IBM DB2 Universal Database

Release Notes

Version 8.2 FixPak 4 (equivalent to Version 8.1 FixPak 11) and subsequent FixPaks

IBM DB2 Universal Database

Release Notes

Version 8.2 FixPak 4 (equivalent to Version 8.1 FixPak 11) and subsequent FixPaks

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.

Contents

About the Release Notes

Content:

The release notes contain late-breaking 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™
```

The topics in the DB2 Information Center (http://publib.boulder.ibm.com/infocenter/db2help/) are updated regularly, but not necessarily for every release. The release notes ship with every release so it is possible that critical information is communicated in the release notes. However, critical information is contained in the release notes when:

- The topics in the DB2 Information Center are not updated for the corresponding release
- The critical information is discovered late in the development cycle, after the DB2 Information Center has been updated

Your primary source for information should be the DB2 Information Center. The release notes should be reviewed for late-breaking information that might apply to the particular release of your DB2 product. The FixPak readme should also be reviewed for specific considerations involving the installation of your DB2 product. Further details about the DB2 Information Center are provided under the *Additional resources* heading of this section.

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.

Revision marks in the DB2 UDB Release Notes indicate text that has been added or changed. A vertical bar (|) indicates information that was added or changed for the current release.

Directory paths:

Windows® systems use backslashes (\) to delimit directories in a directory path. Linux™ and UNIX® 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 or UNIX 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:

The latest version of the DB2 documentation is always the version of the DB2 Information Center hosted at http://publib.boulder.ibm.com/infocenter/db2help/ and any late-breaking changes contained in these Release Notes. You can also install the DB2 Information Center on your own computer or intranet server by using the DB2 HTML Documentation CD that came with your DB2 product. Once you have installed the DB2 Information Center, you should apply the latest documentation FixPak available from the DB2 Support site at http://www.ibm.com/software/data/db2/udb/support/downloadv8_docfix.html. However, DB2 documentation FixPaks are released infrequently and may not reflect the most current level of the DB2 documentation. DB2 product manuals in PDF are also updated infrequently and may not reflect the most current level available.

The DB2 UDB Support Web site contains technotes and service flashes that describe changes to the documentation and known limitations and workarounds discovered after this document was created. The DB2 UDB Support Web site address is http://www.ibm.com/software/data/db2/udb/support.html.

Information about migrating DB2 products is available at http://www-1.ibm.com/support/docview.wss?rs=71&uid=swg21200005 (DB2 migration portal).

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

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/

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 has not been updated since Version 8.2 FixPak 3 (equivalent to Version 8.1 FixPak 10). For details of fixes, see the Product Fix History section of the Release Notes. As well, the FixPak Readme may also contain descriptions of new items in this release.

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/apars.html

Compatibility issues

Revision marks indicate text that has been added or changed. A vertical bar (|) indicates information that was added or changed for Version 8.2 FixPak 4 (equivalent to Version 8.1 FixPak 11).

Backward compatibility

Fixpak level and installation of new products

You might need to install a DB2® product that is at a different level than the version of another DB2® product that is currently installed on the computer. DB2 products are required to be at the same level.

If the product you are installing is at a more recent level than the version of other DB2 products installed on the same computer, you will need to update your existing DB2 products to the more recent level. For example, if you are installing DB2 Connect™ for iSeries™ at a level of Fixpak 10 and your other DB2 products are at a level of Fixpak 9, you need to apply Fixpak 10 to currently installed DB2 products before installing DB2 Connect™ for iSeries™ at a level of Fixpak 10.

Conversely, if you are installing a product on a computer that has a more recent version of a DB2 product installed, there are some guidelines to follow:

On Windows® operating systems

The fixpak can be used to install the product directly on the system at the same level. The license can be added after the installation is complete by using the following command:

```
db2licm -a filename
```

where *filename* is the name of the license file, which can be found on your original media under the `db2\license` directory. You can also add this license to the `db2\license` directory of the fixpak, and the license will be installed by the installation.

On UNIX® and Linux® operating systems

Prerequisites:

Before you install an extra product or component, you must stop the following items:

- Existing DB2 instances
- The DB2 Administration Server (DAS)

The instances and DAS that must be stopped are the ones that belong to the DB2 installation where the extra DB2 product or component will be installed.

Refer to the *FixPak Readme* for further instructions.

Procedure:

1. There are three methods for installing an extra product or component at a DB2 level lower than the currently installed DB2 product or products on the system. Choose one of the following methods:

   Run the db2setup program

   Run db2setup interactively with the GUI or silently with a response file. Do not perform any configuration, such as instance creation, during the installation of the extra product or component with db2setup.

   If the DB2 DAS does not exist on the current system, and the extra product or component requires or supports the DB2 DAS, db2setup will set up the DB2 DAS during the installation. On some platforms, you may experience errors during the DB2 DAS creation with db2setup. These errors are expected and can be ignored.

   The db2setup program can be found on the DB2 product CD or image for the extra product or component you are installing.

   Refer to the *Command Reference* guide and the *Installation and Configuration Supplement* for detailed information about using db2setup.

   Run the db2_install script

   The db2_install script installs any components which are not currently installed in your DB2 installation, except non-English language and message components. Therefore, you should use db2_install to install new products or components, as it will not update existing DB2 components.

   The db2_install script can be found on the DB2 product CD or image for the extra product or component you are installing.

   Refer to the *Installation and Configuration Supplement* for detailed information about using the db2_install script.

   Use the system installer

   Use the system installer to install new products or components.

   Refer to the *Installation and Configuration Supplement* for detailed information about using the system installer.

2. The following tasks must be performed after installing an extra product or component:

   a. Reapply the regular fixpak to all pre-existing products such that new and pre-existing products are at the same level.

      To illustrate this scenario, assume the following conditions:

- DB2 Universal Database™ Enterprise Server Edition is currently installed at a level of FixPak 10.
- Next, you install DB2 Query Patroller™ at FixPak 7 according to the instructions in the previous step.

As a post-installation step, you must reapply regular FixPak 10.

Note: During fixpak installation, you might get an error message similar to the following:

```
The package db2cliv81 is already installed on
the system.

Patch nnnnnnn-nnn installation terminated
abnormally.

To reinstall this patch, deinstall it first
before attempting to reinstall it.
```

This error occurs because the db2cliv81 in the system is already on the level same as the fixpak level being installed. You can ignore this kind of error. Use the system installer to confirm that the DB2 component or package is indeed at the same fixpak level being installed.

b. Run the `db2iupdt` command to update the existing DB2 instances that belong to the current DB2 installation.

c. Run the `dasupdt` command to update the DB2 DAS associated with the current DB2 installation.

d. If needed, run the `db2isetup` command to create a new DB2 UDB instance or configure an existing instance.

Refer to the *FixPak Readme* for details about fixpak installation, instance and DB2 DAS update, and other post-installation steps.

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.

An amendment to that statement is as follows:

> While connections from Version N clients to Version N + 2 servers are possible in some environments, the DB2 support team will only provide support for this configuration as long as Version N is still in service. Once Version N is withdrawn from service, this configuration is no longer supported by the DB2 support team. DB2 Version 7 clients connecting to a DB2 Version 8 server is no longer supported by the DB2 support team because Version 7 has been withdrawn from 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.

Alternate FixPaks (Linux and UNIX)

Prior to DB2 Universal Database (UDB) Version 8, FixPaks functioned only as updates to installed DB2 UDB packages or file sets in one fixed location. This meant that FixPaks installation replaced existing files with updated ones, which were provided in the FixPaks. Multiple DB2 FixPak levels could not exist on a single system. Now, DB2 UDB Enterprise Server Edition (ESE) can exist at multiple fix pack levels in the same system for Linux™–based and UNIX®–based operating systems. This feature, supported in production operating environments since Version 8.1.2, is achieved using the following two FixPak types:

regular FixPaks

- Are available not only for ESE, but for all supported DB2 Version 8 products for the related platforms
- Can be installed directly on top of the existing installation either in the `/usr/opt/db2_08_01` directory on AIX® or the `/opt/IBM/db2/V8.1` directory on other platforms
- Can not be applied on top of any installed alternate FixPaks

alternate FixPaks

- Can be installed as completely new copies of DB2 UDB ESE
- Are installed in a predefined location other than the location used for a regular DB2 UDB installation
- There is no GUI installation procedure for alternate FixPaks

Notes:

1. You are *not* required to perform a multiple FixPak installation if it is unnecessary for your environment. You might consider installation of multiple FixPaks if you need DB2 UDB Version 8 ESE instances at different fixpak levels in the same system. For example, multiple FixPaks enable you to verify the changes contained in the FixPak in your test environment without affecting your production systems.
2. Starting with IBM DB2 UDB Enterprise Server Edition (ESE) for Linux and UNIX, Version 8.1.2, fix packs are supported in production operating environments when they are installed as Multiple fix packs.
3. On Linux, alternate FixPaks are available on the following platforms only:
   - x86 (32-bit)
   - S/390® (31-bit)

4. Two or more DB2 instances running at different fixpak levels on the same system do not support operations which make DB2 Internal Procedure Calls (IPCs), such as Federated queries. All instances involved in such operations on the same system should be at the same DB2 fixpak level.
5. DB2 UDB Version 8 alternate FixPaks only support DB2 ESE on supported Linux and Unix platforms.

To update a multiple FixPak instance to a different FixPak level, perform one of the following operations:

- Install the appropriate regular FixPak on the General Availability (GA) installation and update the instance by running **db2iupdt** from the existing GA path.
- Install the appropriate alternate FixPak to its own unique path and update the instance by running **db2iupdt** from this path.

For further information regarding alternate FixPaks:

- Refer to the latest FixPak Readme
- Visit the IBM support site at http://www.ibm.com/software/data/db2/udb/support.html

Query Patroller Version 8.2.2 query data compatibility with earlier FixPaks

Starting with Version 8.2.2 (equivalent to Version 8.1 FixPak 9), the contents of the TRACK_QUERY_INFO Query Patroller control table that were captured in a 32–bit environment can be used in a 64–bit environment. This capability eases the migration effort to a 64–bit environment. Information captured in the TRACK_QUERY_INFO Query Patroller control table at Version 8.2.2 cannot be used to generate historical data for that query or to execute held queries under any previous FixPak level.

Data Warehouse Center previous server support restrictions

The following limitations exist for previous server support for DB2 Universal Database (UDB) Enterprise Server Edition Version 8 Data Warehouse Center:

Large Object (LOB) support

- If you are using a warehouse control database on a server that is earlier than DB2 UDB Enterprise Server Edition Version 8 you cannot work with LOBs. You must upgrade the warehouse control database to the correct level, or you can move the control database to the system on which the DB2 UDB Enterprise Server Edition Version 8 warehouse server is installed and use the warehouse control database locally from that system.
- To move LOBs between the Data Warehouse Center and DB2 UDB, you must upgrade to DB2 UDB Enterprise Server Edition, Version 8.

Systems Network Architecture (SNA) support

If you use SNA to connect to your warehouse sources and targets, you must change the configuration to TCP/IP over SNA or use the Windows NT® warehouse agent.

Support for EXPORT and LOAD utilities

The Data Warehouse Center Version 8 LOAD utility does not support a Version 7 target database. If you want to keep your target as a Version 7 database, then you must change the LOAD step to a SQL Select and Insert step. SQL Select and Insert steps use a DELETE* statement followed by SELECT and INSERT statements. SQL Select and Insert steps require the database to log all transactions. As a result, the performance for SQL Select and Insert steps is not as efficient as it is for the EXPORT and LOAD utilities.

Development Center APARs required for SQLJ and SQL Assist support on DB2 UDB for OS/390, Version 6 and DB2 UDB for z/OS, Version 7

When using the Development Center on an Application Development client for DB2 Universal Database (UDB) Version 8 on Windows or UNIX operating systems, the following APARs need to be installed on the server to enable SQLJ and SQL Assist support:

DB2 UDB for z/OS, Version 7

- PQ65125 - Provides SQLJ support for building Java™ SQLJ stored procedures
- PQ76858 - Provides SQL Assist support

DB2 UDB for OS/390®, Version 6

- PQ76858 - Provides SQL Assist support

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 other 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, the following updates are logged in the db2diag.log file:

- db2set profile registry updates
- database and database manager configuration parameter updates

The messages for these updates 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

JDK 1.4.2 supported by DB2 Universal Database for Linux, UNIX, and Windows

DB2 Universal Database™ (UDB) for Linux, UNIX, and Windows®, Version 8.2.2 (equivalent to Version 8.1 FixPak 9), supports JDK 1.4.2 on all DB2 UDB supported 32–bit and 64–bit workstation operating system environments. This support includes, but is not limited to, support for building and running Java™ client applications, building and running Java™ routines from the command line, building and running Java™ routines from the DB2 Development Center where it is supported, as well as for running other DB2 tools.

When you install DB2 UDB, Version 8.2, the latest supported version of the Java developer kit will also be installed if it is not already installed, unless the DB2 UDB installation is an update of a previous DB2 UDB Version 8 installation. If you are updating a previous installation of DB2 UDB Version 8, you must install the Java developer kit from the CD.

The following table indicates the DB2 supported 32–bit and 64–bit workstation operating system environments and the latest supported JDK level for each of them. For information about earlier JDK support, refer to the Java Application Development Web page at http://www.ibm.com/software/data/db2/udb/ad/v8/java/.

Table 1. DB2 supported environments with corresponding supported JDK levels

DB2 supported environment	Latest supported JDK level
Windows IA/AMD 32–bit	JDK 1.4.2
Windows IA 64–bit	JDK 1.4.2
Windows AMD/EM64T 64–bit	JDK 1.4.2
AIX® 4.3.3 32–bit	JDK 1.3.1 SR6 [2]
AIX® 5 (hybrid [1])	JDK 1.4.2
Solaris (hybrid [1])	JDK 1.4.2
HPUX RISC & Itanium (hybrid [1])	JDK 1.4.2.01
Linux AMD/EM64T 32-bit, 64–bit (hybrid [1])	JDK 1.4.2 [3]
Linux IA 32–bit	JDK 1.4.2
Linux IA 64–bit	JDK 1.4.2
Linux 390 31–bit	JDK 1.4.2
Linux 390 64–bit	JDK 1.4.2

Table 1. DB2 supported environments with corresponding supported JDK levels (continued)

DB2 supported environment	Latest supported JDK level
Linux PPC (hybrid [1])	JDK 1.4.2

Notes:

1. Hybrid refers to an installation image that contains 32–bit and 64–bit support
2. JDK 1.3.1 Service Release 6 is the only JDK version supported for AIX® 4.3.3.
3. There is no DB2 graphical user interface tools support on Linux AMD/EM64T (32-bit and 64-bit) with JDK 1.4.2.

An updated procedure for setting up the Linux Java Environment is provided next.

Setting up the Linux Java environment

Prerequisites:

- Before implementing the following instructions, perform the setup outlined in the topic "Setting up the UNIX® Java environment", which can be found in the *Application Development Guide: Building and Running Applications* Guide.
- The commands in the following instructions require root authority.

Procedure:

To build Java applications on Linux with DB2 JDBC support:

1. Install and configure one of the supported developer kits listed in the topic "Linux supported development software", which can be found in the *Application Development Guide: Building and Running Applications* Guide.

   To run Java stored procedures or user–defined functions, the Linux runtime linker must be able to access certain Java shared libraries, and DB2 UDB must be able to load both these libraries and the Java virtual machine. The process that runs stored procedures and user–defined functions loads libraries only in secure locations, as defined in the `/etc/ld.so.conf` file. One of these secure locations is `/usr/lib`. The remaining instructions show which libraries require symbolic links in `/usr/lib`.
2. Create symbolic links in `/usr/lib` to point to the Java shared libraries. Depending on the JDK version that you are using, you will have link to different shared libraries:

   For the IBM® Developer Kit 1.3

   Create symbolic links to libjava.so, libjvm.so, and libhpi.so. You can create the symbolic links by running the following commands as root:

   ```
   cd /usr/lib
   ln -fs JAVAHOME/jre/bin/libjava.so .
   ln -fs JAVAHOME/jre/bin/classic/libjvm.so .
   ln -fs JAVAHOME/jre/bin/libhpi.so .
   ```

   where `JAVAHOME` is the base directory for the IBM® Developer Kit. If DB2 UDB cannot find these libraries, you will get a -4301 error when trying to run a Java routine, and there will be messages in the administration notification log about libraries not found.

   For the IBM® Developer Kit 1.4.1

   Create symbolic links to libjava.so, libjvm.so, libhpi.so, and libjsig.so. You can create the symbolic links by running the following commands as root:

```
cd /usr/lib
ln -fs JAVAHOME/jre/bin/libjava.so
ln -fs JAVAHOME/jre/bin/classic/libjvm.so
ln -fs JAVAHOME/jre/bin/libhpi.so
ln -fs JAVAHOME/jre/bin/libjsig.so
```

where JAVAHOME is the base directory for the IBM Developer Kit. If DB2 UDB cannot find these libraries, you will get a -4301 error when trying to run a Java routine, and there will be messages in the administration notification log about libraries not found.

For the IBM Developer Kit 1.4.2 on Linux platforms other than AMD64/EM64T

Create symbolic links to libjava.so, libjvm.so, libhpi.so, libjsig.so, libjitc.so, libxhpi.so, and libdbgmalloc.so. You can create the symbolic links by running the following commands as root:

```
cd /usr/lib
ln -fs JAVAHOME/jre/bin/libjava.so
ln -fs JAVAHOME/jre/bin/classic/libjvm.so
ln -fs JAVAHOME/jre/bin/libhpi.so
ln -fs JAVAHOME/jre/bin/libjsig.so
ln -fs JAVAHOME/jre/bin/libjitc.so
ln -fs JAVAHOME/jre/bin/libxhpi.so
ln -fs JAVAHOME/jre/bin/libdbgmalloc.so
```

where JAVAHOME is the base directory for the IBM Developer Kit. If DB2 UDB cannot find these libraries, you will get a -4301 error when trying to run a Java routine, and there will be messages in the administration notification log about libraries not found.

For the IBM Developer Kit 1.4.2 on Linux AMD64/EM64T

This developer kit is different from the kit on other Linux platforms. Follow the instructions outlined in the *Alternative Procedure* section that follows, and put the following line in /etc/ld.so.conf:

```
JAVAHOME/jre/bin
```

where JAVAHOME is the base directory for the IBM Developer Kit. If DB2 UDB cannot find these libraries, you will get a -4301 or -1042 error when trying to run a Java routine.

Alternative procedure:

Instead of explicitly creating links to the shared libraries in the /usr/lib directory, you can add the name of the directory that stores the Java shared libraries to the /etc/ld.so.conf file. This file requires root permission. After updating /etc/ld.so.conf , you must run the **ldconfig** command as root to activate your changes. If you encounter any problems with this alternative procedure, create the links in the /usr/lib directory as previously instructed.

Microsoft XP fix is needed on 64-bit operating systems

If you are using the Microsoft® XP 64–bit 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

The following DB2 products are supported on Windows XP for development or test purposes only (production environments require Windows 2000 or Windows Server 2003):

- DB2 UDB Enterprise Server Edition
- DB2 Connect Enterprise Edition

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.

Raw I/O log enablement (Linux with 2.6 kernel)

To use logs with raw I/O devices prior to DB2 Universal Database (UDB) Version 8.2.2 (equivalent to Version 8.1 FixPak 9), it was necessary to bind a physical device to the Linux raw character device driver with the `raw` utility. Starting with DB2 UDB Version 8.2.2 (equivalent to Version 8.1 FixPak 9), on the 2.6 Linux kernel, raw I/O for logs can be specified directly. For example, to use device partition `/dev/sdb1` for raw logs for the SAMPLE database, issue the following command:

```
db2 update db cfg for sample using newlogpath /dev/sdb1
```

Although DB2 UDB still supports the method of using the raw utility for raw I/O, recent distributions have deprecated this feature and might remove it in the future. The preferred method is to use the new method by specifying the devices directly.

Red Hat Linux support with the Data Warehouse Center

DB2 Universal Database, Version 8.2 supports Red Hat Enterprise Linux AS Versions 3 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.

Connection concentrator required with WebSphere MQ Transaction Manager and DB2 for OS/390

When running applications in an IBM® WebSphere® MQ (formerly known as IBM MQSeries®) environment, WebSphere® MQ can act as an XA-compliant transaction manager, coordinating any distributed, two-phase commit transactions. When WebSphere® MQ is acting as a transaction manager in this way, and the data sources are from the DB2 family of products, there are several configuration requirements. Most of these requirements are already documented. For example, you must set the DB2 configuration parameter TP_MON_NAME to ″MQ″ at the DB2 runtime client.

However, there is a configuration requirement that has not been documented. The requirement is specific to DB2 Connect when connecting to data sources that are DB2 for OS/390® servers: When using WebSphere MQ to coordinate distributed transactions involving DB2 for z/OS® and DB2 for iSeries servers, the DB2 Connect connection concentrator feature must be enabled at the gateway. The connection concentrator is enabled when the value of the MAX_CONNECTIONS configuration parameter is greater than the value of MAX_COORDAGENTS. If you do not enable the connection concentrator, unexpected transaction behavior will result.

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 2. 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 UNIX and 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 UNIX and 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 3. 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 UNIX and Linux, the `sqllib/conv` directory is linked to the installation 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 4. 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 5. 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 UNIX and 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.

MVS operating system is not supported

Despite being mentioned in the documentation, the MVS™ operating system is no longer supported by DB2 Universal Database. MVS has been replaced with z/OS.

Backup and restore operations (Linux 390)

Backup and restore operations to and from multiple tape devices might not work if you are using the Linux 390 operating system.

Enabling view docking when accessing the Development Center with Hummingbird Exceed

When accessing the Development Center on UNIX with Hummingbird® Exceed, the XTEST extension version 2.2 must be enabled before you can move and dock views by dragging their title bars within the Development Center.

To enable the XTEST extension:

1. From the Start menu, select **Programs** –> **Hummingbird Connectivity 7.0** –> **Exceed** –> **XConfig**. The XConfig window opens.
2. Optional: If your configuration requires a password, enter the XConfig password.
3. Double click the **Protocol** icon. The Protocol window opens.
4. Select the **X Conformance Test Compatibility** checkbox.
5. In the **Protocol** window, click the **Extensions...** button. The Protocol Extensions window opens.
6. In the Enable Extensions list, select the **XTEST(X11R6)** checkbox.
7. Click **OK**.

Installation, migration, upgrade, and configuration information

Revision marks indicate text that has been added or changed. A vertical bar (|) indicates information that was added or changed for Version 8.2 FixPak 4 (equivalent to Version 8.1 FixPak 11).

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 UNIX and 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.

New options for the DB2 Setup wizard (Windows)

Specifying the installation path at the command line

The DB2® Setup wizard allows the installation path to be specified at the command line using the new option:

`-p` *`Installation Directory`*

This option overrides any value specified in the response file. The new `-p [INSTALLDIR]` option can be used to change the installation path of the product without changing the response file.

New NO_CONFIG option to the installation

The NO_CONFIG keyword, when added to the response file, disables any up and running operations. The installation still sets environment variables and performs other basic setup that is required to create instances and to create the DB2® Administration Server manually after the installation has completed. No nonessential configuration is done.

If this option is specified in the response file, any keywords that are used during up and running setup are validated, but the configuration associated with these keywords is not done.

Progress display

A new option has been provided to display only the progress window of the DB2® Setup wizard when running an unattended installation.

The new `-m` option must be used in conjunction with `-u [response file]` parameter. If the `-m` option is specified without the response file option, the installation returns a message indicating that the argument is incorrect. If the `-m` option is specified, only the windows that do not require any user input are displayed.

For example, once the setup.exe command is run, the installation displays windows, but it does not prompt for any input. When the installation ends, the windows close and the setup.exe process finishes.

Changes to the DB2 Setup wizard (Windows)

DB2 Setup wizard options (setup.exe)

The setup.exe file that starts the DB2 installation now waits until the installation exits before returning. This setting is the default behavior. Currently, a `-w` option exists, however this parameter no longer changes the behavior of the DB2 Setup wizard.

You can use the new `-c` option to force the DB2 Setup wizard to exit immediately after starting the installation. If you select this option, the return code of the installation is not available by monitoring the exit code of the setup.exe file.

An error is returned if the `-w` and `-c` options are specified together.

Extended security installation restrictions (Windows)

DB2 Universal Database™ (UDB) Version 8.2 introduced a new feature that secures access to DB2 resources by exploiting the security features in the NTFS file system and security features of the Windows® operating system. Secure access is accomplished mainly by modifying the Access Control Lists (ACL) of DB2 files, registry entries and runtime memory objects.

By default, the DB2 installation wizard enables this new security feature. Windows® extended security can be disabled by performing a custom installation and clearing the **Enhanced Windows® security** checkbox in the Enable operating system security for DB2 objects window.

The only recommended way to disable the new security feature is by running the **db2secv82** command. However, the **db2secv82** command must be run immediately following installation. More information on the **db2secv82** command can be found in the DB2 Information Center at http://publib.boulder.ibm.com/infocenter/db2help/ for details.

You can install DB2 UDB with a user ID, but unless that user ID belongs to one of the DB2ADMNS, DB2USERS, or the Local Administrators groups, that user ID will not be able to run any DB2 commands. Only users who belong to the DB2ADMNS, DB2USERS, or Local Administrators groups have access to the DB2 system files, registry keys, network shares, and DB2 services on the local computer where DB2 UDB runs.

Generally, users that belong to the DB2ADMNS group, as well as users that belong to the Local Administrators group, have full control of DB2 resources. Users that belong to the DB2USERS group have write and execute access.

After installing DB2 UDB Version 8.2, users who need read access (to run queries, for example) must be added to the DB2USERS groups. Users who need full control access, such as database administrators, must be added to the DB2ADMNS groups.

User management needs to be done in accordance with the security policies in your environment.

Affects of the new security measures on a typical installation

During a typical installation of DB2 UDB on Windows, extended security is enabled by default. By default, DB2 UDB creates the DB2ADMNS and DB2USERS groups. If there is a conflict with existing group names, you will be prompted to change the group names. If required, you can specify your own values.

If you choose to use an existing security group, the security policies for that group will be modified to satisfy what is required by DB2 UDB. When you choose an existing security group, you might be elevating users' privileges if the users already exist in that group.

Affects of the new security measures on a custom or FixPak installation

During a custom or FixPak installation of DB2 UDB on Windows, the **Enable operating system security for DB2 objects** panel opens. If you want to enable the enhanced Windows security, you can accept the default by clicking Next.

If you want to disable the enhanced Windows security, you can clear the **Enable operating system security** check box in the **Enable operating system security for DB2 objects** window.

Adding user IDs to access DB2 UDB after installation

After a typical or custom installation, you must add your user IDs to the appropriate DB2ADMNS and DB2USERS user groups to give them access to DB2 UDB. To add users to the appropriate group:

1. Start the Users and Passwords Manager tool.
2. Select the user name you want to add from the list.
3. Click **Properties**. In the Properties window, click the Group Membership tab.
4. Select the **Other** radio button.
5. Select the appropriate group from the drop-down list.

Enabling the enhanced Windows security after installation

If you installed DB2 UDB and chose not to enable the new security feature, you can still do so after installation by running the db2secv82.exe command.

Once you enable this security feature using the db2secv82.exe command, you have two options for backing out:

Option 1
Immediately run the db2secv82.exe -r command again without making any additional changes to the system. If there have been any changes at all made to the system you must use option 2.

Option 2
Add the Everyone group to the DB2ADMNS and DB2USERS groups. Adding the Everyone group to these security groups effectively gives all users all privileges to use DB2 UDB.

However, it is not recommended that you disable security because this could result in unwanted security exposures. Disabling security could result in some users being unable to use DB2 UDB because the ACL created by DB2 UDB at runtime no longer matches the ACL that has been applied to the file system.

Directory path cannot contain blanks (Linux and UNIX)

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.

Red Hat Enterprise Linux 4 Requirement

The compat-libstdc++-33 package must be installed prior to installing DB2 Universal Database for Linux™ on Red Hat Enterprise Linux® 4.

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

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-ppc-1.4.1-2.0.ppc.rpm).

If a later level of the RPM (such as IBMJava2-SDK-ppc-1.5.0-2.0.ppc.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:

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

   where *path* is the 1.5 installation path, such as `/opt/IBMJava2-15`.

Uncompressing installation images (Linux and UNIX)

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 and UNIX)* 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.

If you have more than one DB2 product installed then you will be required to download and uncompress an image for each DB2 product before installing.

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 then uncompress each of these FixPak images to the same 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 and UNIX)

Prerequisites:

Before you start the DB2 Setup wizard:

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

  Note: DB2 UDB Version 8.2.2 (equivalent to Version 8.1 FixPak 9) introduces a new set of installation images for Linux® distributions on x86 (32-bit Intel® and AMD) and x86-64 (64-bit Intel® EM64T and AMD64) architectures based on the 2.6 kernel, such as Red Hat Enterprise Linux 4 and SuSE Linux Enterprise Server 9. Ensure that you have acquired the appropriate set of installation images for your Linux distribution before proceeding with the installation.
- 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 and UNIX)* 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.

Note: The DB2 Run-Time Client Lite product uses other command line parameters. Visit the DB2 Information Center at http://publib.boulder.ibm.com/infocenter/db2help/ for details.

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 and UNIX)

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 6. 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

Binding Query Patroller packages after applying fixpaks

If you have Query Patroller installed, after applying a fixpak and performing all post-fixpak installation tasks, perform the following steps:

1. Log in as a user with DBADM authority.
2. Change to the proper directory:
   - `INSTPATH/bnd` (Linux and UNIX)
   - `INSTPATH\bnd` (Windows)

   where INSTPATH is the DB2 UDB instance path.
3. Run the following commands:

```
db2 connect to dbname
db2 bind @qpserver.lst blocking all grant public
```

Query Patroller installation at FixPak 3 level or later

Query Patroller is a query management system used to control the flow of queries against your DB2 UDB database. In DB2 UDB Version 8.1.2, DB2 Query Patroller became a stand-alone product. It is no longer a component of the DB2 Warehouse Manager.

If you have DB2 UDB Version 8 FixPak 3 or later installed and you install the DB2 Query Patroller base or GA version, you must reapply DB2 UDB FixPak 3 or later. Otherwise, the Query Patroller changes are not applied to the DB2 UDB FixPak level you are at.

If you are installing the Query Patroller server, you must also update the DB2 UDB instances after reapplying the DB2 UDB FixPak level. The instances must be restarted after they are updated.

Query Patroller server installation

When you are installing Query Patroller server note the following:

- If you select **Computer usage based** from the **Select the installation type** window using the DB2 Setup wizard, and you select **Server** from the **Select how this computer will be used** window, the Query Patroller Center will not be installed. If you want to install the Query Patroller Center, select either the **Complete** or the **Custom** installation types from the **Select the installation type** window.
- The *DB2 Query Patroller*™ *Guide* uses installation panels from the Computer usage based installation type to illustrate how to install Query Patroller server using the DB2 Setup wizard. You do not have to use the Computer usage based installation type to install Query Patroller server. You can use either the Complete or the Custom installation types to install Query Patroller server.
- DB2 Clients which have Query Patroller Client Tools installed must connect to a partition that has the Query Patroller server installed. In a typical partitioned database environment, the Query Patroller server is installed on all partitions, so you can choose any partition to be the coordinator partition and still be able to use the Query Patroller client tools.

Defining a new database partition group in Query Patroller

When you define a new table space during Query Patroller installation in a partitioned environment, you can select an existing database partition group. Alternatively, you can set up a new database partition group for the new table space. If you choose to set up a new database partition group, you can select only one database partition from the list box on the install panel for that new database partition group. If you want to add additional partitions into the new database partition group, you must add them manually after the installation is complete.

Query Patroller client tools installation

DB2 clients do not require the Query Patroller client tools to be installed on them in order to submit queries to the Query Patroller server.

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.

DB2 license policy for DB2 Universal Database Workgroup Server Edition

The Internet license policy is not valid for DB2 Universal Database Workgroup Server Edition. If you require a license for Internet users, you need to purchase DB2 Universal Database Workgroup Server Unlimited Edition.

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 installation 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 7. 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)
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 section titled "Setting up the Linux Java Environment" in the Compatibility Issues section of these Release Notes shows how to create the links to a Java Development Kit on a Linux client.

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 (UNIX and Linux)

If you are installing the DB2 Information Center on UNIX and 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*.

Ready for Tivoli enablement (UNIX)

When you purchase an IBM software product that carries the Ready for Tivoli® logo, you can manage your IBM software products through various Tivoli offerings. The Tivoli products allow you to automatically discover, monitor, and inventory one or more Ready for Tivoli applications.

IBM software products that are Ready for Tivoli can be managed through products such as IBM Tivoli Configuration Manager. IBM Tivoli Monitoring for Databases provides support for all leading databases including DB2 Universal Database (UDB), Oracle, and Informix®.

For additional information, point your browser to the IBM Web site at http://www.ibm.com/software/tivoli/

Ready for Tivoli Instrumentation and DB2 UDB Version 8:

To install and configure your instrumentation, have your Tivoli administrator do the following:

1. The Tivoli Ready signature file for this DB2 product is named xxx.sys. As of FixPak 4 these signature files will be refreshed per FixPak instead of per version release. Confirm the signature files are installed in the directory:
   ```
   %install DB2DIR%/tivready
   ```
2. Install and configure Tivoli GEM 2.2 Tivoli Ready enablement on all the computers that you intend to monitor. To download Tivoli Ready enablement

and detailed installation and use instructions, point your browser to http://www.ibm.com/software/sysmgmt/products/support/

3. Advanced database manageability is achieved through IBM Tivoli Monitoring for Databases. The ITM for Databases product uses the new advanced edition Distributed Monitoring product (called IBM Tivoli Monitoring or ITM) and provides significant enhancement in monitoring capability based on the use of this new monitoring engine. ITM for Databases provides DB2 UDB support through a Proactive Analysis Component (PAC). This PAC is tightly integrated with DB2 UDB and provides an out-of-the-box set of monitors for quick deployment and activation. Custom monitors, thresholds, and tasks can also be defined by the DBA.

   Additional information available at: http://www.ibm.com/software/tivoli/

Information on other Tivoli products is available at: http://www.ibm.com/software/tivoli/

Installation image reduction tool – db2iprune (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 documentation installation restriction (Windows)

On Windows, do not install the DB2 Universal Database (UDB), Version 8 Information Center (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 8. 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

In addition the topics presented here, late-breaking information about migrating DB2 products can be found at the DB2 Migration Portal: http://www-1.ibm.com/support/docview.wss?rs=71&uid=swg21200005

db2updv8 technote

For the latest information on **db2updv8**, including important usage details and workarounds for common problems, see the technote 1178848 at http://www-1.ibm.com/support/docview.wss?rs=71&uid=swg21178848

After running db2updv8, moving back to V 8.1 is not supported

Once you have run the **db2updv8** command to update the system catalogs to the current Version 8 level, falling back to V 8.1 is not supported.

Migration of EXPLAIN tables

In order to create the EXPLAIN tables, the **db2updv8** command must be run when applying DB2 Universal Database™ (UDB) Version 8.2.2 (equivalent to Version 8.1 Fixpak 9) or higher to a DB2 UDB Version 8.2 FixPak 1 (equivalent to Version 8.1 Fixpak 8) or lower.

Running the **db2updv8** command creates the EXPLAIN_GET_MSG function, which allows EXPLAIN.DDL to be run again.

If you attempt to create the explain tables using the EXPLAIN.DDL without running the **db2updv8** command, you might receive the following error message:

```
DB21034E
  The command was processed as an SQL statement because it
  was not a valid Command Line Processor command.
  During SQL processing it returned:

  SQL0440N  No authorized routine named "EXPLAIN_GET_MSG" of
  type "FUNCTION" having compatible arguments was found.
  LINE NUMBER=26.  SQLSTATE=42884
```

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
   ```

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, UNIX, 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 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 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 to DB2 UDB Version 8.2.2 from Version 7 in an MSCS environment

This topic describes how to migrate from DB2 Universal Database™ (UDB) Version 7 to DB2 UDB Version 8.2.2 (equivalent to Version 8.1 FixPak 9) in a Microsoft Cluster Server (MSCS) environment.

Assumptions: A sample configuration will be used to illustrate how to migrate to DB2 UDB Version 8.2.2 (equivalent to Version 8.1 FixPak 9) in an MSCS environment. In this sample configuration, there are two Windows Server 2000 computers with DB2 UDB Enterprise Server Edition Version 7. The initial configuration is a DB2 instance that is comprised of two partitions. In addition, there are two Cluster Groups:

- DB2 Group 0: contains Partition 0 active on computer A. The Administration Server (DAS) is also located here.
- DB2 Group 1: contains Partition 1 active on computer B.

Prerequisite:

Before beginning the migration, back up the database.

Procedure:

To migrate from DB2 UDB Version 7 to Version 8.2.2 (equivalent to Version 8.1 FixPak 9) in an MSCS environment:

1. Set automatic failback to off.

   During the installation process you might have to restart your computer. When this happens the cluster service restarts automatically. Therefore the automatic failback must be set to off so none of the groups automatically failback.

   a. In DB2 Group 0, to disable automatic failback:
      1) From the Cluster Administrator window, right-click **DB2 Group 0**.
      2) Select **Properties**. The DB2 Group 0 Properties window opens.
      3) On the **Failback** tab, select the **Prevent failback** radio button.
      4) Click **OK**.

   b. In DB2 Group 1, to disable automatic failback:
      1) From the Cluster Administrator window, right-click **DB2 Group 1**.
      2) Select **Properties**. The DB2 Group 1 Properties window opens.
      3) On the **Failback** tab, select the **Prevent failback** radio button.
      4) Click **OK**.
2. Move DB2 Group 1 to computer A.
   a. From the Cluster Administrator GUI, click on **DB2 Group 1**.
   b. Right-click, select **Move Group**. The Owner column changes to computer A.
3. Stop the cluster service on computer B.
   a. From the Component Services window, right-click **Cluster Service**.
   b. Click **Stop**.
4. Download and install DB2 UDB Version 8.2.2 (equivalent to Version 8.1 FixPak 9) on computer B.

   Note:

   - DB2 UDB is still running and available on computer A.
   - As part of the installation process , you will be prompted to run the **db2imigr** command to migrate the DB2 instances on this computer. While running **db2imigr**, you might have to restart your computer.
5. Take the DB2 resources on computer A offline.
   a. Partition 0, Partition 1, and the DAS are all resources on computer A. These must be taken offline one at a time. To take computer A resources offline:

        1) From the Cluster Administrator window, in the left window panel, select **Groups**.
        2) Select **DB2 Group 0**. The resources for the group display in the right window panel.
            - For DB2 Group 0 the DB2 resources include Partition 0 and the DAS.
        3) In the right window panel, right-click one of the resources, select **Take Offline**.
            - Repeat this step for each of the DB2 resources in DB2 Group 0.
        4) Select **DB2 Group 1**. The resources for the group display in the right window panel.
            - For DB2 Group 1 the DB2 resources include Partition 1.
        5) In the right window panel, right-click the resource (Partition 1), select **Take Offline**.

6. Start the cluster service on computer B.

   If the installation required a restart of your computer, and, if the **Restart Type** for the cluster service is set to automatic, skip this step because the cluster service is already started.

   To start the cluster service:
    a. From the Component Services window, right-click **Cluster Service**.
    b. Click **Start**.
7. Move DB2 Group 0 and DB2 Group 1 to computer B.
    a. From the Cluster Administrator window, click **DB2 Group 0**.
    b. Right-click, select **Move Group**. The Owner column changes to computer B.
    c. Click on **DB2 Group 1**.
    d. Right-click, select **Move Group**. The Owner column changes to computer B.
8. Bring the DB2 resources on computer B online.
    a. Partition 0, Partition 1, and the DAS are now resources on computer B. These must be brought back online one at a time. To bring these resources online:
        1) From the Cluster Administrator window, in the left window panel, select **Groups**.
        2) Select **DB2 Group 0**. The resources for the group display in the right window panel.
            - For DB2 Group 0 the DB2 resources include Partition 0 and the DAS.
        3) In the right window panel, right-click one of the resources, select **Bring Online**.
            - Repeat this step for each of the DB2 resources in DB2 Group 0.
        4) Select **DB2 Group 1**. The resources for the group display in the right window panel.
            - For DB2 Group 1 the DB2 resources include Partition 1.
        5) In the right window panel, right-click the resource (Partition 1), select **Bring Online**.
    b. Migrate all DB2 instances and all DB2 databases.
        1) Migrate all DB2 instances using the **db2imigr** command:
            a) To find all DB2 instances, run the **db2ilist** command.

b) To migrate each DB2 instance, run the **db2imigr** command. This command is located under the `sqllib\bin` directory and must be run for each DB2 instance.

2) Migrate all DB2 databases using the **migrate database** command.

9. Stop the cluster service on computer A.
   a. From the Component Services window, right-click **Cluster Service**.
   b. Click **Stop**.
10. Download and install DB2 UDB Version 8.2.2 (equivalent to Version 8.1 FixPak 9) on computer A.

    Note:

    - DB2 is still running and available on computer A.
    - As part of the installation process, you might have to restart your computer.
    - You do not need to run the **db2imigr** and **migrate database** commands on computer A because you have already run these commands on computer B.
11. Start the cluster service on computer A.

    If the installation required a restart of your computer, and, if the **Restart Type** is set to automatic, skip this step because the cluster service is already started.

    To start the cluster service:
    a. From the Component Services window, right-click **Cluster Service**.
    b. Click **Start**.
12. Move DB2 Group 0 to computer A.
    a. From the Cluster Administrator window, click **DB2 Group 0**.
    b. Right-click, select **Move Group**. The Owner column changes to computer A.

    This action brings the MSCS environment back to the initial configuration.
13. If automatic failback was turned off in Step 1, you must turn it on.
    a. In DB2 Group 0, to enable automatic failback:
       1) From the Cluster Administrator GUI, right-click **DB2 Group 0**.
       2) Select **Properties**. The DB2 Group 0 Properties window opens.
       3) On the **Failback** tab, select the **Allow failback** radio button.
       4) Click **OK**.
    b. In DB2 Group 1, to enable automatic failback:
       1) From the Cluster Administrator window, right-click **DB2 Group 1**.
       2) Select **Properties**. The DB2 Group 1 Properties window opens.
       3) On the **Failback** tab, select the **Allow failback** radio button.
       4) Click **OK**.

Database migration (HP-UX on IA64)

Database migration is not supported for DB2 Universal Database (UDB) for HP-UX on IA64 throughout the Version 8.x releases.

Restoring a Version 7 DB2 backup image to a Version 8 instance is not supported on DB2 UDB for HP-UX on IA64.

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 7 (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}

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 {D8F53726-C7AD-11D4-9155-00203586D551} /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}
```

Known limitations, problems, and workarounds

The following information is the currently known limitations, problems, and workarounds, which apply only to this release. Any limitations and restrictions might or might not apply to other releases of the product. The DB2 Support Web site also contains technotes and service flashes that describe changes to the documentation and known limitations and workarounds discovered after this document was created. The DB2 Support Web site address is http://www.ibm.com/software/data/db2/udb/support.html.

Revision marks indicate text that has been added or changed. A vertical bar (|) indicates information that was added or changed for Version 8.2 FixPak 4 (equivalent to Version 8.1 FixPak 11).

Limitations

db2look - Limitation for automatic storage table space

With databases using an automatic storage table space, the **db2look** command will complete successfully. However, it will not show table space information specific to automatic storage table spaces.

SNAP_GET_DB table function returns incomplete results

When invoked in a database partitioning feature (DPF) environment with -2 (global snapshot) provided as input, the SNAP_GET_DB table function returns incomplete results for the following four columns:

- FIRST_ACTIVE_LOG
- LAST_ACTIVE_LOG
- CURRENT_ACTIVE_LOG
- CURRENT_ARCHIVE_LOG

These elements cannot be aggregated for a global snapshot. To retrieve the complete set of values for these snapshot elements, the SNAP_GET_DB table function should be called individually on each partition.

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.

Long field and LOB data cannot be reorganized with inplace table reorganization

During inplace, or online, table reorganization, long field and LOB data cannot be reorganized. Long field and LOB data can be reorganized with a classic, or offline, table reorganization by specifying the LONGLOBDATA parameter with the REORG TABLE command. However, it should be noted that while reorganization of long field and LOB data does allow free space to be reclaimed, it does not improve clustering and can significantly increase the time required to complete the reorganization.

Locking behavior in blocking cursors can appear inconsistent in client applications

In cursor stability (CS) isolation, the row associated with the current cursor position on the database server is always locked. During a blocked fetch, multiple rows can be returned in a single buffer. In most cases, the last row processed on the server side will not fit exactly and will be split across two buffers. The application will not see this split row until the second buffer is received. Therefore, none of the rows returned to the application will be locked.

In the rare case that the last row processed on the server side is an exact fit, the cursor position will remain on the last row inside the buffer, which means that the last row returned to the application will be locked.

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, UNIX, 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.

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.

DB2SystemMonitor.getServerTimeMicros method not supported

The DB2® Universal JDBC Driver method `DB2SystemMonitor.getServerTimeMicros()` is currently not supported by DB2® Universal Database™ for Linux™, UNIX®, and Windows® servers. The `DB2SystemMonitor.getServerTimeMicros()` method returns 0 when connecting to DB2® Universal Database™ for Linux, UNIX, and Windows® servers.

IBM XL C/C++ compiler run time must be installed (Linux on iSeries and pSeries systems, 64-bit)

The following requirements must be met before installing 64-bit DB2 for Linux on iSeries™ and pSeries® systems:

- You must be running on iSeries™ or pSeries® hardware that is supported by the Linux distribution.
- You must be running one of the following two distributions:
  - SUSE LINUX Enterprise Server (SLES) 9, or
  - Red Hat Enterprise Linux 3 Update 3 (Taroon)
- You must be download and install the IBM® XL C/C++ Advanced Edition V7.0 for Linux Run-Time Environment Component. For more information, see

  `http://www.ibm.com/support/docview.wss?uid=swg24007906`

  or follow these steps:

  1. Visit the XL C/C++ support page at http://www.ibm.com/software/awdtools/xlcpp/support/
  2. Under the ″Self Help: Download″ header, select ″Latest Updates (PTFs)″
  3. Under the ″Linux″ header, select ″XL C/C++ Advanced Edition V7.0 for Linux Run-Time Environment″
  4. Follow the instructions from the ″XL C/C++ Advanced Edition V7.0 for Linux Run-Time Environment″ page.

Universal JDBC Driver Type 2 not supported on DB2 UDB Version 8.2.2 (Linux AMD64)

DB2 Universal Database™ (UDB) Version 8.2.2 (equivalent to Version 8.1 FixPak 9) on Linux AMD64 does not support the Universal JDBC Driver Type 2. Support is planned for a future fixpak. JDBC support in Version 8.2.2 is available using Universal JDBC Driver Type 4 and the Legacy JDBC Type 2 driver (COM.ibm.db2.jdbc.app.DB2Driver).

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.

ConnectNode CLI/ODBC configuration keyword limitation

The ConnectNode keyword does not affect the Control Center. The Control Center always connects to the catalog node referred to by the SQL_CONN_CATALOG_NODE setting.

SQLColumns function (CLI) restriction

The SQLColumns() function does not support returning data from an alias of an alias. When called against an alias of an alias, the SQLColumns() function returns an empty result set.

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 utility restrictions

The CLI LOAD utility does not support parameter markers if the prepared SQL statement for inserting data contains a SELECT clause.

Prior to Version 8.2 FixPak 3 (equivalent to V8.1 Fixpak 10), the CLI LOAD utility did not allow target columns to be specified if the INSERT statement included a VALUES clause.

However, starting with Version 8.2 FixPak 3 (equivalent to V8.1 Fixpak 10), the CLI LOAD utility allows target columns to be specified in an INSERT statement with a VALUES clause. For example, the following statement is now supported by CLI LOAD:

```
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.

Tools catalog database creation not supported (AIX, Solaris Operating Environments, and HP-UX)

The creation of the tools catalog is not supported during the installation of DB2 Universal Database (UDB) against 64-bit instances on the hybrid 64-bit platforms. The hybrid platforms are:

- AIX
- Solaris Operating Environments
- HP-UX
- Other platforms that support both 32-bit and 64-bit instances

If you want to create a tools catalog against a 64-bit instance, this can be done after the installation of DB2 UDB either through the Command Line Processor using the **CREATE TOOLS CATALOG CLP** command or using the Control Center. You will also need to install a 64-bit IBM Developer Kit for Java for this operation. Refer to the DB2 Administration Server section of the *DB2 Administration Guide* for more details.

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.

SNA support limitations in Version 8

The following support has been withdrawn from DB2 Universal Database (UDB) Enterprise Server Edition (ESE) Version 8 for Windows and UNIX operating systems, and from DB2 Connect Enterprise Edition (CEE) Version 8 for Windows and UNIX operating systems:

- Multisite update (two-phase commit) capability using SNA cannot be used. Applications that require multisite update (two-phase commit) *must* use TCP/IP connectivity. Multisite update (two-phase commit) using TCP/IP to a host or iSeries database server has been available for several releases. Host or iSeries applications that require multisite update (two-phase commit) support can use the new capability of TCP/IP multisite update (two-phase commit) support within DB2 UDB ESE, Version 8.
- DB2 UDB ESE or DB2 CEE servers no longer accept client connections using SNA. As of Version 8, FixPak 1, DB2 UDB allows the 32-bit version of AIX, Solaris Operating Environment, HP-UX, and Windows applications to access host- or iSeries database servers using SNA. This support allows applications access to host or iSeries database servers using SNA, but using only one-phase commit.
- Sysplex support with DB2 UDB for z/OS is only available using TCP/IP. Sysplex support is not provided using SNA connectivity.
- Change password support is no longer available using SNA connectivity to host database servers.
- All SNA support will be withdrawn in the next version of DB2 UDB and DB2 Connect.

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.

Failures when reporting a successful login during a connection attempt (AIX)

When using OS authentication on AIX, DB2 Universal Database (UDB) attempts to report a successful login to AIX upon successful authentication during a connection attempt. Prior to Version 8 FixPak 5, if DB2 UDB was unable to report a successful login, the connection would fail despite the user being authenticated. Starting with Version 8 FixPak 5, the connection is allowed to continue and the failure is logged in the `db2diag.log` file.

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.

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.

GUI tools for Linux operating systems on AMD64

The GUI tools that come with DB2 Universal Database (UDB) are Java–based graphical utilities. Examples of DB2 UDB GUI tools are:

- Control Center
- Information Center
- Configuration Assistant
- Development Center

Since Linux operating systems on AMD64 do not currently have a 64–bit Java™ SDK or Runtime Environment, running these tools requires that you do so from a 32–bit DB2 UDB instance.

You can still use these tools to manage databases in local and remote 64–bit instances. For example, to use the Control Center you must create a 32–bit instance first. After a 32–bit instance is created, you need to manually catalog the local 64–bit instance that you want to administer.

More information on DB2 for Linux on AMD64 can be found at the following URL:

ftp://ftp.software.ibm.com/software/data/pubs/papers/linuxamd64.pdf

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.

Do not partition Information Catalog Center tables

Tables that the Information Catalog Manager uses must be contained within a single database partition. Numerous methods are available to put the tables within a single partition. The following procedure is one approach that works.

1. Open a DB2 Command Line Processor and issue these commands:
   a.
   ```
   CREATE DATABASE PARTITION GROUP partition_group_name
          ON DBPARTITIONNUM partition_number
   ```
   b.
   ```
   CREATE REGULAR TABLESPACE tablespace_name
          IN DATABASE PARTITION GROUP partition_group_name
          MANAGED BY SYSTEM USING ('cname')
   ```

   where *partition_group_name* is the same in both commands.
2. Click Start –> Programs –> IBM DB2 –> Set-up Tools –> Manage Information Catalog Wizard.
3. On the Options page, specify the table space name in the **Table space** field.

Query Patroller limitations when DYN_QUERY_MGMT is disabled

Query Patroller cannot perform the following actions if the database configuration parameter DYN_QUERY_MGMT is disabled:

- Release queries from a held state
- Make a running or queued query run in the background when the query is in the foreground

If you attempt to release a query from held state, or change a foreground query to a background query when DYN_QUERY_MGMT is set to DISABLE, an error message will be displayed and the state of the query will not change. If held queries are scheduled to run and DYN_QUERY_MGMT is disabled at the time they start running, an error message is written to the qpdiag.log file and the queries are left in held state.

Query Patroller result tables now use DB2QPRT schema

Starting with FixPak 5, all new result tables are created in the schema DB2QPRT rather than in the schema of the submitter.

DROPIN privilege on the DB2QPRT schema is granted to operators whose profiles were created prior to installation of FixPak 5 and had either:

- The MONITORING privilege with edit authority
- The HISTORICAL ANALYSIS privilege with edit authority

DROPIN privilege on the DB2QPRT schema is granted the first time Query Patroller creates a result table in this schema.

Operators who are given MONITORING privilege with edit authority or HISTORICAL ANALYSIS privilege with edit authority after the installation of FixPak 5 are also granted DROPIN privilege on the DB2QPRT schema upon creation or update of their profiles.

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

Exporting to an IXF file with indexes on columns containing – and + characters

When the **export** command is issued with a IXF file format and a SELECT * clause, index information is collected, if applicable.

Problem:

If the column names specified in the index contain – or + characters, the index information will not be collected and you will receive SQL code SQL27984W. The export will complete and the data being exported will not be affected. However, the index information will not be saved in the IXF file.

Workaround:

If you will be recreating the table using the **import** command with the CREATE parameter, the indexes will not be recreated. To create the indexes separately, use the **db2look** utility.

CLI0116E or SQL0428N error when calling the db2ReadLog API

Problem:

Calling the db2ReadLog API from an application can result in an error when the application disconnects from the database if a commit or rollback is not performed before the disconnect:

- A CLI0116E error might be generated if the db2ReadLog API is called from a CLI application.
- A SQL0428N error might be generated if the db2ReadLog API is called from an embedded SQL application written in C.

Workaround 1:

For non-embedded SQL applications, set autocommit mode on before calling the db2ReadLog API.

Workaround 2:

Issue a COMMIT or ROLLBACK statement after calling the db2ReadLog API and before disconnecting from the database.

The "db2gcf –k" command fails on DB2 UDB Workgroup Server 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.

Using the db2gcf system command with the –k parameter on DB2 UDB Workgroup Server will fail.

Workaround:

The "db2gcf -k" command works only on DB2 UDB Enterprise Server Edition and not on DB2 UDB Workgroup Server Edition.

SQL1224 error from DRDA wrapper (AIX)

If a 32-bit DB2 Universal Database (UDB) server is run on an AIX system and an application running on the same system has more than one local database connection through the DRDA® wrapper, then the application might get the following error:

```
SQL1822N  Unexpected error code "-1224" received from
data source "W3_SERVER2".
Associated text and tokens are
```

```
  func="DriverConnect"
  msg="SQL1224N  A database agent could not be started to
                service a request, or was terminated as
                a result of a database system shutdown
                or a force command. "
SQLSTATE=560BD
```

To avoid this error, put the following entry in the federated configuration file (*instance_directory*/cfg/db2dj.ini):

```
EXTSHM=ON
```

Note: When you add entries to the federated configuration file, you must stop and restart DB2 UDB for the changes to take effect.

Alternately, you can catalog the local DB2 UDB database as being on a TCP/IP node. For example:

```
CATALOG TCPIP NODE my_node REMOTE my_host SERVER 123;
CATALOG DB mydb AT NODE my_node;
CREATE WRAPPER drda;
CREATE SERVER my_server TYPE DB2/UDB VERSION 8 WRAPPER drda
   AUTHORIZATION "my_id" PASSWORD "my_pw"
   OPTIONS(ADD DBNAME 'MYDB');
```

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.

The simplified Chinese locale (AIX)

AIX has changed the code set bound to the simplified Chinese locale Zh_CN on:

- AIX Version 5.1.0000.0011 or later
- AIX Version 5.1.0 with maintenance level 2 or later

The code set has been changed from GBK (code page 1386) to GB18030 (code page 5488 or 1392). Since DB2 Universal Database (UDB) for AIX supports the GBK code set natively and the GB18030 code set via Unicode, DB2 UDB will default the Zh_CN locale’s code set to ISO 8859-1 (code page 819), and in some operations will also default the locale’s territory to the United States (US).

To work around this limitation, you have two options:

- You can override the locale’s 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, change your locale from Zh_CN to either ZH_CN or zh_CN. The ZH_CN locale’s code set is Unicode (UTF-8), while the zh_CN locale’s code set is eucCN (code page 1383).

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.

Merant Driver Manager incompatibility (UNIX)

There are incompatibilities with Unicode support when the Merant Driver Manager accesses the DB2 ODBC driver on UNIX. These incompatibilities cause the Merant Driver Manager to use Unicode even if the application did not request Unicode use. This situation can lead to problems with components such as the Data Warehouse Center, Information Catalog Manager, and MQSI, which require the Merant Driver Manager to support non-IBM data sources. You can use an alternate DB2 ODBC driver library without Unicode support enabled until a permanent solution is available.

An alternative DB2 ODBC driver library without Unicode support enabled is included with DB2 Universal Database (UDB) Version 8.1 for AIX, HP-UX, and Solaris Operating Environment. To use this alternative library, you must create a copy of it, giving the copy the original DB2 ODBC driver library's name.

Note: The alternative (_36) library contains the Unicode functions required by the DB2 JDBC driver. Using this library allows JDBC applications, including WebSphere® Application Server, to work successfully with DB2 UDB.

To switch to the non-Unicode ODBC library on AIX, HP-UX, or the Solaris Operating Environment, see the following instructions. Because this is a manual process, you must carry it out every time you update your product, including after the application of successive FixPaks or modification levels.

Procedure:

AIX

To create the alternate library on AIX:

1. As the instance owner, shut down all database instances using the **db2stop force** command.
2. Using the DB2 administration server (DAS) instance ID, shut down the DAS instance using the **db2admin stop force** command.
3. Back up the original `db2.o` file in the `/usr/lpp/db2_81/lib` directory.
4. Using root authority, issue the **slibclean** command.
5. Copy file db2_36.o to backup file db2.o, ensuring that ownership and permissions remain consistent. Use the following commands:

   ```
   cp db2_36.o db2.o
   -r--r--r-- bin:bin for db2.o
   ```

To switch back to the original object, follow the same procedure using the backup file instead of the db2_36.o file.

Solaris Operating Environment

To create the alternate library in a Solaris Operating Environment:

1. As the instance owner, shut down all database instances using the **db2stop force** command.
2. Using the DB2 administration server (DAS) instance ID, shut down the DAS instance using the **db2admin stop force** command.
3. Back up the original `libdb2.so.1` file in the `/opt/IBMdb2/V8.1/lib` directory.
4. Copy file `libdb2_36.so.1` to backup file `libdb2.so.1`, ensuring that ownership and permissions remain consistent. Use the following commands:

   ```
   cp libdb2_36.so.1 libdb2.so.1
   -r-xr-xr-x bin:bin libdb2.so.1
   ```
5. Issue the **db2iupdt <instance>** command for each database instance and the **dasupdt <das_instance>** command for the DAS instance.

To switch back to the original object, follow the same procedure using the backup file instead of the libdb2_36.so.1 file.

HP-UX PA-RISC

To create the alternate library on HP-UX PA-RISC:

1. Shut down all database instances using the **db2stop force** command.
2. Shut down the DB2 administration server (DAS) instance using the **db2admin stop force** command.
3. Back up the original `libdb2.sl` file in the `/opt/IBMdb2/V8.1/lib` directory.
4. Copy file `libdb2_36.sl` to backup file `libdb2.sl`, ensuring that the ownership and permissions remain consistent. Use the following command to ensure the consistency:

   ```
   cp libdb2_36.sl libdb2.sl
   -r-xr-xr-x bin:bin for libdb2.sl
   ```
5. Issue the **db2iupdt <instance>** command for each database instance and the **dasupdt <das_instance>** command for the DAS instance.

To switch back to the original object, follow the same procedure using the backup file instead of the libdb2_36.sl file.

HP-UX on IA64

To create the alternate library on HP-UX on IA64:

1. Shut down all database instances using the **db2stop force** command.
2. Shut down the DB2 administration server (DAS) instance using the **db2admin stop force** command.
3. Back up the original `libdb2.so` file in the `/opt/IBMdb2/V8.1/lib` directory.
4. Copy file `libdb2_36.so` to backup file `libdb2.so`, ensuring that the ownership and permissions remain consistent. Use the following command to ensure the consistency:

   ```
   cp libdb2_36.so libdb2.so
   -r-xr-xr-x bin:bin for libdb2.so
   ```
5. Issue the **db2iupdt <instance>** command for each database instance and the **dasupdt <das_instance>** command for the DAS instance.

To switch back to the original object, follow the same procedure using the backup file instead of the libdb2_36.so file.

Other UNIX Operating Systems

Contact IBM Support if you require assistance with DB2 UDB and the Merant Driver Manager on other UNIX operating systems.

NFS APAR IY32512 – Unavailable threads (AIX)

AIX 5 NFS APAR IY32512 might cause the **db2stop** command to fail on systems with a large number of partitions.

On a server that is receiving large numbers of requests for blocking locks on files that are already locked, the lock daemon might become unresponsive. This situation occurs when all available locked threads are allocated to threads that are waiting for the locks to become available, so there is no thread available to pick up the work when the unlock request is made.

When this situation occurs the stopped nodes must be restarted. There is a DB2 Universal Database workaround for this situation by stopping the nodes one at a time using the NODENUM option of the **db2stop** command.

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.

Connection pooling must be enabled when using Sysplex in DB2 Connect

Problem:

DB2 Connect™ doesn’t route connections to another member of a Distributed Data Facility (DDF) when the connecting member of the DDF in the data-sharing group on OS390 has been shutdown. With Sysplex enabled, DB2 Connect™ routes connections to another member in DDF according to the server list.

The DB2 Connect Version 8 Sysplex was designed with agent pooling in mind. The Sysplex server list is freed if there are no agents and no connections to a database. Therefore, at least one agent must be kept to maintain the Sysplex server list.

Workaround:

Enable connection pooling by running the following commands:

```
db2 update dbm cfg using num_poolagents number
db2stop
db2start
```

where *number* is the maximum number of agents allowed to be pooled on the DB2 instance. Connection pooling is enabled when *number* is greater than 0.

Recommendation:

Set num_poolagents to -1, which resolves to half of the value assigned to the maxagents configuration parameter

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.

Tools catalog database creation fails (HP-UX)

Problem

When installing DB2 Universal Database (UDB) Version 8.2 on HP-UX, the CREATE TOOLS CATALOG command fails with SQLCODE -22209 if it runs under a 32-bit instance and the DB2 Administration Server jdk_path configuration parameter points to the HP-UX SDK 1.4. The failure occurs because DB2 UDB Version 8.2 installs the HP-UX SDK 1.4 by default but the Version 8.2 32-bit instance requires the HP-UX SDK 1.3 to run the CREATE TOOLS CATALOG command successfully.

This failure can also occur when installing DB2 UDB Version 8.1 FixPak 7 if you manually updated the DB2 Administration Server jdk_path configuration parameter to point to the HP-UX SDK 1.4, or if you dropped and recreated the DB2 Administration Server (DAS). The failure occurs because, in either of these cases, the jdk_path configuration parameter changed to point to the HP-UX SDK 1.4.

A DB2 UDB Version 8.2 32-bit instance requires HP-UX SDK 1.3 to run successfully.

Workaround 1

Create the tools catalog under a 64-bit instance.

Workaround 2

Create the tools catalog under a 32-bit instance by performing the following steps:

1. Download the HP-UX SDK 1.3 from the HP-UX Web site: http://www.hp.com/products1/unix/java/
2. Install the HP-UX SDK 1.3.
3. Update the DB2 Administration Server jdk_path configuration parameter to point to the HP-UX SDK 1.3. For example:
   ```
   db2 update admin config using JDK_PATH /opt/java1.3
   ```
4. Restart the DB2 Administration Server.

```
db2admin stop
db2admin start
```

5. Re-run the CREATE TOOLS CATALOG command under the 32-bit instance.

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 9. 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

GUI tools are not supported for zSeries servers (Linux)

With the exception of the DB2 Setup wizard, the GUI tools will not work on zSeries servers running the Linux operating system. This limitation includes any items normally launched from the Installation launchpad, such as the Quick Tour.

If you want to use the GUI tools with one of these systems, install the administrative tools on a client system with a different system configuration, and use this client to connect to your zSeries server.

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"
```

Information Catalog Center log file not generated when importing tag language files

If an Information Catalog Center log file is not generated when you import tag language files to the Information Catalog Center, perform the following troubleshooting steps:

When running db2icmimport from a command line:

- If output files were not generated (.xml, .out, .err, .log), there is probably an error on the command line. Verify that the first five arguments, which are UserId, Password, Database, Catalog, and Tagfile, are correct. View the syntax by entering the **db2icmimport** command. If this does not solve the problem, modify **db2icmimport** to capture the output of the **db2javit** command by using the -g option to save the output to a file. For example:

```
db2javit -j:com.ibm.db2.common.icm.tag.IcmImport -w: -i:
   -o:"-Xmx128m -Xms32m" -g:"d:\temp\myimport.trc" ...
```

- If a log file was not generated, there is usually a parsing error. Look at the .xml file and the .out file. If you can, insert a ″:COMMIT.CHKPID(DEBUG)″ command at the beginning of the tag language file. This command generates debug report messages and checks the .xml and .out files for parsing errors.
- After parsing, errors should appear in the .log file. When the debug report is being generated, look at the .log file and the .out file for information.
- Always check the .err file to see if there is a runtime error.

When importing tag language files using the Information Catalog Center GUI:

- When you import tag language files using the GUI interface, no .out or .err files are generated.
- If a .log or .xml file is generated, try to debug by using those files.
- If a .log or .xml file is not generated or does not help, run the import process from a command line to get more information.

Binding Query Patroller packages

If the Query Patroller packages are not bound after applying a fixpak, a user without DBADM authority or proper Query Patroller privileges can encounter the following error when using the Query Patroller Center or Query Patroller command line:

```
SQL0001N - Binding or precompilation did not complete successfully.
```

If you are using the Query Patroller Center, the SQL0001N error is logged in qpdiag.log file. If you are using the Query Patroller command line, the SQL0001N is returned to the console

Auto-bind code exists to initiate automatic binding. However, the automatic binding fails when the connecting user does not have the necessary privileges to execute all statements in the Query Patroller packages. A symptom of this problem is missing folders in the Query Patroller Center.

To avoid this problem, the qpserver.lst packages should be bound manually by a user with DBADM authority or necessary privileges after applying a fixpak.

Unavailability of ports with Query Patroller (Windows)

Problem:

Submitted queries in Query Patroller might receive SQL code -29007 when there are no more available ports on Windows XP or Windows 2003. The likelihood of this error increases with an increasing number of clients accessing Query Patroller.

Workaround:

Set the following Windows registry variables:

```
MaxUserPort=65534
TcpTimedWaitDelay=30
```

and restart your system for the changes to take effect.

Details about setting Windows registry variables can be found at the Microsoft® Help and Support website at http://support.microsoft.com/.

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 10. 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 11. Replacement sample programs for XML Extender (Linux and UNIX)

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 and UNIX 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.

Differences between SNA and TCP/IP when using DB2 Connect

When connecting to an OS/390 system using SNA, the host VTAM® layer automatically flows a commit when a new connection is made. The automatic commit allows the host side thread state to be inactive, and the thread immediately becomes inactive.

However, when connecting to an OS/390 system using TCP/IP, there is no automatic commit. The application itself must flow an explicit commit after the connection to allow the thread to be inactive on the host. Without the explicit commit, the thread is subject to an idle thread time out.

The suggested workaround is to rewrite the application so that it will perform an explicit commit if the connection goes idle after the connection.

Documentation updates

Revision marks indicate text that has been added or changed. A vertical bar (|) indicates information that was added or changed for Version 8.2 FixPak 4 (equivalent to Version 8.1 FixPak 11).

Administration: Implementation

Automatic client reroute configuration (DB2_MAX_CLIENT_CONNRETRIES and DB2_CONNRETRIES_INTERVAL)

By default, the automatic client reroute feature retries the connection to a database repeatedly for up to 10 minutes. It is, however, possible to configure the exact retry behavior using one or both of the following two registry variables:

- DB2_MAX_CLIENT_CONNRETRIES: The maximum number of connection retries attempted by automatic client reroute.
- DB2_CONNRETRIES_INTERVAL: The sleep time between consecutive connection retries, in number of seconds.

If DB2_MAX_CLIENT_CONNRETRIES is set, but DB2_CONNRETRIES_INTERVAL is not, DB2_CONNRETRIES_INTERVAL defaults to 30.

If DB2_MAX_CLIENT_CONNRETRIES is not set, but DB2_CONNRETRIES_INTERVAL is set, DB2_MAX_CLIENT_CONNRETRIES defaults to 10.

If neither DB2_MAX_CLIENT_CONNRETRIES nor DB2_CONNRETRIES_INTERVAL is set, the automatic client reroute feature reverts to its default behavior described previously.

Note:

Users of Type 4 connectivity with the DB2® Universal JDBC Driver should use the following two datasource properties to configure automatic client rerouting:

- maxRetriesForClientReroute: Use this property to limit the number of retries if the primary connection to the server fails. This property is only used if the retryIntervalClientReroute property is also set.
- retryIntervalForClientReroute: Use this property to specify the amount of time (in seconds) to sleep before retrying again. This property is only used if the maxRetriesForClientReroute property is also set.

DB2TIMEOUT registry variable clarification

The DB2TIMEOUT registry variable is no longer supported. This setting was used to control the timeout period for Windows® 3.x and Macintosh clients during long SQL queries. This feature was disabled by default.

Directories created during table space container creation

When creating table space containers, DB2 UDB creates any directory levels that do not exist.

For example, if a container is specified as /project/user_data/container1, and the directory /project does not exist, then DB2 UDB creates the directories /project and /project/user_data.

Starting with DB2 UDB V8.2, FixPak 4, any directories created by DB2 UDB are created with PERMISSION 700. This means that only the owner has read, write, and execute access.

When creating multiple instances, note the following scenario:

1. Using the same directory structure as above, suppose that directory levels /project/user_data do not exist.
2. user1 creates an instance, named user1 by default, then creates a database, and then creates a table space with /project/user_data/container1 as one of its containers.
3. user2 creates an instance, named user2 by default, then creates a database, and then attempts to create a table space with /project/user_data/container2 as one of its containers.

Because DB2 UDB created directory levels /project/user_data with PERMISSION 700 from the first request, user2 does not have access to these directory levels and cannot create container2 in those directories. In this case, the CREATE TABLESPACE operation fails.

There are two methods to resolve this conflict:

1. Create the directory /project/user_data before creating the table spaces and set the permission to whatever access is needed for both user1 and user2 to create the table spaces. If all levels of a table space directory exist, DB2 UDB does not modify the access.
2. After user1 creates /project/user_data/container1, set the permission of /project/user_data to whatever access is needed for user2 to create the table space.

Automatic storage

The format of the names for the containers has changed in such a way that the table space ID and the container ID have also changed. The new format is:

```
<storage path>/<instance>/NODE####
/T#######
/C#######.<EXT>
```

where:

- /T####### is the table space ID. For example, T0000003.
- /C####### is the container ID. For example, C0000012.

Defining a generated column on an existing table

Starting with DB2® Universal Database Version 8.2.2 (equivalent to Version 8.1 FixPak 9), generated columns can be used in unique indexes.

Generated columns cannot be used in constraints, referential constraints, primary keys, and global temporary tables. A table created with LIKE and materialized views does not inherit generated column properties.

Aggregate registry variables

When you have set DB2WORKLOAD=SAP, the user table space SYSTOOLSPACE and the user temporary table space SYSTOOLSTEMPSPACE are not automatically created. These table spaces are used for tables created automatically by the following wizards, utilities, or functions:

- Automatic maintenance
- Design advisor
- Control Center database information panel
- SYSINSTALLOBJECTS stored procedure, if the table space input parameter is not specified
- GET_DBSIZE_INFO stored procedure

Without the SYSTOOLSPACE and SYSTOOLSTEMPSPACE table spaces, you cannot use these wizards, utilities, or functions.

To be able to use the wizards, utilities, or functions, do either of the following:

- Manually create SYSTOOLSPACE (on the catalog node only, if using DPF). For example:

  ```
  CREATE REGULAR TABLESPACE SYSTOOLSPACE
     IN IBMCATGROUP
     MANAGED BY SYSTEM
     USING ('SYSTOOLSPACE')
  ```
- Call SYSINSTALLOBJECTS to create objects, specifying a valid table space, for each of the following tool names: “DB2AC”, “POLICY” and “STMG_DBSIZE_INFO”

After completing at least one of these choices, create a user temporary table space (also on the catalog node only, if using DPF). For example:

```
CREATE USER TEMPORARY TABLESPACE SYSTOOLSTMPSPACE
   IN IBMCATGROUP
   MANAGED BY SYSTEM
   USING ('SYSTOOLSTMPSPACE')
```

Once the table space SYSTOOLSPACE and the temporary table space SYSTOOLSTEMPSPACE are created, you can use the wizards, utilities, or functions mentioned earlier.

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 concurrent I/O 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, UNIX, 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.

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)

Automatic client reroute considerations for cataloging on a DB2 Connect server

Consider the following two items involving alternate server connectivity with DB2 Connect server:

- The first consideration involves using DB2 Connect server for providing access to a host or iSeries database on behalf of both remote and local clients. In such situations, confusion can arise regarding alternate server connectivity information in a system database directory entry. To minimize this confusion, consider cataloging two entries in the system database directory to represent the same host or iSeries database. Catalog one entry for remote clients and catalog another for local clients.
- Secondly, the alternate server information that is returned from a target server is kept only in cache. If the DB2 process is terminated, the cache information, and therefore the alternate server information, is lost.

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.

Kerberos authentication details

Kerberos and client principals:

You can override the Kerberos server principal name used by the DB2® Universal Database (UDB) server on UNIX® and Linux™ operating systems. Set the DB2_KRB5_PRINCIPAL environment variable to the desired fully qualified server principal name. The instance must be restarted because the server principal name is only recognized by DB2 UDB after **db2start** is run.

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_CONTINUE_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

Comparison of the DB2_FORCE_FCM_BP registry variable in 32-bit and 64-bit environments

When you enable the DB2_FORCE_FCM_BP registry variable there is one fewer shared memory segments available for other uses, particularly for database buffer pools. Enabling the DB2_FORCE_FCM_BP registry variable thus reduces the maximum size of database buffer pools. Note that because of the large number of available shared memory segments in a 64-bit environment, this reduction in the number of shared memory segments should be an issue only in 32-bit environments.

RUNSTATS recommended after table creation

When a table is first created, system catalog statistics are set to -1 to indicate that the table has no statistics. Until statistics are gathered, DB2 UDB uses default values for SQL statement compilation and optimization. Updating the table or index statistics might fail if the new values are inconsistent with the default values. Therefore, run the **runstats** command on a table or index before manually updating statistics for either.

New reason code for SQL1169N

SQL error message SQL1169N has a new reason code 5 to indicate that a column of an explain table is too small.

- Explanation for reason code 5: A column of an explain table is too small. The db2diag.log will contain messages identifying the schema name, table name and column name as well as the column size required to contain the data generated by the Explain facility.
- User Response: Decide whether the required column size can be accommodated in your table and database. If it can, increase the column size using the ALTER TABLE statement.

Optimization strategies for MDC tables

The following text is an update to *Administration Guide: Performance, Chapter 6. Understanding the SQL compiler.*

MDC roll out may be used even if a RID index is part of the optimization plan regardless of the presence of a WHERE clause in the DELETE statement. As a result, when listing the conditions that must be met to allow roll out and the use of a more efficient way to delete rows, the condition that a "RID index was not chosen by the optimizer to find the rows to be deleted, unless there is no WHERE clause in the DELETE statement" should be removed.

Further, you are able to tell if MDC roll out is in effect because **db2expln** output shows the phrase "Cell Delete". Note that **db2exfmt** does not show this information.

The following text is an update to *Appendix A. DB2 Registry and Environment Variables*:

The description of DB2_MDC_ROLLOUT should be changed such that the condition that a "RID index was not chosen by the optimizer to find the rows to be deleted, unless there is no WHERE clause in the DELETE statement" should be removed from the list.

NEWLOGPATH, MIRRORPATH, and OVERFLOWLOGPATH configuration parameter description clarification

If you update the *newlogpath, mirrorpath, or overflowlogpath* configuration parameter values in a DB2 UDB Enterprise Server Edition environment, the node number will be appended to the path name regardless of the number of nodes on the system. This applies to both single-partition and multi-partition systems in a DB2 UDB Enterprise Server Edition environment.

DB2_COLLECT_TS_REC_INFO default value

The default value for DB2_COLLECT_TS_REC_INFO is *ON*. In DB2 UDB V 8.1 FixPak 7, the default value for the DB2_COLLECT_TS_REC_INFO registry variable was changed to *ON*. The current documentation incorrectly specifies the default for this variable as *OFF*.

The Governor utility

A governor instance consists of a front-end utility and one or more daemons. Each instance of the governor that you start is specific to an instance of the database manager. By default, when you start the governor a governor daemon starts on each partition of a partitioned database. However, you can specify that a daemon be started on a single partition that you want to monitor.

Notes:

1. When the governor is active, its snapshot requests might affect database manager performance. To improve performance, increase the governor wake-up interval to reduce its CPU usage.
2. Governor daemons issue LOCAL snapshots to the local instance while running. Therefore, any rules that contain `setlimit` clauses are applied to output from LOCAL snapshot output rather than the aggregated result from GLOBAL snapshots.

Each governor daemon collects information about the applications that run against the database. The governor daemon then checks this information against the rules that you specify in the governor configuration file for this database.

Choosing a table reorganization method

When considering in-place table reorganization (instead of classic table reorganization), be aware that in-place table reorganization requires more log space.

Because in-place table reorganization logs its activities so that recovery is possible after an unexpected failure, it requires more log space than classic reorganization.

It is possible that in-place reorganization will require log space equal to several times the size of the reorganized table. The amount of required space depends on the number of rows that are moved and the number and size of the indexes on the table.

Recommendation: Choose in-place table reorganization for 24x7 operations with minimal maintenance windows.

An online table reorganization of a DMS table allows the starting of an online backup operation of a table space in which the table resides while the reorganization is happening. There may be lock waits of the reorganization operation during the truncate phase.

Refer to the REORG TABLE syntax descriptions for detailed information about executing these table reorganization methods.

Large page support for FCM memory (AIX 5L 64-bit)

On AIX® 5L 64–bit, the DB2_LARGE_PAGE_MEM registry variable now supports the keyword FCM.

By default, on AIX® 5L™ 64–bit, FCM memory is in the DBMS memory set. However, when the registry variable DB2_FORCE_FCM_BP is enabled, FCM memory is in its own memory set. On AIX 5L™ 64–bit, DB2_LARGE_PAGE_MEM supports the specification of the DBMS memory set. When FCM memory is in the DBMS memory set, and large page support is enabled for that memory set, FCM memory will be in large pages. When FCM memory is in its own memory set, the FCM keyword must be added to the value of the DB2_LARGE_PAGE_MEM registry variable to enable large pages for FCM memory.

DB2_RESOURCE_POLICY registry variable accepts a new element

Starting with DB2 Universal Database™ (UDB) Version 8.2.2 (equivalent to Version 8.1 FixPak 9), the configuration file specified by the DB2_RESOURCE_POLICY registry variable accepts a SCHEDULING_POLICY element. The SCHEDULING_POLICY element can be used on some platforms to select:

- The operating system scheduling policy that is used by the DB2 UDB server

  Note: The capability to set an operating system scheduling policy extends to DB2 for AIX®. It was previously available only for DB2 UDB for Windows® using the DB2NTPRICLASS registry variable.
- The operating system priorities that are used by individual DB2 server agents

The registry variables DB2PRIORITIES and DB2NTPRICLASS can be used separately to control the operating system scheduling policy and set DB2 agent priorities.

However, the specification of a SCHEDULING_POLICY element in the resource policy configuration file provides a single place to specify both the scheduling policy and the associated agent priorities.

Example 1:

Selection of the AIX SCHED_FIFO2 scheduling policy with a priority boost for the db2 log writer and reader processes:

```
<RESOURCE_POLICY>
   <SCHEDULING_POLICY>
      <POLICY_TYPE>SCHED_FIFO2</POLICY_TYPE>
      <PRIORITY_VALUE>60</PRIORITY_VALUE>

      <EDU_PRIORITY>
         <EDU_NAME>db2loggr</EDU_NAME>
         <PRIORITY_VALUE>56</PRIORITY_VALUE>
      </EDU_PRIORITY>

      <EDU_PRIORITY>
         <EDU_NAME>db2loggw</EDU_NAME>
         <PRIORITY_VALUE>56</PRIORITY_VALUE>
      </EDU_PRIORITY>
   </SCHEDULING_POLICY>
</RESOURCE_POLICY>
```

Example 2:

Replacement for DB2NTPRICLASS=H on Windows.

```
<RESOURCE_POLICY>
   <SCHEDULING_POLICY>
      <POLICY_TYPE>HIGH_PRIORITY_CLASS</POLICY_TYPE>
   </SCHEDULING_POLICY>
</RESOURCE_POLICY>
```

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 0x10000000 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.

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 "0x10000000(hex) 268435456(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 0x20000000 to the new value given.

New communication registry variable

The DB2TCP_CLIENT_RCVTIMEOUT registry variable has been added in Version 8.2.

Table 12. Communications variables

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

New performance variable

The DB2_LARGE_PAGE_MEM performance variable has been added in Version 8.2.

Table 13. Performance variables

Variable name	Operating systems	Values
Description		
DB2_LARGE_PAGE_MEM	AIX 5.x 64-bit only Linux	Default=NULL Use * to denote that all applicable memory regions should use large page memory, or a comma-separated list of specific memory regions that should use large page memory. Available regions vary by operating system. On AIX 5.x 64-bit, the following regions can be specified: DB, DBMS, or PRIVATE. On Linux, the following region can be specified: DB.

Table 13. Performance variables (continued)

Variable name	Operating systems	Values
Description		

Large page memory is supported only for DB2 Universal Database (UDB) for AIX 5L, 64-bit Edition, and DB2 UDB for Linux.

The DB2_LARGE_PAGE_MEM registry variable is used to enable large page support when running on AIX 5.x or any Linux architecture with the appropriate kernel support. This registry variable deprecates the DB2_LGPAGE_BP registry variable, which can only be used to enable large-page memory for the database shared memory region. This can now be enabled by setting DB2_LARGE_PAGE_MEM=DB. Any documentation that mentions enabling large pages with the DB2_LGPAGE_BP registry variable can be treated as synonymous with setting DB2_LARGE_PAGE_MEM=DB.

Large page usage is primarily intended to provide performance improvements to high performance computing applications. Memory access intensive applications that use large amounts of virtual memory may obtain performance improvements by using large pages. To enable DB2 UDB to use large pages, you must first configure the operating system to use large pages.

Enabling large private pages will increase DB2 UDB memory usage by a significant amount, as each DB2 UDB agent will consume at least 1 large page (16MB) of physical memory. To enable large pages for agent private memory on 64-bit DB2 UDB for AIX (the DB2_LARGE_PAGE_MEM=PRIVATE setting), the following conditions must be met, in addition to configuring large pages on the operating system:

- The instance owner must possess the CAP_BYPASS_RAC_VMM and CAP_PROPOGATE capabilities.
- The kernel must support interfaces that allow a process to modify its page size at runtime. .

On 64-bit DB2 UDB for AIX, enabling this variable reduces the size of the shared memory segment backing database memory to the minimum requirement. The default is to create a 64 GB segment: see the database shared memory size (*database_memory*) database configuration parameter for more details. This avoids pinning more shared memory in RAM than is likely to be used.

By setting this variable, the ability to dynamically increase the overall database shared memory configuration (for example, to increase the size of buffer pools) will be limited.

On Linux, there is an additional requirement for the availability of the **libcap.so** library. This library must be installed for this option to work. If this option is turned on, and the library is not on the system, DB2 UDB will disable the large kernel pages and continue to function as it would previously.

On Linux, to verify that large kernel pages are available, issue the following command:

```
cat /proc/meminfo
```

If it is available, the following three lines should appear (with different numbers depending on the amount of memory configured on your machine):

```
HugePages_Total:   200
HugePages_Free:    200
Hugepagesize:    16384 KB
```

If you do not see these lines, or if the HugePages_Total is 0, configuration of the operating system or kernel is required.

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 runtime, 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

Range-clustered tables

A range-clustered table cannot be created in a database having more than one partition.

Catalog table space design

When creating a database, three table spaces are defined, including the SYSCATSPACE table space for the system catalog tables. The page size that becomes the default for all table spaces is set when the database is created. If a page size greater than 4096 or 4 K is chosen, the page size for the catalog tables is restricted to a row size that it would have if the catalog table space had a page size of 4 K. The default database page size is stored as an informational database configuration parameter called *pagesize*.

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 14. 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 14. 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– and UNIX–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 15. 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 and UNIX 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 and UNIX systems)`
		`HOLD_CURSOR=F`
		`CHAIN_END=T`
		`SUSPEND_CURSOR=F`
		`TOC=T`

Table 15. 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 and UNIX 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 16. 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

Log record structure clarification

This documentation update applies to Note #3 for *Table 103. Add/Delete/Non-update Long Field Record Log Record Structure* in the section entitled *Long Field Manager Log Records* in the topic entitled *DB2 UDB log records*. The text for Note #3 should read:

3. Long field data length in 512-byte sectors (actual data length is recorded as the first 4 bytes of LF descriptor, which is logged in the subsequent insert/delete/update log record as part of the formatted user data record). The value of this field is always positive. The long field manager never writes log records for zero length long field data that is being inserted, deleted, or updated.

db2Backup API oBackupsize parameter

In DB2 DB2 V8, the oBackupsize parameter from the db2Backup API accounts for full backups and not for delta or incremental backups. The oBackupsize parameter represents the size of the backup image (in MB).

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

Dynamic reconfiguration samples (AIX)

A new sample file, `ibm_db2_sln_upart_smt`, is available with DB2 Universal Database™ (UDB) Version 8.2 FixPak 3 (equivalent to V8.1 Fixpak 10). The following table shows the name and description of the sample file.

Table 17. Dynamic reconfiguration sample scripts

Sample script name	File description
`ibm_db2_sln_upart_smt`	This Korn shell Dynamic Reconfiguration script (DR script) for AIX facilitates the use of Dynamic Logical Partitioning (DLPAR) capabilities provided with AIX Version 5.3 running on POWER5-based pSeries® systems, such as p5 570 and p5 590. This script is similar to the ibm_db2_sln DR script but is optimized to support POWER5™ and AIX V5.3 features such as micropartitioning and SMT. Further information can be found in the DR script itself.

The `ibm_db2_sln_upart_smt` sample script is in the `sqllib/samples/DLPAR` directory on DB2 UDB for AIX.

Linux supported development software

For installation details, refer to "Linux 2.6 kernel installation images" in the *DB2 UDB Version 8.2.2 Release Notes* (*New in this release* section).

The following tables describe the DB2 Linux architecture support as of the release of FixPak 9. Check the preceding validate Web site for updates to this support:

```
http://www.ibm.com/db2/linux/validate
```

Table 18. Linux on Intel® x86 (32-bit)

Distributions	Kernel	Library	Comments
Conectiva Linux Enterprise Edition (CLEE)	2.4.19	glibc 2.2.5	Powered by United Linux 1.0
LINX Rocky Secure Server 2.1	2.4.21	glibc 2.2.5	
Red Flag Advanced Server 4.0	2.4.21-as.2	glibc 2.2.93-5	
Red Flag Function Server 4.0	2.4.20-8smp	glibc 2.2.93-5	
Red Hat Enterprise Linux 2.1 AS/ES/WS	2.4.9-e16	glibc 2.2.4	
Red Hat Enterprise Linux (RHEL) 3 AS/ES/WS	2.4.21-7.EL	glibc-2.3.2-95.3	
Red Hat Enterprise Linux (RHEL) 4	2.6.9	glibc-2.3.3	Also requires the compat-libstdc++-33 package
SCO Linux 4.0	2.4.19	glibc 2.2.5	Powered by United Linux 1.0
SuSE Pro 8.0	2.4.18	glibc 2.2.5	
SuSE Pro 8.1	2.4.19	glibc 2.2.5	

Table 18. Linux on Intel® x86 (32-bit) (continued)

Distributions	Kernel	Library	Comments
SuSE Linux Enterprise Server (SLES) 7	2.4.7	glibc 2.2.2	
SuSE Linux Enterprise Server (SLES) 8	2.4.19	glibc 2.2.5	Validated up to SuSE Service Pack 2 level
SuSE Linux Enterprise Server (SLES) 9	2.6.5	glibc-2.3.3	
Turbolinux 7 Server	2.4.9	glibc 2.2.4	
Turbolinux 8 Server	2.4.18-5	glibc 2.2.5	
Turbolinux Enterprise Server 8	2.4.19	glibc 2.2.5	
United Linux 1.0	2.4.19	glibc 2.2.5	

Table 19. Linux on Intel® x86 (32-bit) non-enterprise distributions (no longer supported by the vendor)

Distributions	Kernel	Library	Comments
Red Hat 7.2	2.4.9-34	glibc 2.2.4	
Red Hat 7.3	2.4.18	glibc 2.2.5	
Red Hat 8.0	2.4.18-14	glibc 2.2.93-5	
SuSE 7.3	2.4.10	glibc 2.2.4	

Table 20. Linux on s/390 and zSeries® (31-bit kernel version supported on s/390; 64-bit on zSeries)

Distributions	Kernel	Library	Comments
Red Hat 7.2	2.4.9-38	glibc 2.2.4	
Red Hat Enterprise Linux (RHEL) 4	2.6.9	glibc-2.3.3	Also requires the compat-libstdc++-33 package
SuSE Linux Enterprise Server (SLES) 7	2.4.7-58	glibc 2.2.4	compat.rpm contains libstdc++ 6.1. Use JDK 1.3.1 SR 1 for Java™
SuSE Linux Enterprise Server (SLES) 8	2.4.19	glibc 2.2.5	Powered by United Linux 1.0
SuSE Linux Enterprise Server (SLES) 9	2.6.5	glibc-2.3.3	
Turbo Linux Enterprise Server (TLES) 8	2.4.19	glibc 2.2.5	Powered by United Linux 1.0
United Linux 1.0	2.4.19	glibc 2.2.5	

Table 21. Linux on x86-64

Distributions	Kernel	Library	Comments
Red Hat Enterprise Linux (RHEL) 3 AS/ES/WS	2.4.21-7.EL	glibc-2.3.2-95.3	
Red Hat Enterprise Linux (RHEL) 4	2.6.9	glibc-2.3.3	Also requires the compat-libstdc++-33 package

Table 21. Linux on x86-64 (continued)

Distributions	Kernel	Library	Comments
SuSE Linux Enterprise Server (SLES) 8.0	2.4.19-SMP	glibc 2.2.5-16	
SuSE Linux Enterprise Server (SLES) 9	2.6.5	glibc-2.3.3	

Table 22. Linux on POWER™ (iSeries and pSeries®)

Distributions	Kernel	Library	Comments
Red Hat Enterprise Linux (RHEL) 3 AS	2.4.21-7.EL	glibc-2.3.2-95.3	
Red Hat Enterprise Linux (RHEL) 4	2.6.9	glibc-2.3.3	Also requires the compat-libstdc++-33 package
SuSE Enterprise Server (SLES) 8	2.4.19-16	glibc 2.2.5	Powered by United Linux 1.0
SuSE Linux Enterprise Server (SLES) 9	2.6.5	glibc-2.3.3	
Turbolinux Enterprise Server 8	2.4.19-16	glibc 2.2.5	Powered by United Linux 1.0
United Linux 1.0	2.4.19	glibc 2.2.5	

Table 23. Linux on IA64

Distributions	Kernel	Library	Comments
Red Hat Enterprise Linux 2.1 AS/ES/WS	2.4.18-e.12smp	glibc	
Red Hat Enterprise Linux (RHEL) 3 AS/ES/WS	2.4.21-7.EL	glibc-2.3.2-95.3	
Red Hat Enterprise Linux (RHEL) 4	2.6.9	glibc-2.3.3	Also requires the compat-libstdc++-33 package
SuSE Linux Enterprise Server (SLES) 8	2.4.19-SMP	glibc 2.2.5	Powered by United Linux 1.0
SuSE Linux Enterprise Server (SLES) 9	2.6.5	glibc-2.3.3	
United Linux 1.0	2.4.19	glibc 2.2.5	

DB2 UDB for Linux 2.6 kernel-based Linux distributions for Intel® x86 supports the following programming languages and compilers:

C GNU/Linux gcc version 3.3

C++ GNU/Linux g++ version 3.3

COBOL
Micro Focus COBOL Server Express Version 2.2 with Service Pack 1

Java™ IBM® Developer Kit and Runtime Environment for Linux, Java™ 2 Technology Edition, Version 1.3.1 and 1.4.1 Service Release 1, 32-bit version

Note: DB2 UDB installs the latest supported version of the developer kit if it is not already installed, unless the DB2 UDB installation is an update of a previous DB2 UDB Version 8 installation. If a previous DB2 UDB Version 8 installation is being updated, the supported developer kit must be manually installed from the CD-ROM.

Perl Perl 5.004_04 or later, DBI 0.93 or later

PHP PHP 4.3.4 or later

REXX Object REXX Interpreter for Linux Version 2.1

A 32-bit instance on **DB2 UDB for 2.6 kernel-based Linux distributions on x86-64** supports the following programming languages and compilers:

C GNU/Linux gcc version 3.3

Note: The ″-m32″ compiler option must be used to generate 32-bit applications or routines (stored procedures and user-defined functions).

C++ GNU/Linux g++ version 3.3

Notes:

1. These versions of the GNU/Linux g++ compiler do not accept integer parameters for some fstream functions. Consult the compiler documentation for more information.
2. The ″-m32″ compiler option must be used to generate 32-bit applications or routines (stored procedures and user-defined functions).

Java IBM® Developer Kit and Runtime Environment for Linux x86, Java 2 Technology Edition, Version 1.3.1 Service Release 4, 32-bit version, and Version 1.4.1 Service Release 1, 32-bit version.

Note: DB2 UDB installs the latest supported version of the developer kit if it is not already installed, unless the DB2 UDB installation is an update of a previous DB2 UDB Version 8 installation. If a previous DB2 UDB Version 8 installation is being updated, the supported developer kit must be manually installed from the CD-ROM.

Perl Perl 5.8

PHP PHP 4.3.4 or later

A 64-bit instance on **DB2 UDB for 2.6 kernel-based distributions for Linux on x86-64** supports the following programming languages and compilers:

C GNU/Linux gcc version 3.3

C++ GNU/Linux g++ version 3.3

Note: These versions of the GNU/Linux g++ compiler do not accept integer parameters for some fstream functions. Consult the compiler documentation for more information.

Java DB2 UDB does not currently support any 64-bit Java Developer Kit for Linux on x86-64.

Perl Perl 5.8

PHP PHP 4.3.4 or later

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 POWER 64-bit)

The compiler option ″–m64″ is required when using gcc/g++ to build C/C++ applications and routines for a 64–bit instance on DB2 Universal Database for Linux on POWER.

The compiler option "–q64" is required when using xlc/xlC to build C/C++ applications and routines for a 64–bit instance on DB2 Universal Database for Linux on POWER.

Compile and link command for Micro Focus COBOL stored procedures (HP-UX)

The compile and link command shown in the DB2 Universal Database Version 8.2 documentation for building stored procedures using Micro Focus COBOL on HP-UX is incorrect. The compile command contained in the actual script `sqllib/samples/cobol_mf/bldrtn` is correct. The compile and link commands are now combined into one single command, using the `-y` option to specify that the desired output is a shared library.

Minimum supported version of Micro Focus COBOL (HP-UX)

The minimum supported version of the Micro Focus COBOL compiler and run time on HP-UX is *Micro Focus Server Express 2.2 - Service Pack 1* plus Fix Pack *Fixpack22.02_14 for HP-UX PA-RISC 11.x (32/64bit)*. This Fix Pack is available from the Micro Focus Support Line Web site at http://supportline.microfocus.com .

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)

Trusted_Connection CLI/ODBC configuration keyword

Keyword description:
Allow a connection to be made with the current authenticated user.

Syntax:
Trusted_Connection=Yes

Note: This keyword has no effect if set in the `db2cli.ini` file. It should instead be provided in the connection string to `SQLDriverConnect()`.

Default setting:
DB2 CLI uses the user ID and password information provided in the connection string to `SQLDriverConnect()`, not the current authenticated user.

Usage notes:

DB2 CLI applications that connect to a database typically connect using the function `SQLDriverConnect()`. One of the input arguments for this function is the *DriverCompletion* value, which determines when a window will be opened. The following are the values of *DriverCompletion* :

- SQL_DRIVER_PROMPT: A dialog is always initiated.
- SQL_DRIVER_COMPLETE: A dialog is only initiated if there is insufficient information in the connection string.
- SQL_DRIVER_COMPLETE_REQUIRED: A dialog is only initiated if there is insufficient information in the connection string. Only mandatory information is requested. The user is prompted for required information only.
- SQL_DRIVER_NOPROMPT: The user is not prompted for any information. A connection is attempted with the information contained in the connection string. If there is not enough information, SQL_ERROR is returned.

Note: More details on *DriverCompletion* can be found in the documentation for `SQLDriverConnect()`.

Some applications, for example, those in a Kerberos environment, might require that a user be able to connect to a DB2 UDB server without providing a user ID or password. If the application uses the SQL_DRIVER_NO_PROMPT option on the `SQLDriverConnect()` call, the connection is attempted without the user authentication. This keyword is then not required.

In the case where a third party application is involved and the prompt level used by the application is something other than SQL_DRIVER_NO_PROMPT, DB2 CLI opens a window to request the missing information. Setting Trusted_Connection to Yes, by providing it to the input connection string for SQLDriverConnect() ("Trusted_Connection=Yes"), causes DB2 CLI to ignore any user ID or password string (including blank strings) from the connection string and ignore the prompt level of the connection function. DB2 CLI uses the current authenticated user to attempt the connection to the database. If the connection attempt fails, the user is prompted for the user ID and password.

This keyword is used only in the connection string for SQLDriverConnect(); setting it in the db2cli.ini file has no effect.

Diagnostic table update for SQLDescribeParam function (CLI)

The SQLDescribeParam() function returns the description of a parameter marker associated with a prepared SQL statement.

The diagnostics table has been updated with SQLSTATE HYC00.

Diagnostics:

Table 24. SQLDescribeParam SQLSTATEs

SQLSTATE	Description	Explanation
HYC00	Driver not capable	The schema function stored procedures are not accessible on the server. Install the schema function stored procedures on the server and ensure they are accessible.

Asynchronous execution of Call Level Interface

The DB2 Call Level Interface (DB2 CLI) can run a subset of functions asynchronously. The DB2 CLI driver returns control to the application after calling the function but before that function has finished executing. The function returns SQL_STILL_EXECUTING each time it is called until it is finished running, at which point it returns a different value (for example, SQL_SUCCESS).

Asynchronous execution is beneficial only on single-threaded operating systems. Applications that run on multithreaded operating systems should execute functions on separate threads. Asynchronous execution is possible for those functions that normally send a request to the server and then wait for a response. Rather than waiting, a function executing asynchronously returns control to the application. The application can then perform other tasks, or return control to the operating system, and use an interrupt to repeatedly poll the function until a return code other than SQL_STILL_EXECUTING is returned.

Support for asynchronous execution of CLI is included in DB2 Universal Database (UDB), starting with Version 8.2 FixPak 1 (equivalent to Version 8.1 FixPak 8). For documentation of this feature, refer to the DB2 UDB Version 7 Information Center at http://publib.boulder.ibm.com/infocenter/db2v7luw/index.jsp. All of the information in the Version 7 documentation applies to Version 8.2 FixPak 1 (equivalent to Version 8.1 FixPak 8) and later. The DB2 Version 8 Information Center does not contain any documentation of this feature.

SQL_ATTR_PING_DB connection attribute

SQL_ATTR_PING_DB (DB2 CLI v8.2):

SQL_ATTR_PING_DB is a 32-bit integer that is used with the `SQLGetConnectAttr()` function to get the network response time of the existing connection between the DB2 UDB client and the DB2 UDB server. Response time is reported in microseconds.

If a connection has previously been established and has been dropped by the database, a value of 0 is reported. If the connection has been closed by the application, then an SQLSTATE of 08003 is reported. This connection attribute can be returned by `SQLGetConnectAttr()`, but it cannot be set by `SQLSetConnectAttr()`. Any attempt to set this attribute will result in an SQLSTATE of HYC00 (Driver not capable).

SQLBindParameter function (CLI)

In the documentation for the SQLBindParameter function, the description in the Input parameter section is incorrect. The correct description is as follows:

Input parameter

A 0 *ColumnSize* means that DB2 CLI will use the maximum length for the SQL type provided as the size of the column or stored procedure parameter. DB2 CLI will perform any necessary conversions using this size.

SQLMoreResults function (CLI)

In the documentation for the SQLMoreResults function, the SQL_ATTR_ROW_ARRAY_SIZE statement attribute is referred to incorrectly. The correct statement attribute is SQL_ATTR_PARAMSET_SIZE. The Usage section should read as follows:

This function is used to return multiple results set in a sequential manner upon the execution of:

- A parameterized query with an array of input parameter values specified with the SQL_ATTR_PARAMSET_SIZE statement attribute and `SQLBindParameter()`

Additional environment attributes

In addition to being CLI connection attributes, the following attributes are also supported as CLI environment attributes:

- SQL_ATTR_INFO_ACCTSTR
- SQL_ATTR_INFO_APPLNAME
- SQL_ATTR_INFO_USERID
- SQL_ATTR_INFO_WRKSTNNAME

For information on these attributes, refer to the CLI connection attributes documentation in the DB2 Information Center or in the *CLI Guide and Reference Volume 2* .

Dynamic scrollable cursors requirement

To perform updates and deletions on rows in a dynamic scrollable cursor's result set, the UPDATE or DELETE statement must include all the columns of at least one unique key in the base table. This can be the primary key or any other unique key.

RetCatalogAsCurrServer CLI/ODBC configuration keyword

Keyword description:

Catalog functions return the CURRENT SERVER value instead of the null value for the catalog columns.

db2cli.ini keyword syntax:

RetCatalogAsCurrServer= **0** | 1

Default setting:

If the target DBMS returns null for the catalog columns, the CURRENT SERVER value will not be substituted.

Usage notes:

If the catalog functions for the target DBMS return a null value for the catalog columns, setting RetCatalogAsCurrServer to 1 causes the DBMS to return the CURRENT SERVER value instead.

- 0 = Catalog functions return the null value for the catalog columns (default).
- 1 = Catalog functions return the CURRENT SERVER value, instead of the null value, for the catalog columns.

For example, assume the catalog function `SQLTables()` returns a result set where the values in the TABLE_CAT column are null values. Setting RetCatalogAsCurrServer to 1 causes the DBMS to return the CURRENT SERVER value in the TABLE_CAT column.

Note: This keyword is supported by DB2 UDB for Linux, UNIX® and Windows® Version 8.2 FixPak 3 (equivalent to Version 8.1 FixPak 10) and later.

ReceiveTimeout CLI/ODBC configuration keyword

Keyword description:

Specify the time in seconds to wait for a reply from the server on an established connection before terminating the attempt and generating a communication timeout error.

db2cli.ini keyword syntax:

ReceiveTimeout = **0** | **1** | **2** | ... | **32767**

Default setting:

The client waits indefinitely for a reply from the server on an established connection.

Equivalent connection attribute:

SQL_ATTR_RECEIVE_TIMEOUT

Usage notes:

The default value of 0 indicates that the client waits indefinitely for a reply. The receive timeout has no effect during connection establishment; it is only supported for TCP/IP and is ignored for any other protocol.

SQL_ATTR_RECEIVE_TIMEOUT connection attribute

SQL_ATTR_RECEIVE_TIMEOUT (DB2 CLI v8)

A 32–bit integer value that is the number of seconds a client waits for a reply from a server on an established connection before terminating the attempt and generating a communication timeout error. The default value of 0 indicates the client waits indefinitely for a reply. The receive timeout has no effect during connection establishment; it is only supported for TCP/IP, and is ignored for any other protocol. Supported values are integers from 0 to 32767.

Reopt CLI/ODBC configuration keyword

Keyword description:
Enable query optimization or reoptimization of SQL statements that have special registers or parameter markers.

db2cli.ini keyword syntax:
Reopt = **2** | **3** | **4**

Default setting:
No query optimization occurs at query execution time. The default estimates chosen by the compiler are used for special registers or parameter markers.

Equivalent statement and connection attribute:
SQL_ATTR_REOPT

Usage notes:
Optimization occurs by using the values available at query execution time for the special registers or parameter markers instead of the default estimates that are chosen by the compiler. The valid values of the keyword are:

- 2 = SQL_REOPT_NONE. This is the default. No query optimization occurs at query execution time. The default estimates chosen by the compiler are used for the special registers or parameter markers. The default "NULLID " package set is used to execute dynamic SQL statements.
- 3 = SQL_REOPT_ONCE. Query optimization occurs once at query execution time, when the query is executed for the first time. The "NULLIDR1" package set, which is bound with the REOPT ONCE bind option, is used.
- 4 = SQL_REOPT_ALWAYS. Query optimization or reoptimization occurs at query execution time every time the query is executed. The "NULLIDRA" package set, which is bound with the REOPT ALWAYS bind option, is used.

The "NULLIDR1" and "NULLIDRA" are reserved package set names, and when used, REOPT ONCE and REOPT ALWAYS are implied respectively. These package sets have to be explicitly created with the following commands:

```
db2 bind db2clipk.bnd collection NULLIDR1
db2 bind db2clipk.bnd collection NULLIDRA
```

If both the Reopt and CurrentPackageSet keywords are specified, CurrentPackageSet takes precedence.

SQL_ATTR_REOPT statement and connection attribute

SQL_ATTR_REOPT (DB2 CLI v8)
A 32–bit integer value that enables query optimization for SQL statements that contain special registers or parameter markers. Optimization occurs by using the values available at query execution time for special registers or parameter markers, instead of the default estimates that are chosen by the compiler. The valid values of the attribute are:

- 2 = SQL_REOPT_NONE. This is the default. No query optimization occurs at query execution time. The default estimates chosen by the compiler are used for the special registers or parameter markers. The default "NULLID " package set is used to execute dynamic SQL statements.
- 3 = SQL_REOPT_ONCE. Query optimization occurs once at query execution time, when the query is executed for the first time. The "NULLIDR1" package set, which is bound with the REOPT ONCE bind option, is used.
- 4 = SQL_REOPT_ALWAYS. Query optimization or reoptimization occurs at query execution time every time the query is executed. The "NULLIDRA" package set, which is bound with the REOPT ALWAYS bind option, is used.

The "NULLIDR1" and "NULLIDRA" are reserved package set names, and when used, REOPT ONCE and REOPT ALWAYS are implied respectively. These package sets have to be explicitly created with these commands:

```
db2 bind db2clipk.bnd collection NULLIDR1
db2 bind db2clipk.bnd collection NULLIDRA
```

SQL_ATTR_REOPT and SQL_ATTR_CURRENT_PACKAGE_SET are mutually exclusive, therefore, if one is set, the other is not allowed.

CurrentPackageSet CLI/ODBC configuration keyword

Keyword description:
Issues the SET CURRENT PACKAGESET statement after every connection.

db2cli.ini keyword syntax:
CurrentPackageSet = *schema name*

Default setting:
The clause is not appended.

Equivalent connection attribute:
SQL_ATTR_CURRENT_PACKAGE_SET

Usage notes:

This option issues the SET CURRENT PACKAGESET SQL statement with the CurrentPackageSet value after every connection to a database. By default this clause is not appended.

The SET CURRENT PACKAGESET SQL statement sets the schema name (collection identifier) that is used to select the package to use for subsequent SQL statements.

CLI/ODBC applications issue dynamic SQL statements. Using this option you can control the privileges used to run these statements:

- Choose a schema to use when running SQL statements from CLI/ODBC applications.
- Ensure the objects in the schema have the desired privileges and then rebind accordingly.
- Set the CurrentPackageSet option to this schema.

The SQL statements from the CLI/ODBC applications will now run under the specified schema and use the privileges defined there.

The following package set names are reserved: "NULLID ", "NULLIDR1", "NULLIDRA".

If both the Reopt and CurrentPackageSet keywords are specified, CurrentPackageSet takes precedence.

SQL_ATTR_CURRENT_PACKAGE_SET connection attribute

SQL_ATTR_CURRENT_PACKAGE_SET (DB2 CLI v5)

A null-terminated character string that indicates the schema name (collection identifier) that is used to select the package for subsequent SQL statements. Setting this attribute causes the SET CURRENT PACKAGESET SQL statement to be issued. If this attribute is set before a connection, the SET CURRENT PACKAGESET SQL statement will be issued at connection time.

CLI/ODBC applications issue dynamic SQL statements. Using this connection attribute, you can control the privileges used to run these statements:

- Choose a schema to use when running SQL statements from CLI/ODBC applications.
- Ensure the objects in the schema have the desired privileges and then rebind accordingly. This typically means binding the CLI packages (sqllib/bnd/db2cli.lst) using the COLLECTION <collid> option. Refer to the BIND command for further details.
- Set the CURRENTPACKAGESET option to this schema.

The SQL statements from the CLI/ODBC applications will now run under the specified schema and use the privileges defined there.

Setting the CLI/ODBC configuration keyword CURRENTPACKAGESET is an alternative method of specifying the schema name.

The following package set names are reserved: "NULLID ", "NULLIDR1", "NULLIDRA".

SQL_ATTR_REOPT and SQL_ATTR_CURRENT_PACKAGE_SET are mutually exclusive, therefore, if one is set, the other is not allowed.

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 25 on page 104.

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 25 on page 104 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 25 on page 104) is returned to the client application
- 2 - (default) describe information categorized in level 2 (see Table 25 on page 104) is returned to the client application
- 3 - describe information categorized in level 3 (see Table 25 on page 104) 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 , UNIX, 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 25. 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

DB2 Universal JDBC Driver Type 4 connectivity to DB2 for VM/VSE is not supported

The DB2 Universal JDBC Driver does not support type 4 connectivity to DB2 for VM/VSE databases. The topics titled ″Setting up the Windows Java environment″ and “Installing the DB2 Universal JDBC Driver” in the Application Development Guide: Programming Client Applications, and the DB2 UDB Information Center incorrectly state that the DB2 Universal JDBC Driver supports type 4 connectivity to DB2 for VM/VSE databases.

DB2 Universal JDBC Driver connection concentrator and Sysplex workload balancing

Java applications that use DB2 Universal JDBC Driver type 4 connectivity to access DB2 UDB for z/OS® servers can take advantage of its connection concentrator and Sysplex workload balancing functions.

These functions are similar to the connection concentrator and Sysplex workload balancing functions of DB2 Connect.

The DB2 Universal JDBC Driver connection concentrator can reduce the resources that DB2 UDB for z/OS® database servers require to support large numbers of client applications by letting many connection objects use the same physical connection, which reduces the total number of physical connections to the database server.

DB2 Universal JDBC Driver Sysplex workload balancing can improve availability of a data-sharing group because the driver gets frequent status information about the members of a data-sharing group. The driver uses this information to determine the data-sharing member to which the next transaction should be routed. With Sysplex workload balancing, the DB2 UDB for z/OS server and Workload Manager for z/OS (WLM) ensure that work is distributed efficiently among members of the data-sharing group and that work is transferred to another member of a data-sharing group if one member has a failure.

The DB2 Universal JDBC Driver uses transport objects and a global transport objects pool to support the connection concentrator and Sysplex workload balancing. There is one transport object for each physical connection to the database server. When you enable the connection concentrator and Sysplex workload balancing, you set the maximum number of physical connections to the database server at any point in time by setting the maximum number of transport objects.

At the driver level, you set limits on the number of transport objects using DB2 Universal JDBC Driver configuration properties.

At the connection level, you enable and disable the DB2 Universal JDBC Driver connection concentrator and Sysplex workload balancing and set limits on the number of transport objects using DataSource properties.

You can monitor the global transport objects pool in either of the following ways:

- Using traces that you start using DB2 Universal JDBC Driver configuration properties
- Using an application programming interface

DB2 Universal JDBC Driver configuration properties for connection concentrator and Sysplex workload balancing

Each of the following configuration properties is used for connection concentrator and Sysplex workload balancing

db2.jcc.dumpPool

Specifies the types of statistics that are written for global transport pool events, in addition to the summary statistics that are written. The global transport pool is used for the connection concentrator and Sysplex workload balancing.

The data type of db2.jcc.dumpPool is integer (int.). The db2.jcc.dumpPoolStatisticsOnSchedule and db2.jcc.dumpPoolStatisticsOnScheduleFile configuration properties must also be set for writing statistics before any statistics are written.

You can specify one or more of the following types of statistics with the db2.jcc.dumpPool property:

- `DUMP_REMOVE_OBJECT` (hexadecimal: X'01', decimal: 1)
- `DUMP_GET_OBJECT` (hexadecimal: X'02', decimal: 2)
- `DUMP_WAIT_OBJECT` (hexadecimal: X'04', decimal: 4)
- `DUMP_SET_AVAILABLE_OBJECT` (hexadecimal: X'08', decimal: 8)
- `DUMP_CREATE_OBJECT` (hexadecimal: X'10', decimal: 16)
- `DUMP_SYSPLEX_MSG` (hexadecimal: X'20', decimal: 32)
- `DUMP_POOL_ERROR` (hexadecimal: X'80', decimal: 128)

To trace more than one type of event, add the values for the types of events that you want to trace. For example, suppose that you want to trace DUMP_GET_OBJECT and DUMP_CREATE_OBJECT events. The numeric equivalents of these values are 2 and 16, so you specify 18 for the db2.jcc.dumpPool value.

The default is 0, which means that only summary statistics for the global transport pool are written.

db2.jcc.dumpPoolStatisticsOnSchedule

Specifies how often, in seconds, global transport pool statistics are written to the file that is specified by the db2.jcc.dumpPoolStatisticsOnScheduleFile configuration property. The global transport pool is used for the connection concentrator and Sysplex workload balancing.

The default is -1, which means that global transport pool statistics are not written.

db2.jcc.dumpPoolStatisticsOnScheduleFile

Specifies the name of the file to which global transport pool statistics are written. The global transport pool is used for the connection concentrator and Sysplex workload balancing.

If the db2.jcc.dumpPoolStatisticsOnScheduleFile configuration property is not specified, global transport pool statistics are not written.

db2.jcc.maxTransportObjectIdleTime

Specifies the amount of time, in seconds, that an unused transport object stays in a global transport object pool before it can be deleted from the pool. Transport objects are used for the connection concentrator and Sysplex workload balancing.

The default value for the db2.jcc.maxTransportObjectIdleTime configuration property is 60. Setting db2.jcc.maxTransportObjectIdleTime to a value less than 0 causes unused transport objects to be deleted from the pool immediately. This action is **not** recommended because it can cause severe performance degradation.

db2.jcc.maxTransportObjectWaitTime

Specifies the maximum amount of time, in seconds, that an application waits for a transport object if the db2.jcc.maxTransportObjects value has been reached. Transport objects are used for the connection concentrator and Sysplex workload balancing. When an application waits for longer than the db2.jcc.maxTransportObjectWaitTime value, the global transport object pool throws an SQLException.

The default value for the db2.jcc.maxTransportObjectWaitTime configuration property is -1. Any negative value means that applications wait forever.

db2.jcc.maxTransportObjects

Specifies the upper limit for the number of transport objects in a global transport object pool for the connection concentrator and Sysplex workload balancing. When the number of transport objects in the pool reaches the db2.jcc.maxTransportObjects value, transport objects that have not been used for longer than the db2.jcc.maxTransportObjectIdleTime value are deleted from the pool.

The default value for the db2.jcc.maxTransportObjects configuration property is -1, which means that there is no limit to the number of transport objects in the global transport object pool.

db2.jcc.minTransportObjects

Specifies the lower limit for the number of transport objects in a global transport object pool for the connection concentrator and Sysplex workload balancing. When a JVM is created, there are no transport objects in the pool. Transport objects are added to the pool as they are needed. After the db2.jcc.minTransportObjects value is reached, the number of transport objects in the global transport object pool never goes below the db2.jcc.minTransportObjects value for the lifetime of that JVM.

The default value for the db2.jcc.minTransportObjects configuration property is 0. Any value less than or equal to 0 means that the global transport object pool can become empty.

DB2 Universal JDBC Driver DataSource properties for connection concentrator and Sysplex workload balancing

Each of the following DB2 Universal JDBC Driver DataSource properties is used for connection concentrator and Sysplex workload balancing

enableConnectionConcentrator

Indicates whether the connection concentrator function of the DB2 Universal JDBC Driver is enabled. The connection concentrator function is available only for connections to DB2 UDB for z/OS servers.

The data type of the enableConnectionConcentrator property is boolean. The default is `false`. However, if enableSysplexWLB is set to `true`, the default is `true`.

enableSysplexWLB

Indicates whether the Sysplex workload balancing function of the DB2 Universal JDBC Driver is enabled. The Sysplex workload balancing function is available only for connections to DB2 UDB for z/OS servers.

The data type of the enableSysplexWLB property is boolean. The default is `false`. However, if enableSysplexWLB is set to `true`, enableConnectionConcentrator is set to true by default.

maxTransportObjects

Specifies the maximum number of transport objects that can be used for all connections with the associated `DataSource` object. Transport objects are used for the connection concentrator and Sysplex workload balancing. The maxTransportObjects value is ignored if the enableConnectionConcentrator or enableSysplexWLB properties are not set to enable the use of the connection concentrator or Sysplex workload balancing.

The data type of this property is integer (int.).

If the maxTransportObjects value has not been reached and a transport object is not available in the global transport objects pool, the pool creates a new transport object. If the maxTransportObjects value has been reached, the application waits for the amount of time that is specified by the db2.jcc.maxTransportObjectWaitTime configuration property. After that amount of time has elapsed, if there is still no available transport object in the pool, the pool throws an SQLException.

The maxTransportObjects property does **not** override the db2.jcc.maxTransportObjects configuration property. The maxTransportObjects property has no effect on connections from other `DataSource` objects. If the maxTransportObjects value is larger than the db2.jcc.maxTransportObjects value, maxTransportObjects does not increase the db2.jcc.maxTransportObjects value.

The default value for the maxTransportObjects property is -1, which means that the number of transport objects for the `DataSource` is limited only by the db2.jcc.maxTransportObjects value for the driver..

Example of enabling the DB2 Universal JDBC Driver connection concentrator and Sysplex workload balancing functions in WebSphere Application Server

The following procedure is an example of enabling the DB2 Universal JDBC Driver connection concentrator and Sysplex workload balancing functions with WebSphere® Application Server.

Prerequisites:

Server requirements:

- WLM for z/OS
- DB2 UDB for OS/390® and z/OS, Version 7 or later

Client requirements:

- DB2 Universal JDBC Driver at the FixPak 10 level
- WebSphere® Application Server, Version 5.1 or later

Procedure:

To enable the DB2 Universal JDBC Driver connection concentrator and Sysplex workload balancing functions with WebSphere Application Server:

1. Verify that the DB2 Universal JDBC Driver is at the correct level to support the connection concentrator and Sysplex workload balancing functions by issuing the following command in the command line processor on z/OS, or in System Services on UNIX® :

   ```
   java com.ibm.db2.jcc.DB2Jcc -version
   ```

   Find a line in the output like this:

   ```
   [ibm][db2][jcc] Driver: IBM DB2 JDBC Universal Driver Architecture n n
   ```

   n should be 2.7 or later.
2. Set DB2 Universal JDBC Driver configuration properties to enable the connection concentrator or Sysplex workload balancing for all `DataSource` instances that are created under the driver.

   Set the configuration properties in a DB2JccConfiguration.properties file.

   a. Create a DB2JccConfiguration.properties file or edit the existing DB2JccConfiguration.properties file.

   b. Set the following configuration properties:
      - db2.jcc.minTransportObjects
      - db2.jcc.maxTransportObjects
      - db2.jcc.maxTransportObjectWaitTime
      - db2.jcc.dumpPool
      - db2.jcc.dumpPoolStatisticsOnScheduleFile

      Start with settings similar to these:

      ```
      db2.jcc.minTransportObjects=0
      db2.jcc.maxTransportObjects=1500
      db2.jcc.maxTransportObjectWaitTime=-1
      db2.jcc.dumpPool=0
      db2.jcc.dumpPoolStatisticsOnScheduleFile=
        /home/WAS/logs/srv1/poolstats
      ```

   c. Add the directory path for DB2JccConfiguration.properties to the WebSphere Application Server DB2 Universal JDBC Driver classpath.

3. Set DB2 Universal JDBC Driver data source properties to enable the connection concentrator or Sysplex workload balancing functions.

   In the WebSphere Application Server administrative console, set the following properties for the data source that your application uses to connect to the database server:
   - enableSysplexWLB
   - enableConnectionConcentrator
   - maxTransportObjects

   Assume that you want the connection concentrator function as well as the Sysplex workload balancing function. Start with settings similar to these:

Table 26. Example of data source property settings for DB2 Universal JDBC Driver connection concentrator and Sysplex workload balancing functions

Property	Setting
enableSysplexWLB	true[1]
maxTransportObjects	100

Notes:

1. The enableConnectionConcentrator property is set to true by default because the enableSysplexWLB property is set to true.

4. Restart WebSphere Application Server.

Methods for monitoring DB2 Universal JDBC Driver connection concentrator and Sysplex workload balancing functions

To monitor the DB2 Universal JDBC Driver connection concentrator and Sysplex workload balancing functions, you need to monitor the global transport objects pool. You can monitor the global transport objects pool in either of the following ways:

- Using traces that you start by setting DB2 Universal JDBC Driver configuration properties
- Using an application programming interface

Configuration properties for monitoring the global transport objects pool: The db2.jcc.dumpPool, db2.jcc.dumpPoolStatisticsOnSchedule, and db2.jcc.dumpPoolStatisticsOnScheduleFile configuration properties control tracing of the global transport objects pool.

For example, the following set of configuration property settings cause Sysplex error messages and dump pool error messages to be written every 60 seconds to a file named /home/WAS/logs/srv1/poolstats:

```
db2.jcc.dumpPool=DUMP_SYSPLEX_MSG|DUMP_POOL_ERROR
db2.jcc.dumpPoolStatisticsOnSchedule=60
db2.jcc.dumpPoolStatisticsOnScheduleFile=/home/WAS/logs/srv1/poolstats
```

An entry in the pool statistics file looks like this:

```
time Scheduled PoolStatistics npr:2575 nsr:2575 lwroc:439
hwroc:1764 coc:372 aooc:362 rmoc:362 nbr:2872 tbt:857520 tpo:10
```

The meanings of the fields are:

npr The total number of requests that the DB2 Universal JDBC Driver has made to the pool since the pool was created.

nsr The number of successful requests that the DB2 Universal JDBC Driver has made to the pool since the pool was created. A successful request means that the pool returned an object.

lwroc The number of objects that were reused but were not in the pool. This can happen if a connection object releases a transport object at a transaction boundary. If the connection object needs a transport object later, and the original transport object has not been used by any other connection object, the connection object can use that transport object.

hwroc The number of objects that were reused from the pool.

coc The number of objects that the DB2 Universal JDBC Driver created since the pool was created.

aooc The number of objects that exceeded the idle time that was specified by the db2.jcc.maxTransportObjectIdleTime configuration property and were deleted from the pool.

rmoc The number of objects that have been deleted from the pool since the pool was created.

nbr The number of requests that the DB2 Universal JDBC Driver made to the pool that the pool blocked because the pool had reached its maximum capacity. A blocked request might be successful if an object is returned to the pool before the db2.jcc.maxTransportObjectWaitTime configuration value is exceeded and an exception is thrown.

tbt The total time in milliseconds for requests that were blocked by the pool. This time can be much larger than the elapsed execution time of the application if the application uses multiple threads.

tpo The number of objects that are currently in the pool.

Application programming interfaces for monitoring the global transport objects pool: You can write applications to gather statistics on the global transport objects pool. Those applications create objects in the `DB2PoolMonitor` class and invoke methods to retrieve information about the pool.

For example, the following code creates an object for monitoring the global transport objects pool:

```
import com.ibm.db2.jcc.DB2PoolMonitor;
DB2PoolMonitor transportObjectPoolMonitor =
 DB2PoolMonitor.getPoolMonitor (DB2PoolMonitor.TRANSPORT_OBJECT);
```

After you create the DB2PoolMonitor object, you can use the following methods to monitor the global transport objects pool.

getMonitorVersion

Format:

```
public int getMonitorVersion()
```

Retrieves the version of the DB2PoolMonitor class that is shipped with the DB2 Universal JDBC Driver.

totalRequestsToPool

Format:

```
public int totalRequestsToPool()
```

Retrieves the total number of requests that the DB2 Universal JDBC Driver has made to the pool since the pool was created.

successfullRequestsFromPool

Format:

```
public int successfullRequestsFromPool()
```

Retrieves the number of successful requests that the DB2 Universal JDBC Driver has made to the pool since the pool was created. A successful request means that the pool returned an object.

numberOfRequestsBlocked

Format:

```
public int numberOfRequestsBlocked()
```

Retrieves the number of requests that the DB2 Universal JDBC Driver made to the pool that the pool blocked because the pool had reached its maximum capacity. A blocked request might be successful if an object is returned to the pool before the db2.jcc.maxTransportObjectWaitTime configuration value is exceeded and an exception is thrown.

totalTimeBlocked

Format:

```
public long totalTimeBlocked()
```

Retrieves the total time in milliseconds for requests that were blocked by the pool. This time can be much larger than the elapsed execution time of the application if the application uses multiple threads.

lightWeightReusedObjectCount

Format:

```
public int lightWeightReusedObjectCount()
```

Retrieves the number of objects that were reused but were not in the pool. This can happen if a connection object releases a transport object at a transaction boundary. If the connection object needs a transport object later, and the original transport object has not been used by any other connection object, the connection object can use that transport object.

heavyWeightReusedObjectCount

Format:

```
public int heavyWeightReusedObjectCount()
```

Retrieves the number of objects that were reused from the pool.

createdObjectCount

Format:

```
public int createdObjectCount()
```

Retrieves the number of objects that the DB2 Universal JDBC Driver created since the pool was created.

agedOutObjectCount

Format:

```
public int agedOutObjectCount()
```

Retrieves the number of objects that exceeded the idle time that was specified by the db2.jcc.maxTransportObjectIdleTime configuration property and were deleted from the pool.

removedObjectCount

Format:

```
public int removedObjectCount()
```

Retrieves the number of objects that have been deleted from the pool since the pool was created.

totalPoolObjects

Format:

```
public int totalPoolObjects()
```

The number of objects that are currently in the pool.

OleDbReportIsLongForLongTypes CLI/ODBC configuration keyword

The OleDbReportIsLongForLongTypes keyword is supported by the following database servers:

- DB2 UDB for z/OS Version 6 with PTF UQ93891
- DB2 UDB for z/OS Version 7 with PTF UQ93889
- DB2 UDB for z/OS Version 8 with PTF UQ93890
- DB2 UDB for Linux, UNIX and Windows Version 8.2 (equivalent to Version 8.1 FixPak 7) and later

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

The OleDbSQLColumnsSortByOrdinal keyword is supported by the following database servers:

- DB2 UDB for z/OS Version 6 with PTF UQ93891
- DB2 UDB for z/OS Version 7 with PTF UQ93889
- DB2 UDB for z/OS Version 8 with PTF UQ93890
- DB2 UDB for Linux, UNIX and Windows Version 8.2 (equivalent to Version 8.1 FixPak 7) and later

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 27. 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 28. 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 29. 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:

```
►►──java──com.ibm.db2.jcc.DB2Binder──-url jdbc:db2://server──┬────────┬──/database──-user user-ID──────►
                                                             └─:──port─┘

►──-password password──┬──────────────┬──┬──────────────────────────────┬─────────────────────────────►
                       └─-size integer─┘  └─-collection collection-name─┘

►──┬─────────────────────────────────┬──┬───────┬───────────────────────────────────────────────────►◄
   │           ┌─,──────────────┐    │  └─-help─┘
   │           ▼                │    │
   └─-tracelevel ──trace-option─┴────┘
```

Rerouting DB2 Universal JDBC driver clients

The automatic client reroute feature in DB2 Universal Database (UDB) for Linux, UNIX, 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 and UNIX)

.so is now accepted as a file name extension for user-written security plug–in libraries on all Linux and UNIX 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 and UNIX platforms, *.so* is the only supported file name extension for security plug–in libraries.

Restrictions on security plug–in libraries

On AIX, security plug–in libraries can have a file name extension of *.a* or *.so*. The mechanism used to load the plug–in library depends on which extension is used:

Plug–in libraries with a file name extension of *.a*

Plug–in libraries with file name extensions of *.a* are assumed to be archives containing shared object members. These members must be named *shr.o* (32–bit) or *shr64.o* (64–bit). A single archive can contain both the 32–bit and 64–bit members, allowing it to be deployed on both types of platforms.

For example, to build a 32–bit archive style plug–in library:

```
xlc_r -qmkshrobj -o shr.o MyPlugin.c -bE:MyPlugin.exp
ar rv MyPlugin.a shr.o
```

Plug–in libraries with a file name extension of *.so*

Plug–in libraries with file name extensions of *.so* are assumed to be dynamically loadable shared objects. Such an object is either 32–bit or 64–bit, depending on the compiler and linker options used when it was built. For example, to build a 32–bit plug–in library:

```
xlc_r -qmkshrobj -o MyPlugin.so MyPlugin.c -bE:MyPlugin.exp
```

On all platforms other than AIX, security plug–in libraries are always assumed to be dynamically loadable shared objects.

GSS-API plug-in support for DB2 Universal JDBC Driver

With the release of DB2 UDB Version 8.2 for Linux, UNIX, Windows, you can create your own authentication mechanisms in the form of plug-ins (loadable libraries). The DB2 UDB engine loads and accesses these plug-ins to perform user authentication. In order to support customer applications written in Java, DB2 Universal JDBC Driver provides security plug-in support in DB2 UDB V8.2, FixPak 4.

For Java applications using the DB2 Universal JDBC Driver to perform plug-in authentication, users need to implement their own plug-in by extending the abstract class com.ibm.db2.jcc.DB2JCCPlugin and setting the following properties:

- *User ID* and *password*: these are the user credentials to be used for authentication
- *Plug-in name*: this specifies the name of the plug-in module on the server side
- *Security mechanism*: this must be set to plug-in security
- *Plug-in*: this specifies the plug-in module on the client application side

Note the following example:

```
java.util.Properties properties = new java.util.Properties();
properties.put("user", "db2admin");
properties.put("password", "admindb2");
properties.put("pluginName", "gssapi_simple");
properties.put("securityMechanism",
new String(""+com.ibm.db2.jcc.DB2BaseDataSource.PLUGIN_SECURITY+""));
properties.put("plugin", new JCCSimpleGSSPlugin());
Connection con = java.sql.DriverManager.getConnection(url, properties);
```

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

Multiple result set common language runtime (CLR) procedures

The maximum number of result sets that can be returned by a common language runtime (CLR) procedure is limited. The limit is determined by the maximum number of DB2DataReader objects that the DB2 .NET data provider can simultaneously support having open within a connection. Concurrent active data reader support enables multiple DB2DataReader objects to be opened within a connection. Therefore multiple result sets can be returned from a CLR procedure.

Common language runtime (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 runtime. 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 runtime. At runtime, 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 runtime 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 runtime (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 runtime 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

db2licm - License Management Tool Command parameter clarification

This documentation update applies to the *-p* parameter of the **db2licm** command. The keyword *MEASURED* is no longer valid and is ignored by DB2.

RESTORE DATABASE command TSM example

The example for TSM (example 6) is missing single quotes around the TSM options and should state:

- Specifying a delimited string:

```
db2 restore db sample use TSM options "'-fromnode bar -fromowner dmcinnis'"
```

db2demigdbd usage examples

The following examples of using **db2demigdbd** are shown organized by operating system.

To demigrate system database directory files on UNIX and Linux:

```
 db2demigdbd $HOME/sqllib 1
```

To demigrate the local system database directory files on UNIX and Linux:

```
 db2demigdbd  ~/user/NODE0000  1
```

To demigrate the system and all local database directory files in the instance on UNIX and Linux:

```
 db2demigdbd  $HOME/sqllib  2
```

To demigrate system database directory files on Windows:

```
 db2demigdbd  d:\sqllib\db2  1
```

where *db2* is the current instance.

To demigrate the local system database directory files on Windows:

```
db2demigdbd  d:\db2\NODE0000  1
```

To demigrate the system and all local database directory files in the instance on Windows:

```
db2demigdbd  d:\sqllib\db2 2
```

Usage Notes:

Once you have run the **db2updv8** command to update the system catalogs to the current Version 8 level, falling back to V 8.1 is not supported.

Additional information:

For an example of how to run the command in a DPF (Database Partitioning Feature) environment, see the technote located at http://www.ibm.com/support/docview.wss?rs=71&uid=swg21217949

db2ckbkp command usage note correction

The ′-p′ option displays the number of pages of each object type. However, the output for the **db2ckbkp** command with the ′-p′ option does not show the number of pages for all different object types if the backup was done for DMS table spaces data. In this case, it only displays the total of all pages as *SQLUDMSTABLESPACEDATA*. The value of zero is displayed for *SQLUDMSLOBDATA* and *SQLUDMSLONGDATA* for DMS table spaces.

Here is a sample output section of **db2ckbkp -p** with the backup done for DMS table spaces data:

```
SQLUDMSTABLESPACEDATA (0c):  87
SQLUDMSLOBDATA (0e):          0
SQLUDMSLONGDATA (0f):         0
```

Naming conventions for database objects and for IDs

The following conventions that apply when naming database manager objects, such as databases and tables.

- Character strings that represent names of database manager objects can contain any of the following characters: a-z, A-Z, 0-9, @, #, and $.
- Unless otherwise noted, you can type names in lowercase letters; however, the database manager processes them as if they were uppercase.

  The exception to this convention is character strings that represent names under the Systems Network Architecture (SNA). Many values are case sensitive, such as logical unit names (partner_lu and local_lu). The name must be entered exactly as it appears in the SNA definitions that correspond to those terms.
- A database name or database alias is a unique character string containing from one to eight letters, numbers, or keyboard characters from the set described in the preceding information.

  Databases are cataloged in the system and in local database directories by their aliases in one field, and by their original name in another. For most functions, the database manager uses the name entered in the alias field of the database directories. The exceptions are CHANGE DATABASE COMMENT and CREATE DATABASE, where a directory path must be specified.

- The name or the alias name of a table or a view is an SQL identifier that is a unique character string 1 to 128 characters in length. Column names can be 1 to 30 characters in length.

  A fully qualified table name consists of the *schema.tablename*. The schema is the unique user ID under which the table was created. The schema name for a declared temporary table must be SESSION.
- Local aliases for remote nodes that are to be cataloged in the node directory cannot exceed eight characters in length.
- The first character in the string must be an alphabetic character, @, #, or $; it cannot be a number or the letter sequences SYS, DBM, or IBM.

The following conventions apply when naming user IDs and authentication IDs.

- Character strings that represent names of database manager objects can contain any of the following: a-z, A-Z, 0-9, @, #, and $.
- User IDs and groups may also contain any of the following additional characters when supported by the security plug-in: _, !, %, (,), {, }, –, ., ^.
- User IDs and groups containing any of the following characters must be delimited with quotations when entered through the command line processor: !, %, (,), {, }, –, ., ^,
- The first character in the string must be an alphabetic character, @, #, or $; it cannot be a number or the letter sequences SYS, DBM, or IBM.
- Authentication IDs cannot exceed 30 characters on Windows 32-bit operating systems and 8 characters on all other operating systems.
- Group IDs cannot exceed 30 characters in length.

REORGCHK command

The index statistics formulas used within the REORGCHK command have been revised. The new formulas and their explanations are:

- Formula F5:

  ```
  100 * (KEYS * (ISIZE + LEAF_REC_OVERHEAD) + (CARD - KEYS)
              * DUPKEYSIZE )
      / ((NLEAF - NUM EMPTY LEAFS - 1) *
        (INDEXPAGESIZE - 96) > MIN(50, (100 - PCTFREE))
  ```

  where LEAF_REC_OVERHEAD = 9 and DUPKEYSIZE = 5.

  A reorganization is recommended if there is more than 50 percent free space in the index, or more than PCTFREE% free space in the index if PCTFREE is greater than 50. This formula is checked only if the value of NLEAF – NUM EMPTY LEAFS – 1 is greater than 0. (One is subtracted from the value of NLEAF because the last leaf page allocated is usually not filled.)
- Formula F6:

  ```
  (100 - PCTFREE) *
       [ Floor((100 - min(10, PCTFREE)) / 100 * (INDEXPAGESIZE - 96)
               / (ISIZE + NONLEAF_REC_OVERHEAD)) ** (NLEVELS - 2)]
       * (INDEXPAGESIZE - 96) /
         (KEYS * (ISIZE + LEAF_REC_OVERHEAD)
               + (CARD - KEYS) * DUPKEYSIZE) < 100
  ```

  where NONLEAF_REC_OVERHEAD = 12.

  To determine if recreating the index would result in a tree having fewer levels. This formula checks the ratio between the amount of space in an index tree that has one less level than the current tree, and the amount of space needed. If a tree with one less level could be created and still leave PCTFREE available, then a reorganization is recommended. The actual number of index entries should be more than 90% (or 100 – PCTFREE) of the number of entries an NLEVELS – 1 index tree can handle (only checked if NLEVELS > 1).

Note: All formulas are based on statistics. The statistics and the results from these formulas are estimates only. You should review the results of REORGCHK with this information in mind.

REORG INDEXES/TABLE command

Reorganizes an index or a table.

The REORG INDEXES ALL FOR TABLE table-name option reorganizes all indexes that are defined on a table by rebuilding the index data into unfragmented, physically contiguous pages. If you specify the CLEANUP ONLY option of the index option, cleanup is performed without rebuilding the indexes. If you attempt to use this command against indexes on declared temporary tables, the SQLSTATE 42995 error will be returned.

The REORG TABLE table-name option reorganizes a table by reconstructing the rows to eliminate fragmented data, and by compacting information. The reorganization of the table is done by one of two methods:

Classic
Allows only limited read-only access to the data during the first phases of the reorganization.

Inplace
Allows access to the data throughout the reorganization but without being able to ensure perfectly ordered data. Inplace table reorganization is allowed only on tables with type-2 indexes and without extended indexes.

When you choose either type of reorganization, the indexes for the table are rebuilt after the table is reorganized. However, with the inplace method, the index will be imperfectly reorganized and may require that you reorganize the indexes later to reduce index fragmentation and to reclaim index object space.

Example:

For a classic (offline) REORG TABLE like the default in DB2 Universal Database™ (UDB) Version 7, enter the following command

```
db2 reorg table employee index empid allow no access indexscan
   longlobdata
```

Note: The defaults are different in DB2 UDB Version 8.

Usage notes:

DB2 UDB provides two methods of reorganizing tables: classic and inplace. In general, classic table reorganization is faster, but should be used only if your applications function without write access to tables during the reorganization. If your environment does not allow this restriction, although inplace reorganization is slower, it can occur in the background while normal data access continues.

Classic table reorganization provides the fastest table reorganization, especially if you do not need to reorganize LOB or LONG data. In addition, indexes are rebuilt in perfect order after the table is reorganized. Read-only applications can access the original copy of the table except during the last phases of the reorganization, in which the permanent table replaces the shadow copy of the table and the indexes are rebuilt.

Inplace table reorganization is slower and does not ensure perfectly ordered data, but it can allow applications to access the table during the reorganization. In addition, inplace table reorganization can be paused and resumed later by anyone with the appropriate authority by using the schema and table name.

Updated restrictions:

The REORG utility does not support the use of nicknames.

- The REORG TABLE command is not supported for declared temporary tables.
- The REORG TABLE command cannot be used on views.
- Reorganization of a table is not compatible with range-clustered tables, because the range area of the table always remains clustered.
- An offline table reorganization does not allow the starting of an online backup operation of a table space in which the table resides while the reorganization is happening.
- An online table reorganization of an SMS table does not allow the starting of an online backup operation of a table space in which the table resides while the reorganization is happening.
- An online table reorganization of a DMS table does allow the starting of an online backup operation of a table space in which the table resides while the reorganization is happening. There may be lock waits of the reorganization operation during the truncate phase.
- REORG TABLE cannot use an index that is based on an index extension.

BACKUP DATABASE command

Usage notes:

Note the following restrictions:

- The starting of an online backup operation of a DMS table space is allowed when a table within the same table space is being reorganized online. There may be lock waits of the reorganization operation during the truncate phase.
- The starting of an online backup operation of an SMS table space is not allowed when a table within the same table space is being reorganized online. Both operations require an exclusive lock.

Migrating Databases

Restrictions:

Version 8 documentation ambiguously states that no database migration is required if the database has been migrated to a DB2 UDB Version 8 FixPak level. To be specific, database migration is not required between fixpaks once the database is at a Version 8 level (Version 8.1 or 8.2 or a subsequent fixpak). There are changes to the database directory file structure in Version 8.2 and migration is automatically performed for you when you move from Version 7 or Version 8.1 to Version 8.2. However, if you go back from Version 8.2 to Version 8.1, you must run **db2demigdbd** to restore the database directory file structure. Failing to do so will result in error SQL10004 when you try to access the database.

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.

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.

Monitor and troubleshoot command (db2pd) updates

The Monitor and troubleshoot DB2 command (db2pd) retrieves information from the DB2 UDB memory sets. The db2pd system command has been enhanced in the following ways:

New -hadr parameter

Introduced at Version 8.2 (equivalent to Version 8.1 FixPak 7), the -hadr parameter 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*

New -utilities parameter

Introduced at Version 8.2 (equivalent to Version 8.1 FixPak 7), the -utilities parameter reports Utility information. Descriptions of each reported element can be found in the Utilities section of the *System Monitor Guide and Reference*.

New -activestatements parameter

Introduced at Version 8.2.2 (equivalent to Version 8.1 FixPak 9), the -activestatements parameter returns information about active statements. The following information is returned:

AppHandl
: The application handle using the active statement, including the node and the index.

UOW-ID
: For the application associated with **AppHandl**, the unit of work identifier in which the statement became active.

StmtID
: The statement identifier within the unit of work.

AnchID
: The identifier for the package cache dynamic SQL hash anchor.

StmtUID
: The identifier for the package cache dynamic SQL statement unique within the hash anchor.

EffISO
: The effective isolation level of the statement.

EffLockTOut
: The effective lock timeout value for the statement.

EffDegree
: The effective degree of parallelism of the statement.

StartTime
: The time the statement started.

LastRefTime
: The last time the statement was referenced by the application.

New wait option for the -locks parameter

Starting with Version 8.2.2 (equivalent to Version 8.1 FixPak 9), you can specify the `wait` option with the -locks parameter to return only locks with a waiting status and locks that are being waited on.

New fields returned by the -applications parameter

Starting with Version 8.2.2 (equivalent to Version 8.1 FixPak 9), the -applications parameter returns four new fields:

C-AnchID
: This new field is the identifier for the package cache dynamic SQL hash anchor of the current statement. The value for this field is 0 if no current statement exists. The current hash anchor identifier is populated only when the application is executing a request related to the statement, such as a FETCH request on a cursor. Once the request processing is complete, the values are set to 0. The value is also set to 0 for static SQL statements.

C-StmtUID
: This new field is the package cache dynamic SQL statement unique

identifier within the hash anchor of the current statement. The value for this field is 0 if no current statement exists. The current statement unique identifier is populated only when the application is in the process of executing a request related to the statement. Once the request execution is complete, the values are set to 0. The value is also set to 0 for static SQL statements.

L-AnchID
This new field is the package cache dynamic SQL hash anchor identifier of the application's last executed statement. The value for this field is 0 if no last executed statement exists, such as before any dynamic SQL statement is executed. The value is also set to 0 for static SQL statements. The last anchor hash identifier is populated after each request execution is complete and survives until the completion of the next associated request .

L-StmtUID
This new field is the package cache dynamic SQL statement unique identifier within the hash anchor of the application's last executed statement. The value for this field is 0 if no last executed statement exists or if the last statement executed was static SQL. The last statement identifier is populated after the current request completes and survives until the completion of the next request.

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, UNIX, or Windows database servers from Linux, UNIX, 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 30. 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 30. 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 31. 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 32. 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:

```
>>--UPDATE HISTORY--+-FOR--object-part-+--WITH------------------------------------->
                    '-EID--eid---------'

>--+-LOCATION--new-location--DEVICE TYPE--new-device-type-+-------------------------><
   +-COMMENT--new-comment---------------------------------+
   '-STATUS--new-status-----------------------------------'
```

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
- Enables the following built-in routines:
  - 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

- GET_WRAP_CFG_C
- LIST_FILES_C
- ODBC_DISCOVERY
- HEALTH_HI_REC
- EXEC_DB2_SCRIPT
- HEALTH_HI_REC
- EXEC_DB2_SCRIPT
- ADMIN_LIST_HIST
- GET_DBM_CONFIG

- 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
- Changes the parameter lengths of the following functions:
  - SQLCACHE_SNAPSHOT

  - SNAPSHOT_APPL_INFO
  - SNAPSHOT_STATEMENT
  - SNAPSHOT_LOCKWAIT
  - SNAPSHOT_DATABASE
  - SNAPSHOT_BP
  - SNAPSHOT_LOCK
  - SNAPSHOT_TABLE
  - SNAPSHOT_TBREORG
  - SNAPSHOT_SUBSECT
  - SNAPSHOT_DYN_SQL
  - HEALTH_DBM_INFO
  - HEALTH_DBM_HI
  - HEALTH_DBM_HI_HIS
  - HEALTH_DB_INFO
  - HEALTH_DB_HI
  - HEALTH_DB_HI_HIS
  - DB_PARTITIONS
  - GROUPS_FOR_USER
  - DB_PARTITION
- Changes the following functions to use schema *db2schema* instead of the obsolete *db2schem* and to enable them to modify SQL data:
  - SQLTABLES
  - SQLTABLEPRIVILEGES
  - SQLSTATISTICS
  - SQLSPECIALCOLUMNS
  - SQLPROCEDURES
  - SQLPROCEDURECOLS
  - SQLPRIMARYKEYS
  - SQLFOREIGNKEYS
  - SQLCOLUMNS
  - SQLCOLPRIVILEGES
- Changes the following functions to use schema *db2schema* instead of the obsolete *db2schem* :
  - SQLUDTS
  - SQLGETTYPEINFO

Authorization:

sysadm

Required connection:

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

Command syntax:

```
►►──db2updv8──-d──database-name──┬─────────────────────────────┬──┬────┬──────────────►◄
                                 └─-u──userid──-p──password─┘  └─-h─┘
```

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:

```
►►──db2xprt──┬──────────┬──┬────┬──┬────┬──infile──┬─────────┬──────────────►◄
             ├─/p──path─┤  └─/m─┘  └─/n─┘          └─outfile─┘
             └─/v───────┘
```

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

Bind files used by the export, import and load utilities

A new bind file called db2uImpInsUpdate.bnd has been added to the import utility, with a default isolation level of Read Stability (RS). This bind file is used by the Import Utility only during INSERT_UPDATE. The import utility's INSERT, REPLACE and CREATE options still use the db2uimpm.bnd file.

The db2uImpInsUpdate.bnd bind file cannot be bound with the INSERT BUF option. Attempting to perform IMPORT INSERT_UPDATE while db2uImpInsUpdate.bnd is bound with INSERT BUF causes the import utility to fail, resulting in the following error:

> SQL3525: The "INSERT_UPDATE" option is incompatible with the "INSERT BUF BIND ON DB2UIMPINSUPDATE.BND" option.

Using import with buffered inserts

Version 8.2 of the *Data Movement Utilities Guide and Reference* states:

> The buffered inserts feature cannot be used in conjunction with import operations in which the INSERT_UPDATE parameter is specified. A new bind file (db2uimpm2.bnd) is introduced to enforce this restriction.

Due to the introduction of a new bind file, the statement should be:

> The buffered inserts feature cannot be used in conjunction with import operations in which the INSERT_UPDATE parameter is specified. A new bind file (db2uImpInsUpdate.bnd) is introduced to enforce this restriction.

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

Backup overview

Note the following restrictions:

- The starting of an online backup operation of a DMS table space is allowed when a table within the same table space is being reorganized online. There may be lock waits of the reorganization operation during the truncate phase.

- The starting of an online backup operation of an SMS table space is not allowed when a table within the same table space is being reorganized online. Both operations require an exclusive lock.

Using backup

The following restrictions apply to the backup utility:

- A table space backup operation and a table space restore operation cannot be run at the same time, even if different table spaces are involved.
- If you want to be able to do rollforward recovery in a partitioned database environment, you must regularly back up the database on the list of nodes, and you must have at least one backup image of the rest of the nodes in the system (even those that do not contain user data for that database). Two situations require the backed-up image of a database partition at a database partition server that does not contain user data for the database:
  - You added a database partition server to the database system after taking the last backup, and you need to do forward recovery on this database partition server.
  - Point-in-time recovery is used, which requires that all database partitions in the system are in rollforward pending state.
  - Online backup operations for DMS table spaces are incompatible with the following operations:
    - load
    - Reorganization (offline only); (online reorganization of a table in the table space being backed up can be used, although there may be lock waits during the truncate phase of the reorganization operation).
    - drop table space
    - table truncation
    - index creation
    - not logged initially (used with the CREATE TABLE and ALTER TABLE statements)

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 33. 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 5.3
- 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 or UNIX 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".

Fault monitor and health monitor comparison

The health monitor and the fault monitor are tools that work on a single database instance. The health monitor uses *health indicators* to evaluate the health of specific aspects of database manager performance or database performance. A health indicator measures the health of some aspect of a specific class of database objects, such as a table space. Health indicators can be evaluated against specific criteria to determine the health of that class of database object. In addition, health indicators can generate alerts to notify you when an indicator exceeds a threshold or indicates a database object is in a non-normal state

By comparison, the fault monitor is solely responsible for keeping the instance it is monitoring up and running. If the DB2 UDB instance it is monitoring terminates unexpectedly, the fault monitor restarts the instance. The fault monitor is not available on Windows.

Turning off fault monitoring

To turn off fault monitoring for the database instance DB2INST1, type the following command from a DB2 UDB command window:

```
db2fm -i db2inst1 -f no
```

Note: If the fault monitor registry file does not exist, the default values are used.

To confirm that fault monitor is no longer running for DB2INST1, type the following command on UNIX systems:

```
ps -ef|grep -i fm
```

On Linux systems, type the following command:

```
ps auxw|grep -i fm
```

An entry that shows db2fmd and DB2INST1 indicates that the fault monitor is still running on that instance. To turn off the fault monitor, type the following command as the instance owner:

```
db2fm -i db2inst1 -D
```

Data Warehouse Center

Setting up the warehouse client environment on AIX and Solaris operating systems

On AIX and Solaris operating systems, you must set your locale correctly prior to running the warehouse client.

To set your locale, type the following command from a command line: `export LC_ALL=<locale>`

Note the following example: `export LC_ALL=Ja_JP` where `Ja_JP` is your local locale.

If this environment variable is not set, then GUI applications might not run correctly and can result in Java errors.

Updating the database manager configuration before installing warehouse transformers

In the *DB2 Warehouse Manager Standard Edition Installation Guide*, Version 8.2, the topic “Updating the database manager configuration before installing warehouse transformers” incorrectly instructs you to update the SDK path parameter. You must update the JDK path parameter.

Use the DB2 Command Line Processor to update the database manager configuration for the target DB2 instance before you install warehouse transformers.

Procedure:

To update the database manager configuration before installing warehouse transformers:

1. Set the JDK path parameter (JDK_PATH) to the subdirectory where the JDK is installed:

   ```
   UPDATE DATABASE MANAGER CONFIGURATION USING JDK_PATH path
   ```

   where *path* is the subdirectory where the JDK is installed.

   Note: Do not include the `bin` directory in JDK_path. If you are using a DB2 Universal Database Version 7.2 (or earlier) target that has transformers installed, you should update JDK11_PATH as opposed to JDK_PATH.
2. Set the Java Virtual Machine heap size parameter (JAVA_HEAP_SZ) to 4096:

   ```
   UPDATE DATABASE MANAGER CONFIGURATION USING JAVA_HEAP_SZ 4096
   ```

VW_NETRC environment variable added to the Data Warehouse Center (UNIX)

Starting with Version 8.2 FixPak 10 (equivalent to Version 8.1 FixPak 3) of the Data Warehouse Center, you can set the VW_NETRC environment variable on UNIX operating systems, except for on Linux. Setting the VW_NETRC environment variable to off allows you to manually maintain the `.netrc` file. Refer to your operating system's documentation for information about the correct format for the `.netrc` file.

Running concurrent FTP user-defined programs might corrupt the `.netrc` file. If you want to run concurrent FTP user-defined program steps, you can set the VW_NETRC environment variable to off (VW_NETRC=OFF). Add this environment variable to your agent daemon profile for remote agents and to the `IWH.environment` file for your default or local agent.

DB2 Data Warehouse Center requires Unicode format database

After installing DB2 Universal Database Version 8.1 FixPak 7 or later, you need to run the Warehouse Control Database Management tool to create a new warehouse control database that is in Unicode format.

Prerequisites:

To create and store a copy of the existing warehouse control database, your workstation must have enough disk space to store this copy plus twice as much space as the warehouse control database requires to store temporary files. For example, if the existing warehouse control database is 10 MB, a total of 30 MB must be available in the same instance as the existing warehouse control database.

Procedure:

Follow these steps to create a new warehouse control database in Unicode format:

1. Run the Warehouse Control Database Management tool:
   - **On Windows**: Click **Start** and select **Programs → IBM DB2→ Set up Tools→ Warehouse Control Database Management**.
   - **On AIX**: Run the **db2wcdbm** script.
2. Enter the name of the existing warehouse control database. You are then prompted for a new warehouse control database name. This creates a new Unicode warehouse control database.

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.

Preparing to install warehouse agents

When installing DB2 Warehouse Manager, the warehouse server, warehouse client, and all of the warehouse agents must be at the same version and fix pack level.

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

Concurrent active data reader support

The DB2 .NET Data Provider now supports the use of concurrent active data readers. This means that you can concurrently access data from multiple DB2DataReader instances that use the same DB2Connection instance. Each DB2DataReader instance must be associated with its own DB2Command instance. In order to use the associated DB2Command instance for any other purpose, you must explicitly call the `DB2DataReader.Close` method.

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.

DB2 Connect

Change to authentication type negotiation through a gateway

Starting with DB2 Connect™ Version 8.2.2 (equivalent to Version 8.1 FixPak 9) the gateway is no longer a passive participant during authentication negotiation. Instead, the gateway takes an active role. The authentication type specified in the database directory entry at the gateway overrides the authentication type cataloged at the client. The client, gateway, and server must all specify compatible types. If the cataloged authentication type at the gateway has not been specified in the database directory entry, SERVER authentication will be the default type requested of the server. However, negotiation will still take place between the client and server if the server does not support SERVER authentication. This behavior is in contrast to the client which defaults to SERVER_ENCRYPT if an authentication type has not been specified.

The authentication type cataloged at the gateway is not used if DB2NODE or the SQL_CONNECT_NODE option of the Set Client API has been set at the client. In these cases negotiation is still strictly between the client and the server.

A server with SERVER_ENCRYPT specified as the authentication type in the database manager configuration no longer accepts connections or attachments from clients that request SERVER authentication.

New security scenario

A new security scenario has been added for APPC connections:

Authentication	GSSPLUGIN
Security	none
Validation	GSS API security plugin mechanism

Corrections to diagrams

The following DB2 Connect Enterprise Edition topics have diagrams that are incorrect:

- Accessing host or iSeries DB2 data using DB2 Connect Enterprise Edition
- Accessing DB2 data from the Web using Java

The following table outlines corrections for diagrams in the ″Accessing host or iSeries DB2 data using DB2 Connect Enterprise Edition″ topic.

Table 34. Corrections for diagrams in the ″Accessing host or iSeries DB2 data using DB2 Connect Enterprise Edition″ topic

Location within the topic	Correction
Legend for all four diagrams	• References to ″DB2 for OS/390 V5R1″ should be ″DB2 for OS/390 V6 or later″. • References to ″DB2 for AS/400® V4R2″ should be ″DB2 for iSeries V5R1 or later″.
First diagram (Figure 1: DB2 Connect Enterprise Edition)	All references to ″APPC″ and ″SNA Communications Support″ are incorrect. SNA/APPC is not supported as an inbound protocol for the DB2 Runtime Client by DB2 Linux, Unix, and Windows servers, including DB2 Connect Enterprise Edition.

The following table outlines corrections for diagrams in the ″Accessing DB2 data from the Web using Java″ topic.

Table 35. Corrections for diagrams in the ″Accessing DB2 data from the Web using Java″ topic

Location within the topic	Correction
Legend	• References to ″DB2 for OS/390 V5R1″ should be ″DB2 for OS/390 V6 or later″. • References to ″DB2 for AS/400 V4R2″ should be ″DB2 for iSeries V5R1 or later″.

Development Center

DB2 Development Center Version 8.2 requires level 9.2.9 of the Distributed Debugger

DB2 Development Center Version 8.2 now requires Version 9.2.9 of the IBM Distributed Debugger. If you do not have Version 9.2.9 of the Distributed Debugger installed, you cannot debug Java stored procedures using the Development Center.

Version 9.2.9 of the Distributed Debugger no longer supports Solaris Operating Environments.

For more information about the Distributed Debugger, visit to the Distributed Debugger Web site at http://www.ibm.com/software/awdtools/debugger.

Length restriction in the DB2 Development Center Change Variable Range window

When you are changing the length of a variable using the Change Variable Range dialog in DB2 Development Center, there is a maximum length of 1024 bytes. This limitation is currently documented in an English-only message.

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 36. 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.
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.

Information Catalog Center

Configuration sample scripts

The `ICCConfig.jacl` and `ICCConfig.properties` sample scripts are provided with the Information Catalog Center for the Web with the DB2 Embedded Application Server. You can use these sample scripts to configure the Information Catalog Center for the Web with WebSphere Application Server 5. These scripts are located in the `sqllib\samples\icweb` directory.

Web server configuration

When configuring the Information Catalog Center for the Web with the DB2 Embedded Application Server, if your metadata contains URLs that access files on the server, you must map the URLs to the correct location by using aliases in the Web server configuration. You must also map the help and copyright links. If you use the DB2 Embedded Application Server, a Web server must be configured correctly and running for these links to work even though you don't need to plug in to a Web server.

Installation and Configuration Supplement

DB2 node configuration file - db2nodes.cfg

The `resourcesetname` parameter is only supported on AIX, HP-UX, Solaris Operating Environment and Linux.

On Linux operating systems, the `resourcesetname` column defines a number that corresponds to a Non-Uniform Memory Access (NUMA) node on the system. The `numactl` system utility must be available in addition to a 2.6 kernel with NUMA policy support. Refer to the `man` page for `numactl` for more information about NUMA support on Linux operating systems.

Example using resourcesetname on Linux operating systems:

This example shows how to set up a four node NUMA computer with each logical node associated with a NUMA node.

Procedure:

1. Ensure that NUMA capabilities exist on your system.
2. Issue the following command:

   ```
   $ numactl --hardware
   ```

   Output similar to the following displays:

   ```
   available: 4 nodes (0-3)
   node 0 size: 1901 MB
   node 0 free: 1457 MB
   node 1 size: 1910 MB
   node 1 free: 1841 MB
   ```

```
node 2 size: 1910 MB
node 2 free: 1851 MB
node 3 size: 1905 MB
node 3 free: 1796 MB
```

3. In this example, there are four NUMA nodes on the system. Edit the db2nodes.cfg file as follows to associate each Multiple Logical Node (MLN) with a NUMA node on the system:

   ```
   0 hostname 0 hostname 0
   1 hostname 1 hostname 1
   2 hostname 2 hostname 2
   3 hostname 3 hostname 3
   ```

DB2NOLIOAIO registry variable replaced by DB2LINUXAIO (Linux)

The DB2 Universal Database (UDB) registry variable DB2NOLIOAIO is deprecated as of Version 8.2.2 (equivalent to Version 8.1 FixPak 9). For Linux users, the DB2NOLIOAIO registry variable has been replaced with DB2LINUXAIO.

- To enable asynchronous I/O (AIO) support on Linux, run the following command:

  ```
  db2set DB2LINUXAIO=true
  ```

  and restart DB2 UDB.

  Note: To use AIO, users must install libaio-0.3.98 or later and have a kernel that supports AIO.
- To disable AIO support on Linux, run the following command:

  ```
  db2set DB2LINUXAIO=false
  ```

  and restart DB2 UDB.

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 or UNIX operating systems, or as a user with **Administrator** privileges on Windows operating systems.
2. For Linux– and UNIX–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 and UNIX 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, UNIX, 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, UNIX, 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, UNIX, 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, UNIX, 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 installation 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 37. Comparison of direct I/O and raw I/O

Raw I/O using block device with Direct I/O (new method)	**Raw I/O using character device driver and raw utility (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 DB2 UDB still supports the method of using the raw utility for raw I/O, recent distributions have deprecated this feature and may remove it in the future. The preferred method is to use the new method by specifying the devices directly.

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 and UNIX)

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 and UNIX 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 and UNIX)

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 development rpm 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.

Query Patroller

Query class behavior update

A warning message is returned when one of the following tasks is performed through the Query Patroller Center or Query Patroller command line:

- Adding a query class
- Removing a query class
- Updating the maximum cost of a query for a query class

The warning message is:

```
DQP1024W  Creation, change, or removal of a query class will not
          take effect until the Query Patroller server is restarted.
```

Similarly, the *DB2 Query Patroller™ Guide: Installation, Administration, and Usage*, Version 8.2, states that you must restart the Query Patroller server after creating, changing, or removing query classes for your changes to take effect.

The message and the statement in the guide are no longer accurate. The three query class tasks listed previously will take effect immediately unless there are queued or running queries. If there are queued or running queries, including newly submitted queries, the query class changes will take effect when the queued or running queries complete. If you do not want to wait for all queued and running queries to complete, you must restart the Query Patroller server.

Note: As with earlier versions of Query Patroller, updating the maximum number of queries for a query class always takes effect immediately.

Definition updates for managed query states

The *Canceled* and *Done* query status meanings are updated as follows:

Canceled
The query was canceled, through either the Query Patroller Center or the Query Patroller command line, by the administrator, submitter, or an operator whose profile has the MONITORING privilege with edit authority. Only *running*, *held*, *released*, and *queued* queries can be *canceled*.

Done The query completed successfully.

Note: Although the query itself completed without error, the application may receive an error if the completion was caused by an external event, such as a **DB2 force** application.

Create Explain tables before running Query Patroller historical data generator

When running the historical data generator for Query Patroller, if the Explain tables do not already exist, the generator will create them for you. However, it is strongly recommended that you create the Explain tables before running the historical data generator. When you create the Explain tables be sure you create them on the same partition. Actively creating the Explain tables on the same partition improves the performance of the Explain facility. This improvement increases the performance of the historical data generator.

Checking Query Patroller log files for historical analysis

If the **Explain Run** column of the Query Activity over Time (Historical Analysis) report shows a status of **Ran unsuccessfully** for a query, historical data has not been generated for that query. Therefore, the query will not appear in any historical analysis reports or graphs. As documented in Version 8, to determine why the query was unsuccessful, you can examine the qpuser.log file.

In addition to examining the qpuser.log file, you should examine the qpdiag.log file.

Abnormal shutdown of the historical data generator

If you run the historical data generator and shut it down in an abnormal way, you will receive an error the next time you attempt to run the historical data generator. Examples of abnormal shutdown include:

- DB2 Universal Database stops unexpectedly
- Issuing a `db2stop force` command
- Issuing a `killdb2` command

When the historical data generator shuts down abnormally, you must issue the following command before attempting to rerun the historical data generator:

```
qp -d database generate historical_data stop
```

where *database* identifies the database that the command is being run against.

Dynamic query class updates

Certain query class operations no longer require Query Patroller to be stopped and restarted to take effect.

In the table that follows, an active query is a query whose status is Running or Queued.

Table 38. Conditions for query class changes to take effect

Nature of change	**Conditions for change to take effect**
Addition, removal, or update of a query class.	If there are no active queries, changes take effect immediately.
An update to a query class that involves only a change to the **Maximum number of queries**.	Takes effect immediately, even if there are active queries.

Table 38. Conditions for query class changes to take effect (continued)

Nature of change	Conditions for change to take effect
An update to a query class that involves only a change to the **Maximum cost of a query**.	If there are active queries, the update takes effect when either: • Query Patroller is stopped and restarted. • There are no more active queries. **Note:** When there is a change pending against **Maximum cost of a query**, subsequent query class updates of any kind will not take effect until one of the two previous conditions is met.
Addition or removal of a query class.	If there are active queries, the addition or removal takes effect when either: • Query Patroller is stopped and restarted. • There are no more active queries.

Nested query behavior

Nested queries cannot be queued. Instead, a nested query will run immediately if it exceeds a threshold that would normally cause it to be queued.

Limitations by SQL statement type

Contrary to previous documentation, the queries with the following statements can be queued:

- Queries containing static SQL with host variables
- Queries containing an identity value function (IDENTITY_VAL_LOCAL) or a sequence value function, such as NEXT VALUE FOR or PREVIOUS VALUE FOR

Resolution limitation when using the Terminal Services Client

When using the Terminal Services Client at resolution 640x480 to connect to a remote desktop that is running the Query Patroller Center, the Submission Preferences window might appear blank. For the Submission Preferences window to display properly, you must use a resolution higher than 640x480.

New group support for query submissions

Starting in Version 8.2, DB2 Universal Database (UDB) supports user groups beyond operating system groups. Therefore, there is a slight change in the **Submitter Profile to Use** drop-down list in the Query Submission Preferences window of the Query Patroller Center.

If you are logged in, but do not have either DBADM authority or Edit privilege for Query Patroller user administration, you can only add or update a submission preference for yourself. In this case, the **Submitter Profile to Use** drop-down list contains existing submitter profiles of the DB2 UDB groups that you belong to, instead of just the operating system groups that you belong to.

If you are logged in, and have either DBADM authority or Edit privilege for Query Patroller user administration, you can add or update submission preferences for other users. In this case, the **Submitter Profile to Use** drop-down list contains all existing group submitter profiles.

Query Patroller schedule limitations

When working with schedules in the Query Patroller Center, you can use the Schedule window to save schedules to a file and import them later. If you have a schedule that you saved using FixPak 6 or earlier, you cannot import the schedule using Version 8.2 or later. This limitation is due to the change in serialization between JDK levels introduced with DB2 UDB Version 8.2.

Authorization required to use RUN IN BACKGROUND QUERY command

To run the RUN IN BACKGROUND QUERY command, you must be the submitter who submitted the query originally.

Creating an alias for a result table

As of Query Patroller Version 8.1 FixPak 5, Query Patroller stopped creating result tables in the schema that matched the authorization ID of the submitter of the query. Instead, Query Patroller started creating result tables in a common DB2QPRT schema. To allow result tables to be referenced using the schema of the submitter, Query Patroller Version 8.2 introduces an option to automatically create an alias for each new result table that Query Patroller creates. The result table is created in the DB2QPRT schema and the alias is created in a schema that matches the submitter's authorization ID.

To turn this option on or off, issue the UPDATE QP_SYSTEM command with the CREATE_RESULT_TABLE_ALIASES option:

```
►►──UPDATE QP_SYSTEM USING──┬─DEFAULT──────────────────────────────────┬──────────────►◄
                            └─CREATE_RESULT_TABLE_ALIASES───┬─'Y'─┬─┘
                                                            └─'N'─┘
```

Removing orphaned result table aliases

Aliases created with CREATE_RESULT_TABLE_ALIASES option are automatically dropped when a result table is dropped. However, there are two situations in which a result table may be dropped without the corresponding alias being dropped.

- When the result table is dropped manually without using the qp command line or Query Patroller Center.
- When the result table is dropped using the qp command line or Query Patroller Center under the authority of an operator who is not the submitter of the query and does not have DBADM authority.

To clean up aliases that have no corresponding result tables, a new command, REMOVE RESULT_TABLE_ALIASES, has been created. This command is automatically executed whenever result tables are purged as part of the Query Patroller scheduled result table purging process. The REMOVE RESULT_TABLE_ALIASES command obtains the list of aliases to purge using the following query:

```
with a as (select tabschema, tabname from syscat.tables
           where type = 'A' and tabname like 'QUERY%_RESULTS'),
     t as (select tabname from syscat.tables
           where type = 'T' and tabname like 'QUERY%_RESULTS')
  select all tabschema, tabname from a
  where not exists (select * from t where t.tabname=a.tabname)
```

Prerequisites:

You must have DBADM authority.

Procedure:

1. Issue the REMOVE RESULT_TABLE_ALIASES command

This command removes all aliases that exist after having their corresponding result tables dropped. The aliases were originally created by Query Patroller for result tables.

Command syntax:

```
►►──REMOVE RESULT_TABLE_ALIASES──────────────────────────────────────►◄
```

Note: For information about entering Query Patroller commands using the command line interface, and general syntax for Query Patroller commands, see the Query Patroller command line interface.

Fenced user ID requires write access qpdiag.log file and path

Query Patroller uses some fenced stored procedures which may log entries to the qpdiag.log file. Therefore, the fenced user ID must have access to write to the qpdiag.log file and the path where the qpdiag.log file resides.

Quick Beginnings

Updating the DB2 Information Center installed on your computer or intranet server

To update your locally-installed DB2 Information Center, download the latest DB2 documentation FixPak from http://www.ibm.com/software/data/db2/udb/support/downloadv8_docfix.html. However, DB2 documentation FixPaks are released infrequently and may not reflect the most current level of the DB2 documentation. The latest version of the DB2 documentation is always the version of the DB2 Information Center hosted at http://publib.boulder.ibm.com/infocenter/db2help/

Memory requirements

At a minimum, DB2 Universal Database (UDB) requires 256 MB of RAM. For a system running just DB2 UDB and the DB2 GUI tools, a minimum of 512MB of RAM memory is required. However, 1GB of RAM memory is recommended for improved performance. These requirements do not include any additional memory requirements for other software that is running on your system.

When determining memory requirements, be aware of the following:

- For DB2 client support, these memory requirements are for a base of 5 concurrent client connections. You will need an additional 16 MB of RAM per 5 client connections.
- Memory requirements are affected by the size and complexity of your database system, as well as by the extent of database activity and the number of clients accessing your system.
- On Linux operating systems, a SWAP space that is at least twice as large as the amount of RAM is recommended but not required.

Clarification of DB2 UDB client support

The "DB2 clients" topic in the 8.1 version of the *DB2 Quick Beginnings for Clients* guide 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.

An amendment to that statement is as follows:

> While connections from Version N clients to Version N + 2 servers are possible in some environments, the DB2 support team will only provide support for this configuration as long as Version N is still in service. Once Version N is withdrawn from service, this configuration is no longer supported by the DB2 support team. DB2 Version 7 clients connecting to a DB2 Version 8 server is no longer supported by the DB2 support team because Version 7 has been withdrawn from service.

Modifying kernel parameters (Linux)

Before installing DB2 UDB, you should consider updating your Linux kernel parameters. DB2 Universal Database (UDB) automatically raises the IPC limits where necessary. You might want to raise these limits higher depending on your particular needs.

Prerequisites:

You must have root authority to modify kernel parameters.

Procedure:

To update kernel parameters:

Red Hat and SuSE

Systems using a 2.4.x series kernel have a default value for the message queue parameter (msgmni), which allows only a few simultaneous connections to DB2 UDB. Semaphore array parameters also have to be changed for DB2 UDB to run successfully. To check shared memory segment, semaphore array, and message queue limits, issue the **ipcs -l** command.

The following output is from the **ipcs -l** command.

```
   # ipcs -l

------ Shared Memory Limits --------
max number of segments = 4096             // SHMMNI
max seg size (kbytes) = 262144            // SHMMAX
max total shared memory (kbytes) = 8388608 // SHMALL
min seg size (bytes) = 1

------ Semaphore Limits --------
max number of arrays = 1024               // SEMMNI
max semaphores per array = 250
max semaphores system wide = 256000
max ops per semop call = 32
semaphore max value = 32767

------ Messages: Limits --------
```

```
max queues system wide = 1024              // MSGMNI
max size of message (bytes) = 65535        // MSGMAX
default max size of queue (bytes) = 16384  // MSGMNB
```

where

```
max semaphores system wide =
max number of arrays x max semaphores/array
```

For 32–bit Linux kernels, modify the kernel parameters by adding the following entries to the default system control configuration file, /etc/sysctl.conf:

```
kernel.msgmni = 1024
kernel.sem = "250 256000  32 1024"
kernel.shmmax=268435456
```

For 64–bit Linux kernels, modify the kernel parameters by adding the following entries to the default system control configuration file, /etc/sysctl.conf:

```
kernel.msgmni = 1024
kernel.sem = "250 256000  32 1024"
kernel.shmmax=1073741824
```

Run sysctl with the -p parameter to load in sysctl settings from the default file /etc/sysctl.conf:

```
 sysctl -p
```

The entries from the sysctl.conf file are read during startup by the network initialization script.

On some distributions you might be required to add sysctl -p in one of the system initialization files, such as rc.local, so that kernel parameters are set after each restart.

Modifying kernel parameters (Solaris Operating Environment)

The following information is an amendment to the "Modifying kernel parameters (Solaris Operating Environment)" topic in the *Quick Beginnings for DB2 Servers* guide:

For DB2 Universal Database (UDB) to operate properly, it is recommended that you update your system's kernel configuration parameters. You can use the **db2osconf** utility to suggest recommended kernel parameters.

To use the **db2osconf** command, you must first install DB2 UDB. The **db2osconf** utility can only be run from $DB2DIR/bin.

You must restart your system after modifying kernel parameters.

DB2 Universal Database Express Edition Version 8.2 Basics manual available for download

IBM DB2 Universal Database Express (DB2 UDB Express) is the newest member of the DB2 Universal Database Version 8 product family. It combines the power, function, and reliability of the IBM award-winning DB2 UDB relational database with simplicity in packaging, installation, and deployment at a minimal investment cost to meet the data management needs of small and medium businesses.

DB2 UDB Express is designed for customers with minimal in-house database skills who need an easy-to-install database integrated into their application software solutions. It is a multi-user version of DB2 UDB that supports local and remote applications in stand-alone and local area network (LAN) environments.

For further information on DB2 UDB Express, download *Quick Beginnings for DB2 Express Edition* and *DB2 Universal Database Express Edition Version 8.2 Basics* from the DB2 UDB product manuals Web page at http://www.ibm.com/software/data/db2/udb/support/manualsv8.html

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 and UNIX

```
~/sqllib/spatial
~/sqllib/spatial/bank
```

SQL Administrative Routines

Incorrect column name documented in result set for SNAP_GET_DYN_SQL table function

The ″SNAP_GET_DYN_SQL table function″ topic in the DB2 Information Center Version 8.2.2 (equivalent to Version 8.1 FixPak 9) incorrectly documents the result set for the SNAP_GET_DYN_SQL table function.

One of the columns is incorrectly documented as STMT_TXT.

The correct name for the output column is STMT_TEXT.

Snapshot monitor table functions have version-specific views

Version-specific views have been defined on the following snapshot monitor table functions, introduced in DB2 Universal Database Version 8.2.2 (equivalent to Version 8.1 FixPak 9):

- SNAP_GET_CONTAINER
- SNAP_GET_DB
- SNAP_GET_DYN_SQL
- SNAP_GET_STO_PATHS
- SNAP_GET_TAB
- SNAP_GET_TBSP
- SNAP_GET_TBSP_PART

The version-specific views are as follows:

- SYSCATV82.SNAPCONT
- SYSCATV82.SNAPDB
- SYSCATV82.SNAPDYNSQL
- SYSCATV82.SNAPSTOPATHS
- SYSCATV82.SNAPTAB
- SYSCATV82.SNAPTBSPACE
- SYSCATV82.SNAPTBSPACEPART

Because there is no guarantee that the result tables from the snapshot monitor table functions will remain unchanged from release to release, it is recommended that you use the version-specific views if you want guaranteed result tables. Each view contains all of the columns from the result table of its associated snapshot monitor table function.

GET_DB_CONFIG procedure requires an 8K minimum page size for a user temporary table space

The GET_DB_CONFIG procedure requires a user temporary table space with a page size of at least 8K.

The documented example that shows you how to use the GET_DB_CONFIG procedure should be replaced by the following example.

Using the command line processor (CLP), change the value of the *logretain* and the *userexit* database configuration parameters. Retrieve the original (on disk) and updated (in memory) values by calling the GET_DB_CONFIG procedure.

```
   UPDATE DB CFG USING LOGRETAIN RECOVERY USEREXIT YES
   CALL SYSPROC.GET_DB_CONFIG()
```

The following is an example of partial output from this procedure call.

```
Result set 1
--------------

DBCONFIG_TYPE ... LOGRETAIN   ... USEREXIT...
-------------     -----------     -----------
            0               1               1
            1               0               0

2 record(s) selected.

Return Status = 0
```

SQL Reference

CREATE PROCEDURE (SQL) statement authorizations

If the authorization ID of the statement has SYSADM authority but not DBADM authority, this ID is granted implicit DBADM authority for the purpose of creating the procedure. This applies to the CREATE PROCEDURE statement (SQL) and not the CREATE PROCEDURE statement (External).

EXPLAIN_DIAGNOSTIC: A new Explain table

The EXPLAIN_DIAGNOSTIC table contains an entry for each diagnostic message produced for a particular instance of an explained statement in the EXPLAIN_STATEMENT table.

The EXPLAIN_GET_MSGS table function queries the EXPLAIN_DIAGNOSTIC and EXPLAIN_DIAGNOSTIC_DATA Explain tables and returns formatted messages.

Table 39. EXPLAIN_DIAGNOSTIC table

Column name	Data type	Nullable	Key [1]	Description
EXPLAIN_REQUESTER	VARCHAR(128)	No	PK, FK	Authorization ID of initiator of this Explain request.
EXPLAIN_TIME	TIMESTAMP	No	PK, FK	Time of initiation for Explain request.
SOURCE_NAME	VARCHAR(128)	No	PK, FK	Name of the package running when the dynamic statement was explained or name of the source file when the static SQL was explained.
SOURCE_SCHEMA	VARCHAR(128)	No	PK, FK	Schema, or qualifier, of source of Explain request.
SOURCE_VERSION	VARCHAR(64)	No	PK, FK	Version of the source of the Explain request.
EXPLAIN_LEVEL	CHAR(1)	No	PK, FK	Level of Explain information for which this row is relevant. Valid values are: • **O**: Original Text (as entered by user) • **P**: PLAN SELECTION
STMTNO	INTEGER	No	PK, FK	Statement number within package to which this Explain information is related. Set to 1 for dynamic Explain SQL statements. For static SQL statements, this value is the same as the value used for the SYSCAT.STATEMENTS catalog view.

Table 39. EXPLAIN_DIAGNOSTIC table (continued)

Column name	Data type	Nullable	Key [1]	Description
SECTNO	INTEGER	No	PK, FK	Section number within package that contains this SQL statement. For dynamic Explain SQL statements, this is the section number used to hold the section for this statement at runtime. For static SQL statements, this value is the same as the value used for the SYSCAT.STATEMENTS catalog view.
DIAGNOSTIC_ID	INTEGER	No	PK	ID of the diagnostic for a particular instance of a statement in the EXPLAIN_STATEMENT table.
CODE	INTEGER	No	No	A unique number assigned to each diagnostic message. The number can be used by a message API to retrieve the full text of the diagnostic message.

1. PK means that the column is part of a primary key; FK means that the column is part of a foreign key.

EXPLAIN_DIAGNOSTIC_DATA: A new Explain table

The EXPLAIN_DIAGNOSTIC_DATA table contains message tokens for specific diagnostic messages that are recorded in the EXPLAIN_DIAGNOSTIC table. The message tokens provide additional information that is specific to the execution of the SQL statement that generated the message.

The EXPLAIN_GET_MSGS table function queries the EXPLAIN_DIAGNOSTIC and EXPLAIN_DIAGNOSTIC_DATA Explain tables, and returns formatted messages.

Table 40. EXPLAIN_DIAGNOSTIC_DATA Table

Column name	Data type	Nullable	Key [1]	Description
EXPLAIN_REQUESTER	VARCHAR(128)	No	FK	Authorization ID of initiator of this Explain request.
EXPLAIN_TIME	TIMESTAMP	No	FK	Time of initiation for Explain request.
SOURCE_NAME	VARCHAR(128)	No	FK	Name of the package running when the dynamic statement was explained or name of the source file when the static SQL was explained.
SOURCE_SCHEMA	VARCHAR(128)	No	FK	Schema, or qualifier, of source of Explain request.
SOURCE_VERSION	VARCHAR(64)	No	FK	Version of the source of the Explain request.
EXPLAIN_LEVEL	CHAR(1)	No	FK	Level of Explain information for which this row is relevant. Valid values are: • **O**: Original Text (as entered by user) • **P**: PLAN SELECTION
STMTNO	INTEGER	No	FK	Statement number within package to which this Explain information is related. Set to 1 for dynamic Explain SQL statements. For static SQL statements, this value is the same as the value used for the SYSCAT.STATEMENTS catalog view.
SECTNO	INTEGER	No	FK	Section number within package that contains this SQL statement. For dynamic Explain SQL statements, this is the section number used to hold the section for this statement at runtime. For static SQL statements, this value is the same as the value used for the SYSCAT.STATEMENTS catalog view.
DIAGNOSTIC_ID	INTEGER	No	PK	ID of the diagnostic for a particular instance of a statement in the EXPLAIN_STATEMENT table.
ORDINAL	INTEGER	No	No	Position of token in the full message text.

Table 40. EXPLAIN_DIAGNOSTIC_DATA Table (continued)

Column name	Data type	Nullable	Key [1]	Description
TOKEN	VARCHAR(1000)	Yes	No	Message token to be inserted into the full message text; might be truncated.
TOKEN_LONG	BLOB(3M)	Yes	No	More detailed information, if available.

1. PK means that the column is part of a primary key; FK means that the column is part of a foreign key.

Schema used by the Explain facility

The Explain facility uses the following IDs as the schema when qualifying Explain tables that it is populating:

- 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 those tables.

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 41. 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

Health indicators summary

In the "Health indicators summary" topic in the DB2 Information Center for Version 8.2.2 (equivalent to Version 8.1 FixPak 9), the identifier for the Database automatic storage utilization health indicator is incorrectly documented as *db.db_auto_storage_util*.

The correct identifier for the Database automatic storage utilization health indicator is *db.auto_storage_util*.

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 runtime rollback process

Progress monitoring of runtime 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

Parameter change to dxxGenXML, dxxGenXMLClob, dxxRetrieveXML, and dxxRetrieveXMLClob stored procedures

The `override` parameter's description for the following stored procedures has changed:

- dxxGenXML()
- dxxGenXMLClob()
- dxxRetrieveXML()
- dxxRetrieveXMLClob()

The update is as follows:

Parameters:

Table 42. Override parameter for the dxxGenXML, dxxGenXMLClob, dxxRetrieveXML, and dxxRetrieveXMLClob stored procedures

Parameter	Description	IN/OUT Parameter
override	Overrides the condition in the DAD file. The input value is based on the *overrideType*. • **NO_OVERRIDE**: A NULL string. • **SQL_OVERRIDE**: A valid SQL statement. Using this *overrideType* requires that SQL mapping is used in the DAD file. The input SQL statement overrides the SQL_stmt in the DAD file. • **XML_OVERRIDE**: A string that contains one or more expressions that are separated by the word AND. Using this *overrideType* requires that RDB_node mapping is used in the DAD file.	IN

Decomposing XML documents into DB2 UDB data

Decomposition table size limits:

Decomposition uses RDB_node mapping to specify how an XML document is decomposed into DB2 UDB tables by extracting the element and attribute values and storing them in table rows. The values from each XML document are stored in one or more DB2 UDB tables. Each table can have a maximum of 10240 rows decomposed from each document.

For example, if an XML document is decomposed into five tables, each of the five tables can have up to 10240 rows for that particular document. If the table has rows for multiple documents, it can have up to 10240 rows for each document.

Using multiple-occurring elements (elements with location paths that can occur more than once in the XML structure) affects the number of rows . For example, a document that contains an element <Part> that occurs 20 times, might be decomposed as 20 rows in a table. When using multiple occurring elements, consider that a maximum of 10240 rows can be decomposed into one table from a single document.

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 and UNIX)
   - 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 43. 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 45

UNIX operating systems

The files on the FixPak CD are located as follows

Table 44. 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 45

The following table outlines the language directory names and their corresponding languages.

Table 45. Directory names and their corresponding languages

Directory	Language
ar_AA	Arabic

Table 45. 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

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.

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.

Printed in USA