the *Spatial Extender and Geodetic Extender User's Guide and Reference*.

DB2 Spatial Extender provides another sample program, seBankDemoRunBankDemo, that demonstrates how to add spatial capabilities to an existing information system.

For more information about both sample programs, see the README files in the following directories:

Windows

```
~\sqllib\samples\spatial
~\sqllib\samples\spatial\bank
```

Linux

```
~/sqllib/spatial
~/sqllib/spatial/bank
```

SQL Administrative Routines

EXEC_DB2_SCRIPT procedure unavailable

The documentation currently includes the description of a procedure named EXEC_DB2_SCRIPT. Because this procedure is not part of the product, its description should be disregarded.

GET_DB_CONFIG is a stored procedure

GET_DB_CONFIG is a stored procedure, not a table function. The documentation for GET_DB_CONFIG incorrectly identifies this procedure as a table function. The following text is the correct description.

The schema is SYSPROC.

The GET_DB_CONFIG procedure returns database configuration information. The procedure does not take any arguments. It returns a single result set with two rows containing a column for each parameter. The row with a value of 0 in the DBCONFIG_TYPE column contains the values of the database configuration parameters that are stored on disk. The row with a value of 1 in the DBCONFIG_TYPE column contains the current values of the database configuration parameters that are stored in memory.

This procedure requires a user temporary table space that is used to create a global temporary table to store the result set.

Column expansion for snapshot UDFs

The size of some VARCHAR and CLOB columns in snapshot UDFs have changed to match the maximum database object name standards.

Table 25. Expanded columns for UDFs

UDF		
Column name	**Existing column definition**	**Updated column definition**
SYSFUN.SQLCACHE_SNAPSHOT		
DB_NAME	VARCHAR(8)	VARCHAR(128)
STMT_TEXT	CLOB(64K)	CLOB(16M)
SNAPSHOT_APPL_INFO		
APPL_NAME	VARCHAR(255)	VARCHAR(256)
APPL_ID	VARCHAR(32)	VARCHAR(128)
AUTH_ID	VARCHAR(30)	VARCHAR(128)
CLIENT_NNAME	VARCHAR(20)	VARCHAR(128)
CLIENT_PRDID	VARCHAR(20)	VARCHAR(128)
INPUT_DB_ALIAS	VARCHAR(20)	VARCHAR(128)
CLIENT_DB_ALIAS	VARCHAR(20)	VARCHAR(128)
DB_NAME	VARCHAR(8)	VARCHAR(128)
DB_PATH	VARCHAR(256)	VARCHAR(1024)
EXECUTION_ID	VARCHAR(20)	VARCHAR(128)
CORR_TOKEN	VARCHAR(32)	VARCHAR(128)
TPMON_CLIENT_USERID	VARCHAR(20)	VARCHAR(256)
TPMON_CLIENT_WKSTN	VARCHAR(20)	VARCHAR(256)
TPMON_CLIENT_APP	VARCHAR(20)	VARCHAR(256)
TPMON_ACC_STR	VARCHAR(100)	VARCHAR(200)
SNAPSHOT_STATEMENT		
CURSOR_NAME	VARCHAR(31)	VARCHAR(128)

Table 25. Expanded columns for UDFs (continued)

UDF		
Column name	**Existing column definition**	**Updated column definition**
CREATOR	VARCHAR(32)	VARCHAR(128)
PACKAGE_NAME	VARCHAR(20)	VARCHAR(128)
STMT_TEXT	CLOB(64K)	CLOB(16M)
SNAPSHOT_LOCKWAIT		
TABLE_NAME	VARCHAR(35)	VARCHAR(128)
TABLE_SCHEMA	VARCHAR(32)	VARCHAR(128)
APPL_ID_HOLDING_LK	VARCHAR(32)	VARCHAR(128)
SNAPSHOT_DATABASE		
INPUT_DB_ALIAS	VARCHAR(8)	VARCHAR(128)
DB_NAME	VARCHAR(8)	VARCHAR(128)
DB_PATH	VARCHAR(256)	VARCHAR(1024)
SNAPSHOT_BP		
BP_NAME	VARCHAR(20)	VARCHAR(128)
DB_NAME	VARCHAR(8)	VARCHAR(128)
INPUT_DB_ALIAS	VARCHAR(8)	VARCHAR(128)
DB_PATH	VARCHAR(256)	VARCHAR(1024)
SNAPSHOT_LOCK		
TABLE_NAME	VARCHAR(35)	VARCHAR(128)
TABLE_SCHEMA	VARCHAR(32)	VARCHAR(128)
SNAPSHOT_TABLE		
TABLE_NAME	VARCHAR(35)	VARCHAR(128)
TABLE_SCHEMA	VARCHAR(32)	VARCHAR(128)
SNAPSHOT_TBREORG		
TABLE_NAME	VARCHAR(35)	VARCHAR(128)
TABLE_SCHEMA	VARCHAR(32)	VARCHAR(128)
SNAPSHOT_SUBSECT		
STMT_TEXT	CLOB(64K)	CLOB(16M)
SNAPSHOT_DYN_SQL		
STMT_TEXT	CLOB(64K)	CLOB(16M)
HEALTH_DBM_INFO		
SERVER_INSTANCE_NAME	VARCHAR(8)	VARCHAR(128)
HEALTH_DBM_HI		
SERVER_INSTANCE_NAME	VARCHAR(8)	VARCHAR(128)
HEALTH_DBM_HI_HIS		
SERVER_INSTANCE_NAME	VARCHAR(8)	VARCHAR(128)
HEALTH_DB_INFO		
DB_NAME	VARCHAR(8)	VARCHAR(128)
INPUT_DB_ALIAS	VARCHAR(8)	VARCHAR(128)

Table 25. Expanded columns for UDFs (continued)

UDF		
Column name	**Existing column definition**	**Updated column definition**
DB_PATH	VARCHAR(256)	VARCHAR(1024)
HEALTH_DB_HI		
DB_NAME	VARCHAR(8)	VARCHAR(128)
HEALTH_DB_HI_HIS		
DB_NAME	VARCHAR(8)	VARCHAR(128)
HEALTH_DB_HIC		
DB_NAME	VARCHAR(8)	VARCHAR(128)
HEALTH_DB_HIC_HIS		
DB_NAME	VARCHAR(8)	VARCHAR(128)

Note: The definition of STMT_TEXT for the SNAPSHOT_DYN_SQL, SNAPSHOT_STATEMENT, SNAPSHOT_SUBSECT, and SQLCACHE_SNAPSHOT routines has expanded to CLOB(16M) to allow for future expansion only. Actual output of the statement text data will continue to be truncated at 64K.

SQL Reference

CONNECT statement

When connecting to Windows Server explicitly, the TO *authorization-name* or the USER *host-variable* can be specified using the Microsoft Windows NT Security Account Manager (SAM)-compatible name. The qualifier must be a NetBIOS style name, which has a maximum length of 15 characters. For example, 'Domain\User'.

Schema used by the Explain facility

The Explain facility uses the following IDs as the schema to use in qualifying the explain tables to populate:

- The session authorization ID for dynamic SQL
- The statement authorization ID for static SQL

The schema can be associated with a set of explain tables, or aliases that point to a set of explain tables under a different schema.

If no explain tables are found under the schema, the Explain facility checks for explain tables under the SYSTOOLS schema and attempts to use them.

String representations of datetime values

Time strings:

A string representation of a time is a string that starts with a digit and has a length of at least 4 characters. Trailing blanks may be included; a leading zero may be omitted from the hour part of the time, and seconds may be omitted entirely. If seconds are omitted, an implicit specification of 0 seconds is assumed. Thus, 13:30 is equivalent to 13:30:00.

Valid string formats for times are listed in the following table. Each format is identified by name and associated abbreviation.

Table 26. Formats for String Representations of Times

Format Name	Abbreviation	Time Format	Example
International Standards Organization	ISO	hh.mm.ss	13.30.05
IBM USA standard	USA	hh:mm AM or PM	1:30 PM
IBM European standard	EUR	hh.mm.ss	13.30.05
Japanese Industrial Standard Christian Era	JIS	hh:mm:ss	13:30:05
Site-defined	LOC	Depends on the territory code of the application	–

Starting with Version 8.2, "AM" and "PM" can be represented in lowercase or uppercase.

System monitor

Decoupled applications list without connection concentrator enabled

It is possible to see decoupled applications when issuing the **list applications** command even without connection concentrator enabled.

Progress monitoring of the run time rollback process

Progress monitoring of run time rollback provides progress information of rollback events using application snapshots. Rollback events are of two types:

Unit of work rollback
Includes explicit (user invoked) and implicit (forced) rollback of the entire transaction.

Savepoint rollback
Includes statement and application level savepoints. Nested savepoints are considered a single unit, using the outermost savepoint.

The information provided is the start time of the rollback event, the total work to be done, and completed work. The work metric is bytes.

Total Work units is the range in the log stream that needs to be rolled back for the transaction or savepoint.

Completed Work units shows the relative position in the log stream that has been rolled back.

Updates to Completed Work are made after every log record is processed. Updates are not performed evenly because log records vary in size.

Sample output from GET SNAPSHOT FOR ALL APPLICATIONS command:

```
            Application Snapshot

Application handle         = 6
Application status         = Rollback Active
      Start Time           = 02/20/2004 12:49:27.713720
      Completed Work       = 1024000 bytes
      Total Work           = 4084000 bytes

            Application Snapshot

Application handle         = 10
Application status         = Rollback to Savepoint
      Start Time           = 02/20/2004 12:49:32.832410
      Completed Work       = 102400 bytes
      Total Work           = 2048000 bytes
```

Note: If rollback is not active during a snapshot, then rollback elements will not be displayed.

XML Extender

Decomposing documents larger than 1 MB

You do not need to drop and recreate the stored procedure `dxxShredXML` to decompose documents that are larger than 1 MB. To decompose documents that are larger than 1 MB, invoke the stored procedure `dxxShredXML100MB`, which can shred documents up to 100 MB. Although `dxxShredXML100MB` can handle large documents, you might need to increase other resources for the stored procedure to complete successfully. To invoke the stored procedure through the sample program `dxxshrd`, use the new flag "`-large`". For example:

```
dxxshrd -large mydb xxx.xml
```

If your version of DB2 Universal Database is earlier than Version 8 FixPak 6, you must run `dxxMigv` to migrate XML Extender to the current level to run the new stored procedure.

Configuring MQ XML UDFs with XML Extender

You must configure and enable MQ XML user–defined functions (UDFs) before you can use them.

Prerequisites:

Install the UDFs by following the procedure in the ″Installing DB2 WebSphere MQ functions″ topic, which is found either in the Information Center or the *IBM DB2 Information Integrator Application Developer's Guide*.

Procedure:

To configure and enable MQ XML UDFs with XML Extender:

1. Open a DB2 command prompt window.
2. Connect to the database that will use the MQ XML UDFs by entering the following command:

   ```
   db2 connect to <database>
   ```
3. Change to the `bnd` directory within the path where you installed DB2 Universal Database, such as:
   - SQLLIB/bnd (Linux)

- C:\Program Files\IBM\SQLLIB\bnd (Windows)

4. Bind the database to XML Extender using the following command:
   ```
   db2 bind @dbxxbind.lst
   ```
5. Bind the database for XML Extender to use the MQ XML UDFs using the following command:
   ```
   db2 bind mqxml.bnd
   ```
6. Bind the database to the CLI using the following command:
   ```
   db2 bind @db2cli.lst
   ```

XML Extender environment variable DB2DXX_MIN_TMPFILE_SIZE

DB2 XML Extender can place large documents in temporary files to avoid using a great amount of memory during processing. On systems with large amounts of physical memory, you can avoid moving documents to temporary files, reducing the amount of Input/Output activity. The environment variable DB2DXX_MIN_TMPFILE_SIZE instructs XML Extender to use memory buffers, instead of temporary files, for processing documents smaller than the specified value. The variable is applicable only on the server. If multiple physical nodes participate in a partitioned environment, the variable can be set differently on each node, accurately reflecting the amount of memory installed on each computer. If the environment variable is not set, documents larger than 128 KB will automatically be placed into temporary files during processing. Documents smaller than 128 KB will be processed in memory.

DB2XML.XMLVarchar UDT redefinition

You can redefine the user-defined type (UDT) DB2XML.XMLVarchar up to 32 KB. To change the size of an XMLVarchar UDT, create the UDT before you enable the database for XML Extender.

For more information, see *DB2 XML Extender Administration and Programming*.

Appendix A. DB2 UDB FixPak CD directory structure

Windows operating systems

The files on the FixPak CD are located as follows:

Table 27. Windows files

Files	Location
DB2 product files:	x:\db2
Installation Prerequisites:	x:\doc\<language>\install.txt
Installation Prerequisites (HTML):	x:\doc\<language>\install.htm
License files:	x:\db2\license
Release Notes:	x:\doc\<language>\release.txt
Release Notes (HTML):	x:\doc\<language>\db2ir\index.htm

where:

- `x:` refers to your CD drive
- `<language>` refers to the language directory, consisting of a five-character code that corresponds to one of the languages in Table 29

UNIX® operating systems

The files on the FixPak CD are located as follows

Table 28. UNIX files

Files	Location
DB2 product files:	/cdrom/db2
Installation Prerequisites:	/cdrom/doc/<language>/install.txt
Installation Prerequisites (HTML):	/cdrom/doc/<language>/install.htm
License files:	/cdrom/db2/license
Release Notes:	/cdrom/doc/<language>/release.txt
Release Notes (HTML):	/cdrom/doc/<language>/db2ir/index.htm

where:

- `/cdrom` refers to your mount point
- `<language>` refers to the language directory, consisting of a five-character code that corresponds to one of the languages in Table 29

The following table outlines the language directory names and their corresponding languages.

Table 29. Directory names and their corresponding languages

Directory	Language
ar_AA	Arabic

© Copyright IBM Corp. 2002 - 2004

Table 29. Directory names and their corresponding languages (continued)

Directory	Language
bg_BG	Bulgarian
cs_CZ	Czech
da_DK	Danish
de_DE	German
el_GR	Greek
en_US	English
es_ES	Spanish
fi_FI	Finnish
fr_FR	French
hr_HR	Croatian
hu_HU	Hungarian
it_IT	Italian
iw_IL	Hebrew
ja_JP	Japanese
ko_KR	Korean
nl_NL	Dutch
no_NO	Norwegian
pl_PL	Polish
pt_BR	Brazilian Portuguese
pt_PT	Portuguese
ro_RO	Romanian
ru_RU	Russian
sk_SK	Slovak
sl_SI	Slovenian
sv_SE	Swedish
tr_TR	Turkish
zh_CN	Simplified Chinese
zh_TW	Traditional Chinese

Notes:

1. The directory names might appear in uppercase or lowercase letters, depending on your operating system.
2. All of the directories in the preceding list might not appear on this CD because all language directories are not available on all CDs.
3. Starting with Version 8.2, the Installation Notes are a section in the Release Notes.

Appendix B. Contacting IBM

In the United States, call one of the following numbers to contact IBM:

- 1-800-IBM-SERV (1-800-426-7378) for customer service
- 1-888-426-4343 to learn about available service options
- 1-800-IBM-4YOU (426-4968) for DB2 marketing and sales

In Canada, call one of the following numbers to contact IBM:

- 1-800-IBM-SERV (1-800-426-7378) for customer service
- 1-800-465-9600 to learn about available service options
- 1-800-IBM-4YOU (1-800-426-4968) for DB2 marketing and sales

To locate an IBM office in your country or region, check IBM's Directory of Worldwide Contacts on the web at http://www.ibm.com/planetwide

Product information

Information regarding DB2 Universal Database products is available by telephone or by the World Wide Web at http://www.ibm.com/software/data/db2/udb

This site contains the latest information on the technical library, ordering books, product downloads, newsgroups, FixPaks, news, and links to web resources.

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

- 1-800-IBM-CALL (1-800-426-2255) to order products or to obtain general information.
- 1-800-879-2755 to order publications.

For information on how to contact IBM outside of the United States, go to the IBM Worldwide page at www.ibm.com/planetwide

© Copyright IBM Corp. 2002 - 2004

Appendix C. Notices

IBM may not offer the products, services, or features discussed in this document in all countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

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

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

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country/region or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country/region where such provisions are inconsistent with local law:
INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions; therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product, and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

© Copyright IBM Corp. 2002 - 2004

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 that has been exchanged, should contact:

IBM Canada Limited
Office of the Lab Director
8200 Warden Avenue
Markham, Ontario
L6G 1C7
CANADA

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

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems, and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements, or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

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

COPYRIGHT LICENSE:

This information may contain sample application programs, in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Each copy or any portion of these sample programs or any derivative work must include a copyright notice as follows:

© (*your company name*) (*year*). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. *_enter the year or years_*. All rights reserved.

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both, and have been used in at least one of the documents in the DB2 UDB documentation library.

ACF/VTAM
AISPO
AIX
AIXwindows
AnyNet
APPN
AS/400
BookManager
C Set++
C/370
CICS
Database 2
DataHub
DataJoiner
DataPropagator
DataRefresher
DB2
DB2 Connect
DB2 Extenders
DB2 OLAP Server
DB2 Information Integrator
DB2 Query Patroller
DB2 Universal Database
Distributed Relational Database Architecture
DRDA
eServer
Extended Services
FFST
First Failure Support Technology
IBM
IMS
IMS/ESA
iSeries
LAN Distance
MVS
MVS/ESA
MVS/XA
Net.Data
NetView
OS/390
OS/400
PowerPC
pSeries
QBIC
QMF
RACF
RISC System/6000
RS/6000
S/370
SP
SQL/400
SQL/DS
System/370
System/390
SystemView
Tivoli
VisualAge
VM/ESA
VSE/ESA
VTAM
WebExplorer
WebSphere
WIN-OS/2
z/OS
zSeries

The following terms are trademarks or registered trademarks of other companies and have been used in at least one of the documents in the DB2 UDB documentation library:

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

Intel and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Sleepycat and the names of Sleepycat Software products referenced herein are trademarks, registered trademarks or service marks of Sleepycat Software, Inc.

Other company, product, or service names may be trademarks or service marks of others.

Printed in USA