IBM® DB2® Universal Database

Release Notes

Version 7

IBM® DB2® Universal Database

Release Notes

Version 7

Contents

Welcome to DB2 Universal Database Version 7 FixPak Release Notes!

Note: When viewing as text, set the font to monospace for better viewing of these Release Notes.

The DB2 Universal Database and DB2 Connect Support site is updated regularly. Check http://www.ibm.com/software/data/db2/udb/winos2unix/support for the latest information.

This file contains information for the following products that was not available when the DB2 manuals were printed:

```
    IBM DB2 Universal Database Personal Edition, Version 7.2
    IBM DB2 Universal Database Workgroup Edition, Version 7.2
    IBM DB2 Universal Database Enterprise Edition, Version 7.2
    IBM DB2 Data Links Manager, Version 7.2
    IBM DB2 Universal Database Enterprise - Extended Edition, Version 7.2
    IBM DB2 Query Patroller, Version 7.2
    IBM DB2 Personal Developer's Edition, Version 7.2
    IBM DB2 Universal Developer's Edition, Version 7.2
    IBM DB2 Data Warehouse Manager, Version 7.2
    IBM DB2 Relational Connect, Version 7.2
 IBM DB2 Connect Personal Edition, Version 7.2
 IBM DB2 Connect Enterprise Edition, Version 7.2
```

An additional Release Notes file, installed as READCON.TXT, is provided for the following products:

```
    IBM DB2 Connect Personal Edition, Version 7.2
    IBM DB2 Connect Enterprise Edition, Version 7.2
```

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

Information about this product is available online at http://www.ibm.com/software/data/db2/lifesciencesdataconnect.

The following books were updated for FixPak 4, and the latest PDFs are available for download online at http://www.ibm.com/software/data/db2/udb/winos2unix/support:

```
Administration Guide
Application Building Guide
Application Development Guide
Command Reference
```

```
Data Recovery and High Availability Guide and Reference
Data Warehouse Center Administration Guide
Message Reference
SQL Reference
DB2 Warehouse Manager Installation Guide
```

The information in these notes is in addition to the updated references. All updated documentation is also available on CD. This CD can be ordered through DB2 service using the PTF number U478862. Information on contacting DB2 Service is available at http://www.ibm.com/cgi-bin/db2www/data/db2/udb/winos2unix/support/help.d2w/report.

The What's New book contains an overview of some of the major DB2 enhancements for Version 7.2. If you don't have the 7.2 version of the What's New book, you can view it and download it from http://www.ibm.com/software/data/db2/udb/winos2unix/support.

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.

Note: Throughout these Release Notes, when reference is made to Windows NT, this includes Windows 2000. The reference further includes Windows XP when in the context of products listed in "Support for Windows ME, Windows XP and Windows 2000 Datacenter Edition Platforms" on page 20, unless otherwise specified.

Note: A revision bar (|) on the left side of a page indicates that the line has been added or modified since the Release Notes were first published.

Part 1. Read Me First

Version 7 Release Notes

These Release Notes have been updated up to FixPak 8. The information contained in them is still valid for users of later FixPaks. This list is not exhaustive, but highlights the major documentation changes. Be sure to look over any sections of the release notes that relate to your work environment to be sure that you keep abreast of all pertinent updates. Please see the FixPak Readmes for information on any new function added to Version 7 after FixPak 8.

Product Notes

Supported CPUs on DB2 Version 7 for the Solaris Operating Environment

CPU versions previous to UltraSparc are not supported.

Chinese Locale Fix on Red Flag Linux

If you are using Simplified Chinese Red Flag Linux Server Version 1.1, contact Red Flag to receive the Simplified Chinese locale fix. Without the Simplified Chinese locale fix for Version 1.1, DB2 does not recognize that the code page of Simplified Chinese is 1386.

Additional Locale Setting for DB2 for Linux in a Japanese and Simplified Chinese Linux Environment

An additional locale setting is required when you want to use the Java GUI tools, such as the Control Center, on a Japanese or Simplified Chinese Linux system. Japanese or Chinese characters cannot be displayed correctly without this setting. Please include the following setting in your user profile, or run it from the command line before every invocation of the Control Center.

```
For a Japanese system:
   export LC_ALL=ja_JP

For a Simplified Chinese system:
   export LC_ALL=zh_CN
```

Limitation for Japanese on PTX

If you are running DB2 UDB in Japanese on a PTX system, it is possible that some of the processes DB2 uses will not inherit the correct locale information. To avoid this, manually set the DB2CODEPAGE and DB2COUNTRY registry variables to correspond to your locale.

Control Center Problem on Microsoft Internet Explorer

There is a problem caused by Internet Explorer (IE) security options settings. The Control Center uses unsigned jars, therefore access to system information is disabled by the security manager.

To eliminate this problem, reconfigure the IE security options as follows:

1. Select **Internet Options** on the **View** menu (IE4) or the **Tools** menu (IE5).
2. On the Security page, select **Trusted sites zone**.

3. Click **Add Sites...**.

4. Add the Control Center Web server to the trusted sites list. If the Control Center Web server is in the same domain, it may be useful to add only the Web server name (without the domain name). For example:

```
http://ccWebServer.ccWebServerDomain
http://ccWebServer
```

   Note: When entering the URL, you must either use the https:// prefix, or deselect the **Require server verification (https:) for all sites in this zone** option in order to add the site

5. Click **OK**.

6. Click on **Settings...**(IE4) or on **Custom Level...** (IE5)..

7. Scroll down to **Java** → **Java Permissions** and select **Custom**.

8. Click **Java Custom Settings...**.

9. Select the Edit Permissions page.

10. Scroll down to Unsigned Content → Run Unsigned Content → Additional Unsigned Permissions → System Information and select **Enable**.

11. Click **OK** on each open window.

Loss of Control Center Function

In DB2 version 7.2, Version 6 Control Center clients prior to FixPak 6 and version 7 clients prior to FixPak 2 lose nearly all functionality when used with DB2 version 7.2. To fix this, upgrade your Version 6 clients to FixPak 6 or later, and your Version 7 clients to FixPak 2 or later.

There should be no problems introduced against downlevel Control Center clients by applying FixPak 2 to a DB2 server.

Netscape CD not Shipped with DB2 UDB

The Netscape CD is no longer being shipped with DB2 UDB. Netscape products are available from http://www.netscape.com.

Error in XML Readme Files

The README.TXT file for DB2 XML Extender Version 7.1 says the following under "Considerations":

   3. The default version of DB2 UDB is DB2 UDB Version 7.1. If you wish to use DB2 UDB Version 6.1 on AIX and Solaris systems, you should ensure that you are running with DB2 UDB V6.1 instance and with the DB2 UDB V6.1 libraries.

This is incorrect. **The DB2 XML Extender is supported only with DB2 Version 7.1 and 7.2.**

The files readme.aix, readme.nt, and readme.sun list Software Requirements of:
- DB2 UDB 6.1 with FP1_U465423 or higher (AIX)
- DB2 Universal Database Version 6.1 or higher with FixPak 3 installed (NT)
- DB2 UDB Version 6.1 with FixPak FP1_U465424 or higher (Sun)

This is incorrect. **The DB2 XML Extender requires DB2 Version 7.1 or 7.2.**

New Business Intelligence Enhancements in DB2 Version 7.2

In the Version 7.2 What's New book and some other documentation, reference is made to new Business Intelligence enhancements that have been added in Version 7.2. These enhancements will be made available at a later date.

FixPak 2A and Later Causes Problems in IBM DB2 OLAP Server

If you use IBM DB2 OLAP Server on UNIX, you might encounter problems with DB2 OLAP Server after you install FixPak 2A of DB2 Universal Database V7. FixPak 2A, and later FixPaks, install new ODBC drivers that support UNICODE, but DB2 OLAP Server does not support these new drivers. The workaround for DB2 OLAP Server is to switch your ODBC files to point to the non-UNICODE ODBC drivers.

The non-UNICODE drivers have been renamed to add "_36" in their names. For example, for the Solaris Operating Environment, the driver libdb2.so was renamed to libdb2_36.so. For more information about changing ODBC drivers, see "Loading and Configuring ODBC for the SQL Interface" in Chapter 4, "Installing on AIX, Solaris Operating Environment, and HP-UX," of the *OLAP Setup and User's Guide*.

Segmentation Violation When Using WebSphere 3.5.5

If you are running the WebSphere 3.5.5 user profile sample with DB2 V7.2 FixPak 4 or later on Linux390, you might receive a SIGSEGV 11 (*) segmentation violation. This problem relates to a defect in the JDK, and will occur with both JDK 1.2.2 and JDK 1.3.

The problem may additionally affect other JDBC applications.

The November service release of JDK 1.2.2 fixes this problem. JDK 1.3 will be fixed in its January service release.

A workaround for this problem is to turn off the JIT with the following command:

```
export JAVA_COMPILER=NONE
```

Veritas AIX Volume Manager Support

DB2 UDB Enterprise Edition, FixPak 7 or later, can be used with Veritas AIX Volume Manager Version 3.2 on AIX 5.1 ML 2 or later. Use of Veritas AIX Volume Manager with any other versions of DB2 UDB, including Enterprise Extended Edition, is not supported at this time.

Fix Required for Java Applications on AIX V4

Java application running on AIX 4.3.3 may terminate unexpectedly if the kernel fileset bos.mp or bos.up are at 4.3.3.77.

Run the command **lslpp –l bos.ip bos.mp** to determine the kernel fix level.

It is recommended that all Java customers running on AIX 4.3.3 upgrade to 4.3.3.78. A fix is available that will update the kernel to suggested level. You will need to obtain the fix for Authorized Problem Analysis Report (APAR) IY25282.

Installing the fix for IY25282 will correct the Java termination condition. A further fix will be released in the first quarter of 2002 using APAR number IY26149.

Note that AIX 5.1C ships with IY25377, which contains the same fix.

Fur further information and advice, contact AIX Support.

db2stop Hangs on AIX 5 Operating Systems due to an NFS problem

If you are using AIX 5, the db2stop command may hang if your system has a large number of database partitions. A workaround for this problem is to stop each partition separately using the NODENUM option of the db2stop command. The problem is fixed by AIX APAR IY32512.

Online Documentation (HTML, PDF, and Search) Notes

Supported Web Browsers on the Windows 2000 Operating System

We recommend that you use Microsoft Internet Explorer on Windows 2000.

If you use Netscape, please be aware of the following:

- DB2 online information searches may take a long time to complete on Windows 2000 using Netscape. Netscape will use all available CPU resources and appear to run indefinitely. While the search results may eventually return, we recommend that you change focus by clicking on another window after submitting the search. The search results will then return in a reasonable amount of time.

- You may notice that online help initially displays correctly in a Netscape browser window, but may not appear if you try to access it from a different part of the Control Center without first closing the browser window. If you close the browser window and request help again, the correct help comes up. You may be able to fix this problem by following the steps in "Error Messages when Attempting to Launch Netscape" on page 10. You can also get around the problem by closing the browser window before requesting help for the Control Center.

- When you request Control Center help, or a topic from the Information Center, you may get an error message. To fix this, follow the steps in "Error Messages when Attempting to Launch Netscape" on page 10.

Searching the DB2 Online Information on the Solaris Operating Environment

If you are having problems searching the DB2 online information on Solaris operating environments, check your system's kernel parameters in /etc/system. Here are the minimum kernel parameters required by DB2's search system, NetQuestion:

```
semsys:seminfo_semmni 256
semsys:seminfo_semmap 258
semsys:seminfo_semmns 512
semsys:seminfo_semmnu 512
semsys:seminfo_semmsl 50
shmsys:shminfo_shmmax 6291456
shmsys:shminfo_shmseg 16
shmsys:shminfo_shmmni 300
```

To set a kernel parameter, add a line at the end of /etc/system as follows:

```
set <semaphore_name> = value
```

You must reboot your system for any new or changed values to take effect.

Switching NetQuestion for OS/2 to Use TCP/IP

The instructions for switching NetQuestion to use TCP/IP on OS/2 systems are incomplete. The location of the `*.cfg` files mentioned in those instructions is the data subdirectory of the NetQuestion installation directory. You can determine the NetQuestion installation directory by entering one of the following commands:

```
echo %IMNINSTSRV%    //for SBCS installations
echo %IMQINSTSRV%    //for DBCS installations
```

Error Messages when Attempting to Launch Netscape

If you encounter the following error messages when attempting to launch Netscape:

```
Cannot find file <file path> (or one of its components).
  Check to ensure the path and filename are correct and that all
  required libraries are available.

Unable to open "D:\Program Files\SQLLIB\CC\..\doc\html\db2help\XXXXX.htm"
```

you should take the following steps to correct this problem on Windows NT, 95, or 98 (see below for what to do on Windows 2000):

1. From the Start menu, select Programs —> Windows Explorer. Windows Explorer opens.
2. From Windows Explorer, select View —> Options. The Options Notebook opens.
3. Click the **File types** tab. The File types page opens.
4. Highlight Netscape Hypertext Document in the **Registered file types** field and click **Edit**. The Edit file type window opens.
5. Highlight ″Open″ in the **Actions** field.
6. Click the **Edit** button. The Editing action for type window opens.
7. Uncheck the **Use DDE** check box.
8. In the **Application used to perform action** field, make sure that ″%1″ appears at the very end of the string (include the quotation marks, and a blank space before the first quotation mark).

If you encounter the messages on Windows 2000, you should take the following steps:

1. From the **Start** menu, select **Windows Explorer**. Windows Explorer opens.
2. From Windows Explorer, select **Tools —> Folder Options**. The Folder Options notebook opens.

3. Click the **File Types** tab.

4. On the File Types page, in the **Registered file types** field, highlight: HTM Netscape Hypertext Document and click **Advanced**. The Edit File Type window opens.

5. Highlight ″open″ in the **Actions** field.

6. Click the **Edit** button. The Editing Action for Type window opens.

7. Uncheck the **Use DDE** check box.

8. In the **Application used to perform action** field, make sure that ″%1″ appears at the very end of the string (include the quotation marks, and a blank space before the first quotation mark).

9. Click **OK**.

10. Repeat steps 4 through 8 for the HTML Netscape Hypertext Document and SHTML Netscape Hypertext Document file types.

Configuration Requirement for Adobe Acrobat Reader on UNIX Based Systems

Acrobat Reader is only offered in English on UNIX based platforms, and errors may be returned when attempting to open PDF files with language locales other than English. These errors suggest font access or extraction problems with the PDF file, but are actually due to the fact that the English Acrobat Reader cannot function correctly within a UNIX non-English language locale.

To view such PDF files, switch to the English locale by performing one of the following steps before launching the English Acrobat Reader:

- Edit the Acrobat Reader's launch script, by adding the following line after the #!/bin/sh statement in the launch script file:

  ```
  LANG=C;export LANG
  ```

  This approach will ensure correct behavior when Acrobat Reader is launched by other applications, such as Netscape Navigator, or an application help menu.

- Enter LANG=C at the command prompt to set the Acrobat Reader's application environment to English.

For further information, contact Adobe Systems (http://www.Adobe.com).

SQL Reference is Provided in One PDF File

The ″Using the DB2 Library″ appendix in each book indicates that the SQL Reference is available in PDF format as two separate volumes. This is incorrect.

Although the printed book appears in two volumes, and the two corresponding form numbers are correct, there is only one PDF file, and it contains both volumes. The PDF file name is db2s0x70.

Part 2. Installation and Configuration

General Installation, Migration and Configuration Information

Downloading Installation Packages for All Supported DB2 Clients

To download installation packages for all supported DB2 clients, which include all the pre-Version 7 clients, connect to the IBM DB2 FixPaks and Clients Web site at http://www.ibm.com/cgi-bin/db2www/data/db2/udb/winos2unix/support/download.d2w/report.

Making the DB2 EE or DB2 Connect EE Install Image accessible on Linux on S/390

Before proceeding with the instructions concerning this install as discussed in the Quick Beginnings books, create a tar file from the CD contents using the **tar -cvf** command, and place this tar file on a machine that is accessible from the S/390 machine.

DB2 Connect Appendix Information Not Required

The list mapping *Y* values to platforms in the DB2 Connect Quick Beginnings appendix "List Files, Bind Files, and Packages" is not required and should be disregarded.

Installing DB2 on SuSE Linux

Disregard the section "Installing DB2 on SuSE Linux" of Chapter 4 of *DB2 Universal Database for UNIX Quick Beginnings*. The prerequisite outlined in this section is no longer required.

Additional Required Solaris Operating Environment Patch Level

DB2 Universal Database Version 7 for Solaris Operating Environment Version 2.6 requires patch 106285-02 or higher, in addition to the patches listed in the *DB2 for UNIX Quick Beginnings* manual.

Installing DB2 Enterprise-Extended Edition on AIX

Step 4, in the Performing the Installation section of the DB2 Enterprise-Extended Edition for UNIX Quick Beginnings manual states that you should allocate a CD-ROM file system by entering the following command:

```
crfs -v cdrfs -p ro -d cd0
```

For this command to complete successfully, you must also specify the mount point using the —m command.

```
crfs -v cdrfs -p ro -d cd0 -m /cdrom
```

There is also a step missing in the Performing the Installation section. After Step 5, mounting the CD-ROM file system on the control workstation, each node that will participate in your partitioned database system should remotely mount the CD-ROM file system. Assuming that /cdrom does not already exist on participating nodes, use the following commands to export and remotely mount the /cdrom file system on the control workstation:

```
exportfs -i -o ro /cdrom
dsh mkdir /cdrom
dsh mount cws_hostname: /cdrom /cdrom
```

where *cws_hostname* is the host name of the control workstation.

Additional Installation Steps for AIX CICS Users

If you are installing DB2 UDB or any DB2 UDB FixPak on an AIX CICS system, you must carry out the following additional steps after installation. These steps are detailed in the *CICS/6000 Administration Guide*, in the "Configurarion steps for Database 2" section:

1. Create a DB2 UDB for AIX shared object from the libdb2.a library.
2. Build the DB2 switchload file and place it in the directory specified by the XA definition for the database.
3. If you are making use of COBOL in your environment, then re-run the cocsmkcobol tool.

Netscape LDAP directory support

DB2 supports the use of an LDAP directory for central administration and consolidation of database and node directories. In previous releases of DB2, only Microsoft Active Directory and IBM SecureWay Directory were supported. DB2 now also supports the following LDAP Servers: Netscape Directory Server v4.12 or later, iPlanet™ Directory Server 5.0 or later

Extending the Netscape LDAP schema

The following instructions are for Netscape Directory Server 4.1:

The Netscape Directory Server allows applications to extend the schema by adding attribute and object class definitions into the following two files, slapd.user_oc.conf and slapd.user_at.conf. These two files are located in the <Netscape_install path>\slapd-<machine_name>\config directory.

The DB2 attributes must be added to the slapd.user_at.conf as follows

Note: In this context, bin, cis, ces, and dn stand for binary, case insensitive string, case sensitive string, and distinguished name, respectively.

:

```
#############################################################################
#
# IBM DB2 Universal Database V7.2
# Attribute Definitions
#
#############################################################################

attribute binProperty                   1.3.18.0.2.4.305    bin
attribute binPropertyType               1.3.18.0.2.4.306    cis
attribute cesProperty                   1.3.18.0.2.4.307    ces
attribute cesPropertyType               1.3.18.0.2.4.308    cis
attribute cisProperty                   1.3.18.0.2.4.309    cis
attribute cisPropertyType               1.3.18.0.2.4.310    cis
attribute propertyType                  1.3.18.0.2.4.320    cis
attribute systemName                    1.3.18.0.2.4.329    cis
attribute db2nodeName                   1.3.18.0.2.4.419    cis
attribute db2nodeAlias                  1.3.18.0.2.4.420    cis
attribute db2instanceName               1.3.18.0.2.4.428    cis
attribute db2Type                       1.3.18.0.2.4.418    cis
attribute db2databaseName               1.3.18.0.2.4.421    cis
attribute db2databaseAlias              1.3.18.0.2.4.422    cis
attribute db2nodePtr                    1.3.18.0.2.4.423    dn
attribute db2gwPtr                      1.3.18.0.2.4.424    dn
attribute db2additionalParameters       1.3.18.0.2.4.426    cis
attribute db2ARLibrary                  1.3.18.0.2.4.427    cis
attribute db2authenticationLocation     1.3.18.0.2.4.425    cis
attribute db2databaseRelease            1.3.18.0.2.4.429    cis
attribute DCEPrincipalName              1.3.18.0.2.4.443    cis
```

The DB2 object classes must be added to the slapd.user_oc.conf file as follows:

```
#############################################################################
#
# IBM DB2 Universal Database V7.2
# Object Class Definitions
#
#############################################################################

objectclass eProperty
        oid 1.3.18.0.2.6.90
        requires
                objectClass
        allows
                cn,
                propertyType,
                binProperty,
                binPropertyType,
                cesProperty,
                cesPropertyType,
                cisProperty,
```

```
                    cisPropertyType

          objectclass eApplicationSystem
                 oid 1.3.18.0.2.6.8
                 requires
                         objectClass,
                         systemName


          objectclass DB2Node
                 oid 1.3.18.0.2.6.116
                 requires
                         objectClass,
                         db2nodeName
                 allows
                         db2nodeAlias,
                         host,
                         db2instanceName,
                         db2Type,
                         description,
                         protocolInformation
          objectclass DB2Database
                 oid 1.3.18.0.2.6.117
                 requires
                         objectClass,
                         db2databaseName,
                         db2nodePtr
                 allows
                         db2databaseAlias,
                         description,
                         db2gwPtr,
                         db2additionalParameters,
                         db2authenticationLocation,
                         DCEPrincipalName,
                         db2databaseRelease,
                         db2ARLibrary
```

After adding the DB2 schema definition, the Directory Server must be
restarted for all changes to be active.

Support for Windows ME, Windows XP and Windows 2000 Datacenter Edition Platforms

DB2 now supports Microsoft Windows ME, Windows XP, and Windows 2000
Datacenter Edition platforms. Following is additional platform-specific
information.

Windows XP

The following products and versions support 32–bit Windows XP when
installed with FixPak 4 or later:

- IBM DB2 UDB Personal Edition Version 7.2

- IBM DB2 Personal Developer's Edition Version 7.2
- IBM DB2 Universal Developer's Edition Version 7.2
- IBM DB2 Connect Personal Edition Version 7.2
- IBM DB2 Connect Enterprise Edition Version 7.2
- IBM DB2 UDB Workgroup Edition Version 7.2
- IBM DB2 UDB Enterprise Edition Version 7.2
- IBM DB2 Run-Time Client Version 7.2
- IBM DB2 Administration Client Version 7.2
- IBM DB2 Application Development Client Version 7.2

DB2 supports the same national languages on Windows XP systems as on other versions.

Limitations

When entering user IDs and passwords during installation, you may receive a message that a user account entered in the install panel is not valid, even though it is valid. This will only happen with user IDs that exist on the machine. You should not have this problem you enter user names that do not yet exist.

If you choose to install DB2 under any user account but db2admin, then you must ensure that the account name conforms to DB2 naming rules. Most importantly, the name must not contain any spaces. For example my_name is acceptable, but my name is not.

If you receive error 1052 during the product installation, then do the following:

1. Leave the error window open.
2. Open a command window.
3. Run the command **db2start.exe**.
4. Run the command specified in the error window, using the password specified for the user shown on the command line.
5. Return to the error window, and click **OK**. The install will now continue.

If you are using Simplified Chinese and find that fonts in the Control Center do not display properly, modify the sqllib\java\java12\jdk\jre\lib\font.properties.zh by replacing the entry filename.\u5b8b\u4f53=simsun.ttf with filename.\u5b8b\u4f53=simsun.ttc..

Windows ME

The following products and versions support Windows ME when installed with FixPak 2 or later:

- IBM DB2 UDB Personal Edition Version 7.1

- IBM DB2 Personal Developer's Edition Version 7.1
- IBM DB2 Universal Developer's Edition Version 7.1
- IBM DB2 Connect Personal Edition Version 7.1
- IBM DB2 Run-Time Client Version 7.1
- IBM DB2 Administration Client Version 7.1
- IBM DB2 Application Development Client Version 7.1

Limitations

The HTML Search Server capability is not supported on Window ME at this time

When you uninstall DB2, you may receive an error message indicating that the file MFC42U.DLL cannot be found. To fully uninstall DB2, manually delete the sqllib directory after the uninstall activity completes.

Windows 2000 Datacenter Server

The following DB2 products are certified for Windows 2000 Datacenter Server, Windows 2000 Advanced Server, and Windows 2000 Server:

- IBM DB2 Universal Database Enterprise - Extended Edition Version 7.2
- IBM DB2 Universal Database Enterprise Edition 7.2
- IBM DB2 Database Workgroup Edition Version 7.2
- IBM DB2 Connect Enterprise Edition Version 7.2

Installing DB2 in Windows 95

If you are installing DB2 on a non-English Windows 95 system, you need to manually update your version of Winsock to Winsock 2 *before* installing DB2 UDB. The Winsock 2 upgrade utility is available from Microsoft.

Installing DB2 on Windows 2000

On Windows 2000, when installing over a previous version of DB2 or when reinstalling the current version, ensure that the recovery options for all of the DB2 services are set to "Take No Action".

Running DB2 under Windows 2000 Terminal Server, Administration Mode

For DB2 UDB Version 7.1, FixPak 3 and later, DB2 can run under the Windows 2000 Terminal Server, Administration Mode. Prior to this, DB2 only supported the Application Server mode of Windows 2000 Terminal Server.

Microsoft SNA Server and SNA Multisite Update (Two Phase Commit) Support

Host and AS/400 applications cannot access DB2 UDB servers using SNA two phase commit when Microsoft SNA Server is the SNA product in use. Any DB2 UDB publications indicating this is supported are incorrect. IBM Communications Server for Windows NT Version 5.02 or greater is required.

Note: Applications accessing host and AS/400 database servers using DB2 UDB for Windows *can* use SNA two phase commit using Microsoft SNA Server Version 4 Service Pack 3 or greater.

Define User ID and Password in IBM Communications Server for Windows NT (CS/NT)

If you are using APPC as the communication protocol for remote DB2 clients to connect to your DB2 server and if you use CS/NT as the SNA product, make sure that the following keywords are set correctly in the CS/NT configuration file. This file is commonly found in the `x:\ibmcs\private` directory.

Node Definition

TG_SECURITY_BEHAVIOR
> This parameter allows the user to determine how the node is to handle security information present in the ATTACH if the TP is not configured for security

IGNORE_IF_NOT_DEFINED
> This parameter allows the user to determine if security parameters are present in the ATTACH and to ignore them if the TP is not configured for security.
>
> If you use IGNORE_IF_NOT_DEFINED, you don't have to define a User ID and password in CS/NT.

VERIFY_EVEN_IF_NOT_DEFINED
> This parameter allows the user to determine if security parameters are present in the ATTACH and verify them even if the TP is not configured for security. This is the default.
>
> If you use VERIFY_EVEN_IF_NOT_DEFINED, you have to define User ID and password in CS/NT.

To define the CS/NT User ID and password, perform the following steps:

1. **Start —> Programs —> IBM Communications Server —> SNA Node Configuration**. The Welcome to Communications Server Configuration window opens.
2. Choose the configuration file you want to modify. Click **Next**. The Choose a Configuration Scenario window opens.

3. Highlight CPI-C, APPC or 5250 Emulation. Click **Finish**. The Communications Server SNA Node Window opens.

4. Click the [+] beside CPI-C and APPC.

5. Click the [+] beside LU6.2 Security.

6. Right click on User Passwords and select **Create**. The Define a User ID Password window opens.

7. Fill in the User ID and password. Click **OK**. Click **Finish** to accept the changes.

DB2 Install May Hang if a Removable Drive is Not Attached

During DB2 installation, the install may hang after selecting the install type when using a computer with a removable drive that is not attached. To solve this problem, run setup, specifying the -a option:

```
setup.exe -a
```

Error SQL1035N when Using CLP on Windows 2000

If DB2 is installed to a directory to which only some users (e.g. administrators) have write access, a regular user may receive error SQL1035N when attempting to use the DB2 Command Line Processor.

To solve this problem, DB2 should be installed to a directory to which all users have write access.

Migration Issue Regarding Views Defined with Special Registers

Views become unusable after database migration if the special register USER or CURRENT SCHEMA is used to define a view column. For example:

```
create view v1 (c1) as values user
```

In Version 5, USER and CURRENT SCHEMA were of data type CHAR(8), but since Version 6, they have been defined as VARCHAR(128). In this example, the data type for column c1 is CHAR if the view is created in Version 5, and it will remain CHAR after database migration. When the view is used after migration, it will compile at run time, but will then fail because of the data type mismatch.

The solution is to drop and then recreate the view. Before dropping the view, capture the syntax used to create it by querying the SYSCAT.VIEWS catalog view. For example:

```
select text from syscat.views where viewname='<>'
```

IPX/SPX Protocol Support on Windows 2000

This information refers to the Planning for Installation chapter in your Quick Beginnings book, in the section called "Possible Client-to-Server Connectivity Scenarios."

The published protocol support chart is not completely correct. A Windows 2000 client connected to any OS/2 or UNIX based server using IPX/SPX is not supported. Also, any OS/2 or UNIX based client connected to a Windows 2000 server using IPX/SPX is not supported.

Stopping DB2 Processes Before Upgrading a Previous Version of DB2

This information refers to the migration information in your DB2 for Windows Quick Beginnings book.

If you are upgrading a previous version of DB2 that is running on your Windows machine, the installation program provides a warning containing a list of processes that are holding DB2 DLLs in memory. At this point, you have the option to manually stop the processes that appear in that list, or you can let the installation program shut down these processes automatically. It is recommended that you manually stop all DB2 processes before installing to avoid loss of data. The best way to ensure that DB2 processes are not running is to view your system's processes through the Windows Services panel. In the Windows Services panel, ensure that there are no DB2 services, OLAP services, or Data warehouse services running.

Note: You can only have one version of DB2 running on Windows platforms at any one time. For example, you cannot have DB2 Version 7 and DB2 Version 6 running on the same Windows machine. If you install DB2 Version 7 on a machine that has DB2 Version 6 installed, the installation program will delete DB2 Version 6 during the installation. Refer to the appropriate *Quick Beginnings* manual for more information on migrating from previous versions of DB2.

Run db2iupdt After Installing DB2 If Another DB2 Product is Already Installed

The following information should have been available in your Quick Beginnings installation documentation.

When installing DB2 UDB Version 7 on UNIX based systems, and a DB2 product is already installed, you will need to run the **db2iupdt** command to update those instances with which you intend to use the new features of this product. Some features will not be available until this command is run.

Setting up the Linux Environment to Run the DB2 Control Center

This information should be included with the "Installing the DB2 Control Center" chapter in your Quick Beginnings book.

After leaving the DB2 installer on Linux and returning to the terminal window, type the following commands to set the correct environment to run the DB2 Control Center:

```
su -l <instance name>
export JAVA_HOME=/usr/jdk118
export DISPLAY=<your machine name>:0
```

Then, open another terminal window and type:

```
su root
xhost +<your machine name>
```

Close that terminal window and return to the terminal where you are logged in as the instance owner ID, and type the command:

```
db2cc
```

to start the Control Center.

DB2 Universal Database Enterprise Edition and DB2 Connect Enterprise Edition for Linux on S/390

DB2 Universal Database Enterprise Edition and DB2 Connect Enterprise Edition are now available for Linux on S/390. Before installing Linux on an S/390 machine, you should be aware of the software and hardware requirements:

Hardware

S/390 9672 Generation 5 or higher, Multiprise 3000.

Software
- SuSE SLES-7 with the patches listed below or Turbolinux Server 6
- kernel level 2.2.16, with patches for S/390 (see below)
- glibc 2.1.3
- libstdc++ 6.1 (included in the compat.rpm package)

The following patches are required for Linux on S/390:
- SLES-7-PatchCD-1-s390-20020522.iso

For latest updates on supported software for S/390 Linux systems, visit the website http://www.ibm.com/db2/linux/validate.

Notes:

1. Only 32-bit Intel-based Linux and Linux on S/390 are supported.
2. The following are *not* available on Linux/390 in DB2 Version 7:
   - DB2 UDB Enterprise - Extended Edition
   - DB2 Extenders
   - DB2 Data Links Manager
   - DB2 Administrative Client
   - Change Password Support
   - LDAP Support
   - TSM
   - Use of raw devices

Possible Data Loss on Linux for S/390

When using DB2 on Linux for S/390 with a 2.2 series kernel, the amount of available RAM on the Linux machine should be limited to less than 1 GB. Limiting the RAM to 1 GB will avoid possible data loss in DB2 due to a Linux kernel bug.

This only affects DB2 on Linux for S/390 and not Linux on Intel.

A kernel patch will be made available at the IBM developerworks site, after which it will be possible to use more than 1 GB of RAM.

Gnome and KDE Desktop Integration for DB2 on Linux

DB2 now includes a set of utilities for the creation of DB2 desktop folders and icons for launching the most commonly used DB2 tools on the Gnome and KDE desktops for supported Intel-based Linux distributions. These utilities are installed by DB2 Version 7.2 by default, and can be used after the installation to create and remove desktop icons for one or more selected users.

To add a set of desktop icons for one or more users, use the following command:

```
db2icons <user1> [<user2> <user3>...]
```

Note: Note that if icons are generated while a Gnome or KDE desktop environment is running, the user may need to force a manual desktop refresh to see the new icons.

To remove a set of desktop icons for one or more users, use the following command:

```
db2rmicons <user1> [<user2> <user3>...]
```

> **Note:** You must have sufficient authority to generate or remove icons for other users. Typically, **db2icons** and **db2rmicons** can be used to create or remove icons for yourself if you are a normal user, and for others only if you are root, or another user with the authority to write to the specified users home directories.

Solaris Kernel Configuration Parameters (Recommended Values)

The Before You Begin section in the Solaris system chapter of the *DB2 for UNIX Quick Beginnings* and *DB2 Enterprise - Extended Edition Quick Beginnings for UNIX* provides recommended Solaris kernel configuration parameters. The following table provides additional kernel configuration-parameter recommendations for systems with more than 512 MB of real memory.

Table 1. Solaris Kernel Configuration Parameters (Recommended Values)

Kernel Parameter	512 MB-1 GB	1 GB-4 GB	4 GB+
msgsys:msginfo_msgmax	65,535	65,535	65,535
msgsys:msginfo_msgmnb	65,535	65,535	65,535
msgsys:msginfo_msgmap	514	1,026	2,050
msgsys:msginfo_msgmni	512	1,024	2,048
msgsys:msginfo_msgssz	16	32	64
msgsys:msginfo_msgtql	1,024	2,048	4,096
msgsys:msginfo_msgseg	32,767	32,767	32,767
shmsys:shminfo_shmmax	483,183,820 – 966,367,641	966,367,641 – 3,865,470,566	3,865,470,566 – 4,294,967,296
shmsys:shminfo_shmseg	50	100	200
shmsys:shminfo_shmmni	300	1,024	2,048
semsys:seminfo_semmni	1,024	2,048	4,198
semsys:seminfo_semmap	1,026	2,050	4,096
semsys:seminfo_semmns	2,048	4,096	8,192
semsys:seminfo_semmnu	2,048	4,096	8,192
semsys:seminfo_semume	50	50	50

Notes:

1. The limit of the shmsys:shminfo_shmmax parameter is 4 GB for 32–bit systems.
2. The msgsys:msginfo_msgmnb and msgsys:msginfo_msgmax parameters must be set to 65,535 or larger.
3. The msgsys:msginfo_msgseg parameter must be set no higher than 32,767.

4. The shmsys:shminfo_shmmax parameters should be set to the suggested value in the above table or to 90% of the physical memory (in bytes), whichever is higher. For example, if you have 196 MB of physical memory in your system, set the shmsys:shminfo_shmmax parameter to 184,968,806 (196*1024*1024*0.9).

DB2 Universal Database Enterprise - Extended Edition for UNIX Quick Beginnings

Chapter 5. Installing and Configuring DB2 Universal Database on Linux should indicate that each physical node in a Linux EEE cluster must have the same kernel, glibc, and libstdc++ levels.

A trial version of DB2 EEE for Linux can be downloaded from the following Web site: http://www6.software.ibm.com/dl/db2udbdl/db2udbdl-p

shmseg Kernel Parameter for HP-UX

Information about updating the HP-UX kernel configuration parameters provided in your Quick Beginnings book is incorrect. The recommended value for the shmseg kernel parameter for HP-UX should be ignored.

The default HP-UX value (120) should be used instead.

Migrating IBM Visual Warehouse Control Databases

DB2 Universal Database Quick Beginnings for Windows provides information about how the active warehouse control database is migrated during a typical install of DB2 Universal Database Version 7 on Windows NT and Windows 2000. If you have more than one warehouse control database to be migrated, you must use the Warehouse Control Database Management window to migrate the additional databases. Only one warehouse control database can be active at a time. If the last database that you migrate is not the one that you intend to use when you next log on to the Data Warehouse Center, you must use the Warehouse Control Database Management window to register the database that you intend to use.

Migrating Unique Indexes Using the db2uiddl Command

In the DB2 Post-installation Migration Tasks chapter of the DB2 Quick Beginnings manuals, under Optional Post-Migration Tasks, it is stated that you must use the **db2uiddl** command to migrate unique indexes from DB2 version 5.x and DB2 version 6. This is incorrect. Migration of unique indexes using the **db2uiddl** command is only required if you are migrating from a version of DB2 that is pre-version 5.

64-bit AIX Version Installation Error

When using **db2setup** to install a 64-bit AIX DB2 image on an existing AIX operating system, ensure that you are using compatible AIX versions or the installation will fail. AIX Version 5 DB2 images cannot be installed on an existing AIX Version 4 operating system. Similarly, the installation of a 64-bit AIX Version 4 DB2 image on an existing AIX Version 5 operating system will also result in an installation error.

When attempting to install incompatible 64-bit AIX versions, the db2setup utility finds the version mismatch on a prerequisites check resulting in an error message such as the following:

```
DBI1009E Install media and AIX version mismatch.
```

To avoid this error, ensure that you are installing the correct 64-bit AIX version.

Using SMIT

If you use SMIT, you will receive an error for AIX Version 4 DB2 being replaced by AIX Version 5 DB2, however, the reverse does not occur. Therefore, 64-bit AIX Version 5 users should ensure that they are installing the correct version. If db2setup can be launched, without an error message, then the checking of AIX version compatibility was successful.

Note: This incompatibility error is not applicable to 32-bit AIX versions.

Errors During Migration

During migration, error entries in the db2diag.log file (database not migrated) appear even when migration is successful, and can be ignored.

When using Warehouse Control Database Management, errors are logged in the SQLLIB\LOGGING directory. The IWH2RGn.LOG files contain any error information. If there is an error you must correct the error, delete the control database and start again. In the case of an existing control database, this means you need to use the backup copy.

IBM® DB2® Connect License Activation

The installation programs for DB2 Connect Enterprise Edition, DB2 Connect Unlimited Edition, and DB2 Connect Web Starter Kit do not install the product licenses. After installation, these products will operate in the Try-and-Buy mode for a period of 90 days since they do not have the license files. After the 90-day period, the product that you installed will stop functioning unless you activate the proper license.

To activate a license for your product you can use either the DB2 License Center or the **db2licm** command.

Installing Your License Key and Setting the License Type Using the License Center

1. Start the DB2 Control Center and select **License Center** from the **Tools** menu.
2. Select the system for which you are installing a license. The **Installed Products** field will display the name of the product that you have installed.
3. Select **Add** from the **License** menu.
4. In the Add License window, select the **From a file** radio button and select a license file:
   - On Windows servers: `x:\db2\license\connect\`*license_filename* where `x:` represents the CD-ROM drive containing DB2 Connect product CD.
   - On UNIX servers: `/db2/license/connect/`*license_filename*

   where *license_filename* for DB2 Connect Enterprise Edition and DB2 Connect Unlimited Edition is **db2conee.lic**, and for DB2 Connect Web Starter Kit is **db2consk.lic**.
5. Click **Apply** to add the license key.
6. Setting the license type.
   - For DB2 Connect Unlimited Edition and DB2 Connect Web Starter Kit:

     In the License Center, select **Change** from the **License** menu. In the Change License window, select the **Measured usage** check box. Click **OK** to close the Change License window and return to the License Center.

     Note: For the DB2 Connect Web Starter Kit, ensure that the expiry date is set to 270 days from the day you installed the product.
   - For DB2 Connect Enterprise Edition:

     In the License Center, select **Change** from the **License** menu. In the Change License window, select the type of license that you have purchased.
     - If you purchased a Concurrent Users license, select **Concurrent DB2 Connect users** and enter the number of user licenses that you have purchased.

       Note: DB2 Connect Enterprise Edition provides a license for one user. Additional DB2 Connect User licenses must be purchased separately.

– If you purchased a Registered Users license, select **Registered DB2 Connect users** and click **OK** to close the Change License window and return to the License Center. Click on the **Users** tab and add every user ID for which you purchased a license.

Installing your License Key and Setting License Type Using the db2licm Command

You can use the **db2licm** command to add the license key instead of using the License Center. To add your license key using the **db2licm** command:

1. On Windows servers, enter the following command:

   ```
   db2licm -a x:\db2\license\connect\license_filename
   ```

   where x: represents the CD-ROM drive that contains the DB2 Connect product CD.

   On UNIX servers, enter the following command:

   ```
   db2licm -a db2/license/connect/license_filename
   ```

   where *license_filename* for DB2 Connect Enterprise Edition and DB2 Connect Unlimited Edition is db2conee.lic, and for DB2 Connect Web Starter Kit is db2consk.lic.

   Note: For the DB2 Connect Web Starter Kit, ensure that the expiry date is set to 270 days from the day you installed the product.

2. Setting the license type:
   - For DB2 Connect Unlimited Edition and DB2 Connect Web Starter Kit:

     Enter the following command:

     ```
     db2licm -p db2conee measured
     ```
   - For DB2 Connect Enterprise Edition:

     If you purchased Concurrent User licenses, enter the following commands:

     ```
     db2licm -p db2conee concurrent
     db2licm -u N
     ```

     where *N* represents the number of concurrent user licenses that you have purchased.

     If you purchased Registered User licenses, enter the following command:

     ```
     db2licm -p db2conee registered
     ```

License Considerations for Distributed Installations

If you are creating an image for a distributed installation, you need to make special arrangements to install the license after installation. Add the **db2licm** commands described above to your distributed installation scripts.

Accessing Warehouse Control Databases

In a typical installation of DB2 Version 7 on Windows NT, a DB2 Version 7 warehouse control database is created along with the warehouse server. If you have a Visual Warehouse warehouse control database, you must upgrade the DB2 server containing the warehouse control database to DB2 Version 7 before the metadata in the warehouse control database can be migrated for use by the DB2 Version 7 Data Warehouse Center. You must migrate any warehouse control databases that you want to continue to use to Version 7. The metadata in your active warehouse control database is migrated to Version 7 during the DB2 Version 7 install process. To migrate the metadata in any additional warehouse control databases, use the Warehouse Control Database Migration utility, which you start by selecting **Start —> Programs —> IBM DB2 —> Warehouse Control Database Management** on Windows NT. For information about migrating your warehouse control databases, see *DB2 Universal Database for Windows Quick Beginnings*.

IBM e-server p690 and DB2 UDB Version 7 with AIX 5

FixPak 6 in the minimum level of DB2 UDB Version 7 required for use with IBM e-server p690 on an AIX 5 operating system.

Trial Products on Enterprise Edition UNIX CD-ROMs

The DB2 Universal Database (UDB) Enterprise Edition (EE) CD-ROMs for UNIX platforms Version 6 and Version 7 contain a 90-day trial version of DB2 Connect Enterprise Edition (CEE). Because DB2 Connect functionality is built into the DB2 UDB EE product, you do not have to install the DB2 CEE product on systems where DB2 UDB EE is installed to use DB2 Connect functionality. If you install the 90-day trial version of DB2 CEE and decide to upgrade to a licensed version, you must purchase the DB2 CEE product and install the DB2 CEE license key. You do not have to reinstall the product. The instructions for installing the license key is provided in the DB2 EE or DB2 CEE for UNIX Quick Beginnings book.

If you installed the trial CEE product along with your EE installation, and do not want to install CEE permanently, you can remove the CEE 90-day trial version by following these instructions. If you remove the trial version of Connect EE, you will still have DB2 Connect functionality available with DB2 EE.

To remove DB2 Connect Version 7, uninstall the following filesets from the respective platforms:
- On AIX, uninstall the db2_07_01.clic fileset.
- On NUMA-Q and the Solaris Operating Environments, uninstall the db2clic71 package.
- On Linux, uninstall the db2clic71-7.1.0-x RPM.
- On HP-UX, uninstall the DB2V7CONN.clic fileset.

To remove DB2 Connect Version 6, uninstall the following filesets from the respective platforms:
- On AIX, uninstall the db2_06_01.clic fileset.
- On NUMA-Q and the Solaris Operating Environments, uninstall the db2cplic61 package.
- On Linux, uninstall the db2cplic61-6.1.0-x RPM.
- On HP-UX, uninstall the DB2V6CONN.clic fileset.

Trial Products on DB2 Connect Enterprise Edition UNIX CD-ROMs

The DB2 Connect Enterprise Edition (EE) CD-ROMs for UNIX platforms Version 6 and Version 7 contain a 90-day trial version of DB2 Universal Database (UDB) Enterprise Edition (EE). The DB2 UDB EE 90-day trial version is provided for evaluation, but is not required for DB2 Connect to work.

If you install the 90-day trial version of DB2 UDB EE and decide to upgrade to a licensed version, you must purchase the DB2 UDB EE product and install the DB2 UDB EE license key. You do not have to reinstall the product. The instructions for installing the license key are provided in the DB2 EE or DB2 CEE for UNIX Quick Beginnings book. If you installed the trial UDB EE product along with your Connect EE installation, and you do not want to install UDB EE permanently, you can remove the EE 90-day trial version by following these instructions. If you remove the trial version of DB2 UDB EE, it will not impact the functionality of DB2 Connect EE.

To remove DB2 UDB EE Version 7, uninstall the following filesets from the respective platforms:
- On AIX, uninstall the db2_07_01.elic fileset.
- On NUMA-Q and the Solaris Operating Environments, uninstall the db2elic71 package.
- On Linux, uninstall the db2elic71-7.1.0-x RPM.
- On HP-UX, uninstall the DB2V7ENTP.elic fileset.

To remove DB2 UDB EE Version 6, uninstall the following filesets from the respective platforms:

- On AIX, uninstall the db2_06_01.elic fileset.
- On NUMA-Q and the Solaris Operating Environments, uninstall the db2elic61 package.
- On Linux, uninstall the db2elic61-6.1.0-x RPM.
- On HP-UX, uninstall the DB2V6ENTP.elic fileset.

Merant Driver Manager and the DB2 UDB Version 7 ODBC Driver on UNIX

Incompatibilities have been encountered with Unicode support when the Merant Driver Manager accesses DB2's ODBC driver on UNIX. These incompatibilities result in Unicode being used by the Merant Driver Manager regardless of whether the application has requested its use. This can lead to problems with products 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. The affected versions of DB2 UDB include Version 7.1 with FixPak 2 or later, and Version 7.2 at any FixPak level.

An alternative DB2 ODBC driver library without Unicode support enabled was shipped with DB2 Versions 7.1 and 7.2 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) clibrary contains the Unicode functions required by the DB2 JDBC cdriver. Using this library will still allow JDBC applications, including WebSphere Application Server, to work successfully with DB2.

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.

AIX:

To create the necessary library on AIX:
1. As the instance owner, shut down all database instances using **db2stop force**.
2. As the admin instance ID, shut down the administration server instance using **db2admin stop force**.
3. Back up the original db2.o under /usr/lpp/db2_<ver>_<rel>/lib.
4. As root, issue **slibclean**.

5. Copy db2_36.o to 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 db2_36.o.

Solaris Operating Environment:

To create the necessary library:

1. As the instance owner, shut down all database instances using **db2stop force**.
2. As the admin instance ID, shut down the administration server instance using **db2admin stop force**.
3. Back up the original libdb2.so.1 under /opt/IBMdb2/V<ver>.<rel>/lib.
4. Copy libdb2_36.so.1 to 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 **db2iupdt <instance>** for each database instance and **dasiupt <das_instance>**for the DAS instance.

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

HP-UX – Only available for FixPak 4 or later.:

You must install FixPak 4 or later *before* attempting this workaround.

To create the necessary library on HP-UX:

1. Shut down all database instances using **db2stop force**.
2. Shut down the administration server instance using **db2admin stop force**.
3. Back up the original libdb2.sl under /opt/IBMdb2/V<ver>.<rel>/lib.
4. Copy libdb2_36.sl to libdb2.sl, ensuring that the ownership and permissions remain consistent. Use the following command to ensure the consistency:

```
-r-xr-xr-x bin:bin for libdb2.sl
```

5. Issue **db2iupdt <instance>** for each database instance and **dasiupdt <das_instance>** for the DAS instance.

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

Other UNIX Operating Systems:

If you require assistance with DB2 and the Merant Driver Manager on other UNIX operating systems, please contact IBM Support.

Additional Configuration Needed Before Installing the Information Catalog Center for the Web

Before installing the Information Catalog Center for the Web, you must copy the dg_strings.hti file for the language you are installing to the /sqllib/icuweb/macro directory. You can find the dg_strings.hti file in your corresponding language directory. A list of the language codes can be found in the NLS appendix of the Quick Beginnings Guides and most of the other DB2 documentation.

Code Page and Language Support Information – Correction

The Code page and Language Support table in the National Language Support (NLS) appendix of the Quick Beginnings manuals contains the following errors:

- The column heading, "Country Code", should read "Country/Region Code"
- The column heading, "Language", should read "Language/Script"
- The code for Slovenia is "sl", not "si" as indicated in the table

DB2 Data Links Manager Quick Beginnings

Support on AIX 5.1

The DB2 Data Links File Manager and File Filter components are now fully supported on AIX 5.1. All tools and instructions associated with Data Links and previously supported on prior releases of AIX are now fully supported and applicable on AIX 5.1.

Dlfm Start Fails with Message: ″**Error in getting the afsfid for prefix**″

For a Data Links Manager running in the DCE-DFS environment, contact IBM Service if dlfm start fails with the following error:

```
Error in getting the afsfid for prefix
```

The error may occur when a DFS file set registered to the Data Links Manager using ″dlfm add_prefix″ is deleted.

Setting Tivoli Storage Manager Class for Archive Files

To specify which TSM management class to use for the archive files, set the DLFM_TSM_MGMTCLASS DB2 registry entry to the appropriate management class name.

Disk Space Requirements for DFS Client Enabler

The DFS Client Enabler is an optional component that you can select during DB2 Universal Database client or server installation. You cannot install a DFS Client Enabler without installing a DB2 Universal Database client or server product, even though the DFS Client Enabler runs on its own without the need for a DB2 UDB client or server. In addition to the 2 MB of disk space required for the DFS Client Enabler code, you should set aside an additional 40 MB if you are installing the DFS Client Enabler as part of a DB2 Run-Time Client installation. You will need more disk space if you install the DFS Client Enabler as part of a DB2 Administration Client or DB2 server installation. For more information about disk space requirements for DB2 Universal Database products, refer to the *DB2 for UNIX Quick Beginnings* manual.

Monitoring the Data Links File Manager Back-end Processes on AIX

There has been a change to the output of the **dlfm see** command. When this command is issued to monitor the Data Links File Manager back-end processes on AIX, the output that is returned will be similar to the following:

```
     PID    PPID    PGID   RUNAME    UNAME      ETIME DAEMON NAME
   17500   60182   40838     dlfm     root      12:18 dlfm_copyd_(dlfm)
   41228   60182   40838     dlfm     root      12:18 dlfm_chownd_(dlfm)
   49006   60182   40838     dlfm     root      12:18 dlfm_upcalld_(dlfm)
   51972   60182   40838     dlfm     root      12:18 dlfm_gcd_(dlfm)
   66850   60182   40838     dlfm     root      12:18 dlfm_retrieved_(dlfm)
   67216   60182   40838     dlfm     dlfm      12:18 dlfm_delgrpd_(dlfm)
   60182       1   40838     dlfm     dlfm      12:18 dlfmd_(dlfm)

DLFM SEE request was successful.
```

The name that is enclosed within the parentheses is the name of the dlfm
instance, in this case "dlfm".

Installing and Configuring DB2 Data Links Manager for AIX: Additional Installation Considerations in DCE-DFS Environments

In the section called "Installation prerequisites", there is new information that
should be added:

```
You must also install either an e-fix for DFS 3.1,
or PTF set 1 (when it becomes available). The e-fix is available from:

http://www.transarc.com/Support/dfs/datalinks/efix_dfs31_main_page.html
```

Also:

```
The dfs client must be running before you install the Data Links Manager.
Use db2setup or smitty.
```

In the section called "Keytab file", there is an error that should be corrected
as:

```
The keytab file, which contains the principal and password information,
should be called datalink.ktb and ....
```

The correct name: datalink.ktb is used in the example below. The "Keytab
file" section should be moved under "DCE-DFS Post-Installation Task",
because the creation of this file cannot occur until after the DLMADMIN
instance has been created.

In the section called "Data Links File Manager servers and clients", it should
be noted that the Data Links Manager server must be installed before any of
the Data Links Manager clients.

A new section, "Backup directory", should be added:

```
If the backup method is to a local file system,
this must be a directory in the DFS file system.
Ensure that this DFS file set has been created by a
DFS administrator. This should not be a DMLFS file set.
```

Failed ″dlfm add_prefix″ Command

For a Data Links Manager running in the DCE/DFS environment, the **dlfm add_prefix** command might fail with a return code of -2061 (backup failed). If this occurs, perform the following steps:

1. Stop the Data Links Manager daemon processes by issuing the **dlfm stop** command.
2. Stop the DB2 processes by issuing the **dlfm stopdbm** command.
3. Get dce root credentials by issuing the **dce_login root** command.
4. Start the DB2 processes by issuing the **dlfm startdbm** command.
5. Register the file set with the Data Links Manager by issuing the **dlfm add_prefix** command.
6. Start the Data Links Manager daemon processes by issuing the **dlfm start** command.

In the Rare Event that the Copy Daemon Does Not Stop on dlfm stop

It could happen in very rare situations that dlfm_copyd (the copy daemon) does not stop when a user issues a dlfm stop, or there is an abnormal shutdown. If this happens, issue a dlfm shutdown before trying to restart dlfm.

Installing and Configuring DB2 Data Links Manager for AIX: Installing DB2 Data Links Manager on AIX Using the db2setup Utility

In the section ″DB2 database DLFM_DB created″, the DLFM_DB is not created in the DCE_DFS environment. This must be done as a post-installation step.

In the section ″DCE-DFS pre-start registration for DMAPP″, Step 2 should be changed to the following:

```
2. Commands are added to /opt/dcelocal/tcl/user_cmd.tcl to
   ensure that the DMAPP is started when DFS is started.
```

Installing and Configuring DB2 Data Links Manager for AIX: DCE-DFS Post-Installation Task

The following new section, ″Complete the Data Links Manager Install″, should be added:

On the Data Links Manager server, the following steps must be performed to complete the installation:

1. Create the keytab file as outlined under ″Keytab file″ in the section ″Additional Installation Considerations in DCE-DFS Environment″, in the chapter ″Installing and Configuring DB2 Data Links Manager for AIX″.

2. As root, enter the following commands to start the DMAPP:

   ```
   stop.dfs all
   start.dfs all
   ```

3. Run "dlfm setup" using dce root credentials as follows:

   a. Login as the Data Links Manager administrator, DLMADMIN.

   b. As root, issue dce_login.

   c. Enter the command: dlfm setup.

On the Data Links Manager client, the following steps must be performed to complete the installation:

1. Create the keytab file as outlined under "Keytab file" in the section "Additional Installation Considerations in DCE-DFS Environment", in the chapter "Installing and Configuring DB2 Data Links Manager for AIX".

2. As root, enter the following commands to start the DMAPP:

   ```
   stop.dfs all
   start.dfs all
   ```

Installing and Configuring DB2 Data Links Manager for AIX: Manually Installing DB2 Data Links Manager Using Smit

Under the section, "SMIT Post-installation Tasks", modify step 7 to indicate that the command "dce_login root" must be issued before "dlfm setup". Step 11 is not needed. This step is performed automatically when Step 6 (dlfm server_conf) or Step 8 (dlfm client_conf) is done. Also remove step 12 (dlfm start). To complete the installation, perform the following steps:

1. Create the keytab file as outlined under "Keytab file" in the section "Additional Installation Considerations in DCE-DFS Environment", in the chapter "Installing and Configuring DB2 Data Links Manager for AIX".

2. As root, enter the following commands to start the DMAPP:

   ```
   stop.dfs all
   start.dfs all
   ```

Installing and Configuring DB2 Data Links DFS Client Enabler

In the section "Configuring a DFS Client Enabler", add the following information to Step 2:

```
Performing the "secval" commands will usually complete the configuration.
It may, however, be necessary to reboot the machine as well.
If problems are encountered in accessing READ PERMISSION DB files,
reboot the machine where the DB2 DFS Client Enabler has just been installed.
```

Installing and Configuring DB2 Data Links Manager for Solaris Operating Systems

The following actions must be performed after installing DB2 Data Links Manager for Solaris Operating Systems:

1. Add the following three lines to the `/etc/system` file:

   ```
   set dlfsdrv:glob_mod_pri=0x100800
   set dlfsdrv:glob_mesg_pri=0xff
   set dlfsdrv:ConfigDlfsUid=UID
   ```

   where *UID* represents the user ID of the id dlfm.

2. Reboot the machine to activate the changes.

Administrator Group Privileges in Data Links on Windows NT

On Windows NT, the user dlmadmin has the same privileges with regard to files linked using DataLinks as a root user does on UNIX for most functions. The following table compares both.

Operation	Unix (root)	Windows NT (dlmadmin)
Rename	Yes	Yes
Access file without token	Yes	Yes
Delete	Yes	No (see note below)
Update	Yes	No (see note below)

Note: The NTFS disallows these operations for a read-only file. The dlmadmin user can make these operations successful by enabling the write permission for the file.

Minimize Logging for Data Links File System Filter (DLFF) Installation

You can minimize logging for the Data Links File System Filter (DLFF) Installation by changing the `dlfs_cfg` file. The `dlfs_cfg` file is passed to strload routine to load the driver and configuration parameters. The file is located in the `/usr/lpp/db2_07_01/cfg/` directory. Through a symbolic link, the file can also be found in the `/etc` directory. The `dlfs_cfg` file has the following format:

```
d <driver-name> <vfs number> <dlfm id> <global message priority>
  <global module priority> - 0 1
```

where:

d   The d parameter specifies that the driver is to be loaded.

driver-name
> The driver-name is the full path of the driver to be loaded. For instance, the full path for DB2 Version 7 is /usr/lpp/db2_07_01/bin/dlfsdrv. The name of the driver is dlfsdrv.

vfs number
> This is the vfs entry for DLFS in /etc/vfs.

dlfm id
> This is the user id of the DataLinks Manager administrator.

global message priority
> This is a configurable parameter in the DLFS driver. It defines the list of the message categories that will be logged to the system log file.

global module priority
> This is a configurable parameter in the DLFS driver. It defines the list of driver routines, VFS operations and Vnode operations that will be logged to the system log file.

0 1
> 0 1 are the minor numbers for creating non clone nodes for this driver. The node names are created by appending the minor number to the cloned driver node name. No more than five minor numbers can be given (0-4).

A real-world example might look as follows:

```
d /usr/lpp/db2_07_01/bin/dlfsdrv 14,208,255,-1 - 0 1
```

The messages that are logged depend on the settings for the global message priority and global module priority. To minimize logging, you can change the value for the global message priority.

There are four message priority values you can use:

```
#define LOG_EMERGENCY      0x01
#define LOG_TRACING        0x02
#define LOG_ERROR          0x04
#define LOG_TROUBLESHOOT   0x08
```

Most of the messages in DLFF have LOG_TROUBLESHOOT as the message priority. Here are a few alternative configuration examples:

If you do require emergency messages and error messages, set the global message priority to 5 (1+4) in the dlfs_cfg configuration file:

```
d /usr/lpp/db2_07_01/bin/dlfsdrv 14,208,5,-1 - 0 1
```

If only error messages are required, set the global message priority to 4:

```
d /usr/lpp/db2_07_01/bin/dlfsdrv 14,208,4,-1 - 0 1
```

If you do not require logging for DLFS, then set global message priority to 0:

```
            d /usr/lpp/db2_07_01/bin/dlfsdrv 14,208,0,-1 - 0 1
```

Logging Messages after Installation

If you need to log emergency, error, and troubleshooting messages after
installation, you must modify the `dlfs_cfg` file. The `dlfs_cfg` file is located in
the `/usr/lpp/db2_07_01/cfg` directory. The global message priority must be set
to 255 (maximum priority) or to 13 (8+4+1). Setting the priority to 13 (8+4+1)
will log emergency, error, and troubleshooting information.

After setting the global message priority, unmount the DLFS filter file system
and reload the `dlfsdrv` driver to have the new priority values set at load time.
After reloading the dlfsdrv driver, the DLFS filter file system must be
re-mounted.

Note: The settings for dlfs_cfg will remain for any subsequent loading of
dlfsdrv driver until the dlfs_cfg file is changed again.

Minimizing Logging on Sun Solaris Systems

The file dlfs_cfg does not exist on Sun™ Solaris™ systems. Instead, the system
file `/etc/syslog.conf` contains the information used by the system log
daemon (syslogd) to forward a system message to the appropriate log files.
You can minimize logging for the DLFF Installation by commenting out the
entries for kern.notice and kern.debug in `/etc/syslog.conf`. You must then
stop and start syslogd to make your changes take effect.

To reactivate the logging of all the kernel notices and errors, you must
uncomment the entries for kern.notice and kern.debug in `/etc/syslog.conf`,
and then stop and start syslogd again.

DATALINK Restore

Restore of any offline backup that was taken after a database restore, with or
without rollforward, will not involve fast reconcile processing. In such cases,
all tables with DATALINK columns under file link control will be put in
datalink reconcile pending (DRP) state.

Drop Data Links Manager

You can now drop a DB2 Data Links Manager for a specified database. The
processing of some Data Links-related SQL requests, as well as utilities, such
as backup/restore, involve communicating with all DLMs configured to a
database. Previously, DB2 did not have the capability to drop a configured
DLM even though it may have not been operational. This resulted in an
additional overhead in SQL and utilities processing. Once a DLM was added,
the engine communicated with it in the processing of requests, which may
have resulted in the failure of some SQL requests (for example, drop
table/tablespace/database).

Uninstalling DLFM Components Using SMIT May Remove Additional Filesets

Before uninstalling DB2 (Versions 5, 6, or 7) from an AIX machine on which the Data Links Manager is installed, follow these steps:

1. As root, make a copy of /etc/vfs using the command:

   ```
   cp -p /etc/vfs /etc/vfs.bak
   ```

2. Uninstall DB2.

3. As root, replace /etc/vfs with the backup copy made in step 1:

   ```
   cp -p /etc/vfs.bak /etc/vfs
   ```

Before You Begin/Determine Hostname

You must determine the names of each of your DB2 servers and Data Links servers. You will need to know these hostnames to verify the installation. When connecting to a DB2 Data Links File Manager, the DB2 UDB server internally sends the following information to the DLFM:

- Database name
- Instance name
- Hostname

The DLFM then compares this information with its internal tables to determine whether the connection should be allowed. It will allow the connection only if this combination of database name, instance name, and hostname has been registered with it, using the dlfm add_db command. The hostname that is used in the dlfm add_db command must exactly match the hostname that is internally sent by the DB2 UDB server.

Use the exact hostname that is obtained as follows:

1. Enter the hostname command on your DB2 server. For example, this command might return db2server.

2. Depending on your platform, do one of the following:

   - On AIX, enter the host db2server command, where db2server is the name obtained in the previous step. This command should return output similar to the following:

     ```
     db2server.services.com is 9.11.302.341,  Aliases:  db2server
     ```

   - On Windows NT, enter the nslookup db2server command, where db2server is the name obtained in the previous step. This command should return output similar to the following:

     ```
     Server: dnsserv.services.com
     Address: 9.21.14.135
     Name: db2server.services.com
     Address: 9.21.51.178
     ```

- On Solaris Operating Environment, enter cat /etc/hosts | grep 'hostname'. This should return output similar to the following if the hostname is specified without a domain name in /etc/hosts:

  ```
  9.112.98.167 db2server loghost
  ```

  If the hostname is specified with a domain name, the command returns output similar to the following:

  ```
  9.112.98.167 db2server.services.com loghost
  ```

Use db2server.services.com for the hostname when registering a DB2 UDB database using the dlfm add_db command. The DB2 server's internal connections to the DLFM will fail if any other aliases are used in the dlfm add_db command.

A Data Links server is registered to a DB2 database using the DB2 "add datalinks manager for database database_alias using node hostname port port_number" command.

The hostname is the name of the Data Links server. Any valid alias of the Data Links server can be used in this command. DATALINK values that are references to this Data Links server must specify the hostname in the URL value; that is, the exact name that was used in the "add datalinks manager" command must be used when assigning URL values to DATALINK columns. Using a different alias will cause the SQL statement to fail.

Working with the DB2 Data Links File Manager: Cleaning up After Dropping a DB2 Data Links Manager from a DB2 Database

When a DB2 Data Links Manager is dropped from a database using the DROP DATALINKS MANAGER command, the command itself does not clean up the corresponding information on the DB2 Data Links Manager. Users can explicitly initiate unlinking of any files linked to the database and garbage collection of backup information. This can be done using the dlfm drop_dlm command. This command initiates asynchronous deletion of all information for a particular database. The DB2 Data Links Manager must be running for this command to be successful. It is extremely important that this command only be used after dropping a DB2 Data Links Manager; otherwise, important information about the DB2 Data Links Manager will be lost and cannot be recovered.

To initiate unlink processing and garbage collection of backup information for a particular database:

1. Log on to the system as the DB2 Data Links Manager Administrator.
2. Issue the following command:

```
dlfm drop_dlm database instance hostname

  where:

    database is the name of the remote DB2 UDB database;
    instance is the instance under which the database resides; and
    hostname is the host name of the DB2 UDB server
      on which the database resides.
```

3. Log off.

For a complete usage scenario that shows the context in which this command should be used, see the *Command Reference*.

A new error code has been created for this command (see "DLFM1001E (New Error Message)").

User Action for dlfm Client_conf Failure

If, on a DLFM client, dlfm client_conf fails for some reason, "stale" entries in DB2 catalogs may be the reason. The solution is to issue the following commands:

```
db2 uncatalog db <dbname>
db2 uncatalog node <node alias>
db2 terminate
```

Then try dlfm client_conf again.

DLFM1001E (New Error Message)

DLFM1001E: Error in drop_dlm processing.

Cause:

The DB2 Data Links Manager was unable to initiate unlink and garbage collection processing for the specified database. This can happen because of any of the following reasons:

- The DB2 Data Links Manager is not running.
- An invalid combination of database, instance, and hostname was specified in the command.
- There was a failure in one of the component services of the DB2 Data Links Manager.

Action:

Perform the following steps:

1. Ensure that the DB2 Data Links Manager is running. Start the DB2 Data Links Manager if it is not already running.
2. Ensure that the combination of database, instance, and hostname identifies a registered database. You can do this using the "dlfm list registered databases" command on the DB2 Data Links Manager.
3. If an error still occurs, refer to information in the db2diag.log file to see if any component services (for example, the Connection Management Service, the Transaction Management Service, and so on) have failed. Note the error code in db2diag.log, and take the appropriate actions suggested under that error code.

DLFM Setup Configuration File Option

The dlfm setup dlfm.cfg option has been removed. Any references to it in the documentation should be ignored.

Potential Problem When Restoring Files

Problem: When different versions of the same file are linked to a database at different times, the Data Links File Manager (DLFM) Retrieve daemon does not retrieve the correct version of the file from an archive when the database gets restored.

Background: When a database is restored from a backup image, the files that were linked in that backup image also get restored in the Data Links Manager file system (DLFS) from the archive. Here is how the DB2 Data Links Manager retrieve-and-restore process works.

- If the last modification time and size attributes of the current version of a file on disk are *different* from the attributes of the file to be restored from the archive, then the current file on disk is treated as a different version of the file. The current file on disk gets saved as filename.MOD, and the original version of the file from the DLFM archive gets restored. For example, if the current file name is abc, then abc gets copied to abc.MOD.
- If the last modification time and size attributes of the current file on disk are the *same* as those of the file to be restored from the archive, then the Data Links Retrieve daemon assumes that the file has not been modified, and it will not restore the version of the file from the archive.

Important: It is possible to modify a file but not have the last modification time and size attributes change. Such "hidden modifications" are done by making a change that does not affect the file size, and then resetting the last modification time attribute to that of the original file.

Example: Suppose you have a database called DBTEST, and it contains a table with a DATALINK column. You then perform the following tasks, in the order listed:

1. Create a file called `fileA` in a DLFS-mounted volume. This is the first version of the file.
2. Insert the fileA reference (a URL) into the DBTEST database.
3. Take a backup of the DBTEST database.
4. Delete the fileA reference from the DBTEST database.
5. Delete `fileA` from the DLFS-mounted volume.
6. Create another file named `fileA` in the DLFS-mounted volume. This is the second version of the file.
7. Insert the fileA reference (a URL) into the DBTEST database.
8. Restore the DBTEST database from the backup image.

The DLFM Retrieve Daemon copies the second version of `fileA` to `fileA.MOD`, then copies the first version of `fileA` from the archive onto the DLFS-mounted volume as the working version of `fileA`.

However, if both versions of `fileA` have the same last modification time and size attributes, the DLFM Retrieve Daemon does nothing, because it assumes that the files are actually the same version.

The result is that the second version of the file -- rather than the first version -- remains on the DLFS-mounted volume. You have not truly restored the file system to the same state it was at the time of the backup.

Solution: Ensure that your application does not replace a file with a newer version of that file with the same attributes (last modification time and size).

Error when Running Data Links/DFS Script dmapp_prestart on AIX

If the command

```
/usr/sbin/cfgdmepi  -a "/usr/lib/drivers/dmlfs.ext"
```

fails with a return code of 1 when you run the Data Links/DFS script dmapp_prestart, install DFS 3.1 ptfset1 to fix the cfgdmepi.

Tivoli Space Manager Integration with Data Links

DB2 Data Links Manager will now be able to take advantage of the functionality of Tivoli Space Manager. The Tivoli Space Manager Hierarchical Storage Manager (HSM) client program automatically migrates eligible files to storage to maintain specific levels of free space on local file systems. It automatically recalls migrated files when they are accessed, and permits users to migrate and recall specific files.

The prerequisite for this functionality is Tivoli Space Manager Version 4.2.

This new feature benefits customers who have file systems with large files that are required to be moved to tertiary storage periodically, in which the space of the file system needs to be managed on a regular basis. For many customers, Tivoli Space Manager currently provides the means to manage their tertiary storage. The new DB2 Data Links Manager support of Tivoli Space Manager provides greater flexibility in managing the space for DATALINK files. Rather than pre-allocating enough storage in the DB2 Data Links Manager file system for all files which may be stored there, Tivoli Space Manager allows allocations of the Data Links-managed file system to be adjusted over a period of time without the risk of inadvertently filling up the file system during normal usage.

Adding both Data Links and HSM support to a file system

When registering a file system with Hierarchical Storage Management (HSM), register it with HSM *first* and then with the DataLinks File Manager.

1. Register with HSM, using the command "**dsmmigfs add /fs**".
2. Register with DLM, using the command "**dlfmfsmd /fs**".

Data Links support for a file system is reflected in the stanza in /etc/filesystems for an HSM file system via the following entries:

```
vfs = dlfs
mount = false
options = rw,Basefs=fsm
nodename = -
```

Adding Data Links support to an existing HSM file system
Register with DLM, using the command "**dlfmfsmd /fs**".

Adding HSM support to an existing Data Links file system
1. Register with HSM, using the command "**dsmmigfs add /fs**".
2. Register with DLM, using the command "**dlfmfsmd /fs**".

Removing Data Links support from a Data Links-HSM file system
Remove Data Links support, using the command "**dlfmfsmd -j /fs**".

Removing HSM support from a Data Links-HSM file system

1. Remove HSM support, using the command "**dsmmigfs remove /fs**".
2. Remove Data Links support, "**dlfmfsmd -j /fs**".
3. Register with DLM, using the command "**dlfmfsmd /fs**".

Removing both Data Links and HSM support from a Data Links-HSM file system

1. Remove HSM support, using the command "**dsmmigfs remove /fs**".
2. Remove Data Links support, using the command "**dlfmfsmd -j /fs**".

Restrictions and Limitations

This function is currently supported on AIX only.

Selective migration (dsmmigrate) and recall of an FC (Read permission DB) linked file should be done by a root user only.
> Selective migration can be performed only by the file owner which in the case of Read Permission DB files is the DataLink Manager Administrator (dlfm). To access such files a token is required from the host database side. The only user who does not require a token is the "root" user. It will be easier for a "root" user to perform the selective migrate and recall on Read Permission DB files. The dlfm user can migrate an FC file using a valid token only the first time. The second time migration is attempted (after a recall), the operation will fail with error message "ANS1028S Internal program error. Please see your service representative." Running **dsmmigrate** on an FC file by a non-root user will not succeed. This limitation is minor as it is typically the administrators who will access the files on the fileserver.

stat and statfs system calls will show Vfs-type as fsm rather than dlfs, although dlfs is mounted over fsm.
> The above behavior is for the normal functionality of **dsmrecalld**daemons, which performs statfs on the file system to check if its Vfs-type is fsm or not.

Command "dsmls" does not show any output if a file having the minimum inode number is FC (Read permission DB) linked
> The **dsmls** command is similar to the **ls** command and lists the files being administered by TSM. No user action is required

Chapter 4. Installing and Configuring DB2 Data Links Manager for AIX

Common Installation Considerations

Migrating from DB2 File Manager Version 5.2 to DB2 Data Links Manager Version 7

The information in step 3 is incorrect. Step 3 should read as follows:

″3. As DLFM administrator, run the **/usr/lpp/db2_07_01/adm/db2dlmmg** command.

Chapter 6. Verifying the Installation on AIX

Workarounds in NFS environments

This section describes workarounds to known problems when running DB2 Data Links Manager for AIX in NFS environments that do not appear in the current documentation. These problems are NFS-specific and have nothing to do with DB2 Data Links Manager or DB2 Universal Database.

Additional NFS caching issues

Two different caches are maintained on the NFS client for AIX. The NFS client maintains a cache with attributes of recently accessed files and directories. The client also optionally supports a data cache for caching the content of files on the client.

The attribute caching process sometimes produces an unusual condition on an NFS client after a READ PERMISSION DB file is linked. Users are sometimes able to access a READ PERMISSION DB file without an access control token if these users were connected to the machine before the file was linked. Use one of these methods to reduce the likelihood of unauthorized file access:

- Use the **touch** command on the file before executing the SQL **INSERT** statement to set the link.
- Use the **touch** command on the directory containing the file.
- Use the **mount** command with one of the five attribute cache configuration parameters (`actimeo, acregmin, acregmax, acdirmin, acdirmax`) to minimize the time that cached attributes are retained after a file or a directory is modified.

You are most likely to observe unauthorized access of READ PERMISSION DB files during Data Links function testing since only one file is linked and there is little NFS activity. You are less likely to encounter this scenario in a production environment since NFS activity is heavy and the NFS attribute cache usually does not retain the attributes for all linked files.

Installation and Configuration Supplement

Chapter 5. Installing DB2 Clients on UNIX Operating Systems

HP-UX Kernel Configuration Parameters

The recommendation for setting HP-UX kernel parameters incorrectly states that **msgmbn** and **msgmax** should be set to 65535 or higher. Both parameters must be set to exactly 65535.

Chapter 12. Running Your Own Applications

Binding Database Utilities Using the Run-Time Client

The Run-Time Client cannot be used to bind the database utilities (import, export, reorg, the command line processor) and DB2 CLI bind files to each database before they can be used with that database. You must use the DB2 Administration Client or the DB2 Application Development Client instead.

You must bind these database utilities and DB2 CLI bind files to each database before they can be used with that database. In a network environment, if you are using multiple clients that run on different operating systems, or are at different versions or service levels of DB2, you must bind the utilities once for each operating system and DB2-version combination.

UNIX Client Access to DB2 Using ODBC

Chapter 12 (″Running Your Own Applications″) states that you need to update odbcinst.ini if you install an ODBC Driver Manager with your ODBC client application or ODBC SDK. This is partially incorrect. You do not need to update odbcinst.ini if you install a Merant ODBC Driver Manager product.

Chapter 24. Setting Up a Federated System to Access Multiple Data Sources

Federated Systems

A DB2 federated system is a special type of distributed database management system (DBMS). A federated system allows you to query and retrieve data located on other DBMSs. A single SQL statement can refer to multiple DBMSs or individual databases. For example, you can join data located in a DB2 Universal Database table, an Oracle table, and a Sybase view.

A DB2 federated system consists of a server with a DB2 instance, a database that will serve as the federated database, and one or more data sources. The

federated database contains catalog entries identifying data sources and their characteristics. A *data source* consists of a DBMS and data. Supported data sources include:

- Oracle
- Sybase
- Microsoft SQL Server
- Informix
- members of the DB2 Universal Database family (such as DB2 for OS/390, DB2 for AS/4000, and DB2 for Windows)

DB2 Universal Database federated servers communicate with and retrieve data from data sources using protocols, called *wrappers*. The wrapper that you use depends on the operating system on which the DB2 instance is running. *Nicknames* are used to identify the tables and views located at the data sources. Applications can connect to the federated database just like any other DB2 database, and query the data sources using nicknames as if they were tables or views in the federated database.

After a federated system is set up, the information in the data sources can be accessed as though the data is in a single local database. Users and applications send queries to the federated database, which retrieves data from the data sources.

A DB2 federated system operates under some restrictions. Distributed requests are limited to read-only operations in DB2 Version 7. In addition, you cannot execute utility operations (LOAD, REORG, REORGCHK, IMPORT, RUNSTATS, and so on) against nicknames. You can, however, use a pass-through facility to submit DDL and DML statements directly to DBMSs using the SQL dialect associated with that data source.

FixPak 8 or Later Required If Using DB2 Version 8 Data Sources

To successfully create nicknames for DB2 for UNIX and Windows Version 8 tables and views, you must apply the DB2 for UNIX and Windows Version 7.2 Fixpak 8 to your DB2 for UNIX and Windows Version 7.2 federated database. If you do not apply Fixpak 8 to your DB2 for UNIX and Windows Version 7.2 federated database, an error will occur when you access the nicknames.

Restriction

The new wrappers in Version 7.2 (such as Informix on AIX, HP, and Solaris Operating Environment; Oracle on Linux, HP, and Solaris Operating Environment; Sybase on AIX and Solaris Operating Environment; and Microsoft SQL Server on AIX and NT) are not available in this FixPak ; you must purchase DB2 Relational Connect Version 7.2.

Installing DB2 Relational Connect

This section provides instructions for installing DB2 Relational Connect on the server that you will use as your federated system server. Relational Connect is required to access Oracle, Sybase, Microsoft SQL Server, and Informix data sources. DB2 Relational Connect is not required to access members of the DB2 Universal Database family.

Before Installing DB2 Relational Connect:
- Make sure that you have either DB2 Universal Database Enterprise Edition or DB2 Universal Database Enterprise — Extended Edition installed on the federated server.

    On DB2 for UNIX servers:
    > If you intend to include DB2 family databases in your distributed requests, you must have selected the **Distributed Join for DB2** data sources option when you installed DB2 Universal Database. To verify that this option was implemented, check that the **FEDERATED** parameter is set to **YES**. You can check this setting by issuing the **GET DATABASE MANAGER CONFIGURATION** command, which displays all of the parameters and their current settings.

- Make sure that the client software for the data source is installed on your federated server.

Installing DB2 Relational Connect on Windows NT servers
1. Log on to the federated server with the user account that you created to perform the DB2 Universal Database installation.
2. Shut down any programs that are running so that the setup program can update files as required.
3. Invoke the setup program. You can either invoke the setup program automatically or manually. If the setup program fails to start automatically, or if you want to run the setup in a different language, invoke the setup program manually.
   - To automatically invoke the setup program:
     a. Insert the DB2 Relational Connect CD into the drive.
     b. The auto-run feature automatically starts the setup program. The system language is determined, and the setup program for that language is launched.
   - To manually invoke the setup program:
     a. Click **Start** and select the **Run** option.
     b. In the **Open** field, type the following command:

        x:\setup /i *language*

where:

x   Represents your CD-ROM drive.

language
         Represents the country/region code for your language (for
         example, EN for English).

  c. Click **OK**.

The installation launchpad opens.

4. Click **Install** to begin the installation process.

5. Follow the prompts in the setup program.

When the installation is complete, DB2 Relational Connect will be installed
in the directory along with you other DB2 products. For example, the
wrapper library for the Oracle NET8 client software (net8.dll) will be
installed in the c:\Program Files\SQLLIB\bin directory.

Installing DB2 Relational Connect on UNIX Servers

To install DB2 Relational Connect on your UNIX federated server, use the
db2setup utility.

Note: The screens that appear when you use the db2setup utility depend on
what you already have installed on the federated server. These steps assume
that you do not have Relational Connect installed.

1. Log in as a user with root authority.

2. Insert and mount your DB2 product CD-ROM. For information on how
to mount a CD-ROM, see *DB2 for UNIX Quick Beginnings*.

3. Change to the directory where the CD-ROM is mounted by entering the
cd */cdrom* command, where *cdrom* is the mount point of your product
CD-ROM.

4. Type the **./db2setup** command. After a few moments, the Install DB2 V7
window opens. This window lists the items that you currently have
installed, and the items that are available for you to install.

5. Navigate to the distributed join you want to install, such as **Distributed
Join for Informix Data sources**, and press the space bar to select it. An
asterisk appears next to the option when it is selected.

6. Select **OK**. The Create DB2 Services window opens.

7. Since your federated server already contains a DB2 instance, choose the
Do not create a DB2 instance option and select **OK**.

8. A warning appears if you have elected not to create an Administration
Server. Select **OK**. The DB2 Setup Utility window displays a Summary
Report of what will be installed. Since you have not installed Relational
Connect before, there should be two items listed:

   • the product signature for DB2 Relational Connect

• the distributed join for the data source you selected

9. Choose **Continue**. A window appears to indicate this is your final chance to stop the Relational Connect setup. Choose **OK** to continue with the setup. It may take a few minutes for the setup to complete.

10. The DB2 Setup Utility window displays a Status Report which indicates which components installed successfully. Choose **OK**. The DB2 Setup Utility window opens. Choose **Close** and then **OK** to exit the utility.

When the installation is complete, DB2 Relational Connect will be installed in the directory along with your other DB2 products.

• On DB2 for AIX servers, the directory is /usr/lpp/db2_07_01.
• On DB2 for Solaris Operating Environment servers, the directory is /opt/IBMdb2/V7.1.
• On DB2 for HP-UX servers, the directory is /opt/IBMdb2/V7.1.
• On DB2 for Linux servers, the directory is /usr/IBMdb2/V7.1.

Chapter 24. Setting Up a Federated System to Access Multiple Data Sources

Understanding the schema used with nicknames
The nickname parameter in a CREATE NICKNAME statement is a two-part name—the schema and the nickname. If you omit the schema when creating the nickname, the schema of the nickname will be the authid of the user creating the nickname. After a nickname is created, information about the nickname is stored in the catlaog views SYSCAT.TABLES, SYSCAT.TABOPTIONS, SYSCAT.COLUMNS, SYSCAT.COLOPTIONS, and SYSCAT.INDEXES.

Issues when restoring a federated database onto a different federated server
When you restore a federated database backup onto a different federated server, the database image does not contain the new database and node directory information it needs to access the DB2 family data sources. You must catalogue this information when you perform the restore.

Chapter 26. Accessing Oracle Data Sources

In addition to supporting wrappers on AIX and Windows NT, DB2 Universal Database now supports the Oracle wrapper on Linux, the Solaris Operating Environment, and HP-UX. This support is limited to Oracle Version 8. To access the wrappers for these platforms, you need to insert the V7.2 DB2 Relational Connect CD and select Distributed Join for Oracle data sources.

Once you have installed DB2 Relational Connect, you can add an Oracle data source to a federated server:

1. Install and configure the Oracle client software on the DB2 federated server.
2. For DB2 federated servers on UNIX, run the djxlink script to link-edit Oracle SQL*Net or Net8 libraries to your DB2 federated server and create the DB2 federated wrapper library for use with Oracle.
3. Create (or update) the db2dj.ini file to add environment variables for Oracle. This file must contain a definition for the ORACLE_HOME environment variable.
4. (Optional) Set the DB2_DJ_INI and the DB2_DJ_COMM profile registry variables.
5. Check the location and contents of the Oracle tnsnames.ora file on the DB2 federated server, and test the connections to the Oracle server using Oracle sqlplus.
6. Recycle the DB2 instance.
7. Create the wrapper.
8. Create a server definition.
9. Create a user mapping.
10. Test the configuration using Set Passthru.
11. Create nicknames for the tables and views.

Detailed instructions for these steps, including setting the environment variables, are in Chapter 26. Setting Up a Federated System to Access Oracle Data Sources in the *DB2 Installation and Configuration Supplement*. This information is also available online at http://www.ibm.com/software/data/db2/relconnect/.

Documentation Errors

The section, "Adding Oracle Data Sources to a Federated System" has the following errors:

- An additional step is needed between steps 2 and 3 in the book. The first three steps should be as follows:
  1. Install and configure the Oracle client software on the DB2 federated server using the documentation provided by Oracle.
  2. Set the ORACLE_HOME environment variable:

     export ORACLE_HOME=<oracle_home_directory>**.**
  3. For DB2 federated servers running on UNIX platforms, run the djxlink script to link-edit the Oracle SQL*Net or Net8 libraries to your DB2 federated server. Depending on your platform, the djxlink script is located in:

     /usr/lpp/db2_07_01/bin on AIX

     /opt/IBMdb2/V7.1/bin Solaris Operating Environment

     /opt/IBMdb2/V7.1/bin HP-UX

/usr/IBMdb2/V7.1/bin Linux

Run the djxlink script only after installing Oracle's client software on the DB2 federated server.

4. Set data source environment variables by modifying the db2dj.ini file and issuing the **db2set** command. The **db2set** command updates the DB2 profile registry with your settings.

   Detailed instructions for setting the environment variables are in Chapter 26. Setting Up a Federated System to Access Oracle Data Sources of the *DB2 Installation and Configuration Supplement*.

5. Continue the steps from step 3 as written in the book.

- The documentation indicates to set:

  ```
  DB2_DJ_INI = sqllib/cfg/db2dj.ini
  ```

  This is incorrect, it should be set to the following:

  ```
  DB2_DJ_INI = $INSTHOME/sqllib/cfg/db2dj.ini
  ```

Avoiding problems when working with remote LOBs

When working with remote LOB columns, you may encounter an out of memory problem. For example, suppose you run a query that selects LONG data from an Oracle column, and inserts the data into a DB2 table as a CLOB. If you have not increased the database application heap size, you will receive a SQL error indicating "not enough memory". To resolve this error:

1. Disconnect all the applications from the DB2 instance.
2. Update the application heap size using this command:

   ```
   db2 udpate db cfg for dbname using APPLHEAPSZ 1024
   ```

   where *dbname* is the name of the federated database and *1024* is the recommended heap size.
3. Re-initialize the database.

To prevent this problem from occurring, increase your database application heap size. For this change to take effect, reinitialize the database. For example:

1. Update the application heap size using this command:

   ```
   db2 udpate db cfg for dbname using APPLHEAPSZ 1024
   ```

   where *dbname* is the name of the federated database and *1024* is the recommended heap size.
2. Disconnect all the applications from the DB2 instance.
3. Re-initialize the database.

Accessing Sybase Data Sources

Before you add Sybase data sources to a federated server, you need to install and configure the Sybase Open Client software on the DB2 federated server. See the installation procedures in the documentation that comes with Sybase database software for specific details on how to install the Open Client software. As part of the installation, make sure that you include the Sybase catalog stored procedures are installed on the Sybase server and the Sybase Open Client libraries are installed on the DB2 federated server.

After configuring the connection from the client software to the Sybase server, test the connection using one of the Sybase tools. Use the isql tool for UNIX and the SQL Advantage tool for Windows.

To set up your federated server to access data stored on Sybase data sources, you need to:
1. Install DB2 Relational Connect Version 7.2. See "Installing DB2 Relational Connect" on page 57.
2. Add Sybase data sources to your federated server.
3. Specify the Sybase code pages.

This chapter discusses steps 2 and 3.

The instructions in this chapter apply to Windows NT, AIX, and the Solaris Operating Environment. The platform-specific differences are noted where they occur.

Adding Sybase Data Sources to a Federated Server

To add a Sybase data source to a federated server, you need to:
1. Set the environment variables and update the profile registry (AIX and Solaris only).
2. Link DB2 to Sybase client software (AIX and Solaris only).
3. Recycle the DB2 instance (AIX and Solaris only).
4. Create and set up an interfaces file.
5. Create the wrapper.
6. Optional: Set the DB2_DJ_COMM environment variable.
7. Create the server.
8. Optional: Set the CONNECTSTRING server option.
9. Create a user mapping.
10. Create nicknames for tables and views.

These steps are explained in detail in this section.

Step 1: Set the environment variables and update the profile registry (AIX and Solaris only)
Set data source environment variables by modifying the db2dj.ini file and issuing the **db2set** command. The db2dj.ini file contains configuration information about the Sybase client software installed on your federated server. The **db2set** command updates the DB2 profile registry with your settings.

In a partitioned database system, you can use a single db2dj.ini file for all nodes in a particular instance, or you can use a unique db2dj.ini file for one or more nodes in a particular instance. A nonpartitioned database system can have only one db2dj.ini file per instance.

To set the environment variables:
1. Edit the db2dj.ini file located in sqllib/cfg, and set the following environment variable:

   SYBASE="<*sybase home directory*>"

   where <*sybase home directory*> is the directory where the Sybase client is installed.
2. Issue the **db2set** command to update the DB2 profile registry with your changes. The syntax of this command, **db2set**, is dependent upon your database system structure. This step is only necessary if you are using the db2dj.ini file in any of the following database system structures:

   If you are using the db2dj.ini file in a nonpartitioned database system, or if you want the db2dj.ini file to apply to the current node only, issue:

   db2set DB2_DJ_INI=$HOME/sqllib/cfg/db2dj.ini

   If you are using the db2dj.ini file in a partitioned database system, and you want the values in the db2dj.ini file to apply to all nodes within this instance, issue:

   db2set -g DB2_DJ_INI=$HOME/sqllib/cfg/db2dj.ini

   If you are using the db2dj.ini file in a partitioned database system, and you want the values in the db2dj.ini file to apply to a specific node, issue:

   db2set -i *INSTANCEX 3* DB2_DJ_INI=$HOME/sqllib/cfg/*node3.ini*

   where:

   INSTANCEX
           Is the name of the instance.

   3   Is the node number as listed in the db2nodes.cfg file.

node3.ini
> Is the modified and renamed version of the db2dj.ini file.

Step 2: Link DB2 to Sybase client software (AIX and Solaris Operating Environment only)

To enable access to Sybase data sources, the DB2 federated server must be link-edited to the client libraries. The link-edit process creates a wrapper for each data source with which the federated server will communicate. When you run the djxlink script you create the wrapper library. To issue the djxlink script type:

```
djxlink
```

Step 3: Recycle the DB2 instance (AIX and Solaris Operating Environment only)

To ensure that the environment variables are set in the program, recycle the DB2 instance. When you recycle the instance, you refresh the DB2 instance to accept the changes that you made.

Issue the following commands to recycle the DB2 instance:

On DB2 for Windows NT servers:
```
NET STOP instance_name
NET START instance_name
```

On DB2 for AIX and Solaris servers:
```
db2stop
db2start
```

Step 4: Create and set up an interfaces file

To create and set up an interfaces file, you must create the file and make the file accessible.

1. Use the Sybase-supplied utility to create an interfaces file that includes the data for all the Sybase Open Servers that you want to access. See the installation documentation from Sybase for more information about using this utility.

   Windows NT typically names this file sql.ini. Rename the file you just created from sql.ini to interfaces to name the file universally across all platforms. If you choose not to rename sql.ini to interfaces you must use the IFILE parameter or the CONNECTSTRING option that is explained in step 8.

   On AIX and Solaris systems this file is named <instance home>/sqllib/interfaces.

2. Make the interfaces file accessible to DB2.

   On DB2 for Windows NT servers:
   > Put the file in the DB2 instance's %DB2PATH% directory.

On DB2 for AIX and Solaris servers:
> Put the file in the DB2 instance's $HOME/sqllib directory. Use the **ln** command to link to the file from the DB2 instance's $HOME/sqllib directory. For example:
>
> `ln -s -f /home/sybase/interfaces  /home/db2djinst1/sqllib`

Step 5: Create the wrapper

Use the CREATE WRAPPER statement to specify the wrapper that will be used to access Sybase data sources. Wrappers are mechanisms that federated servers use to communicate with and retrieve data from data sources. DB2 includes two wrappers for Sybase, CTLIB and DBLIB. The following example shows a CREATE WRAPPER statement:

`CREATE WRAPPER CTLIB`

where CTLIB is the default wrapper name used with Sybase Open Client software. The CTLIB wrapper can be used on Windows NT, AIX, and Solaris servers.

You can substitute the default wrapper name with a name that you choose. However, if you do so, you must also include the LIBRARY parameter and the name of the wrapper library for your federated server in the CREATE WRAPPER statement. See the CREATE WRAPPER statement in the *DB2 SQL Reference* for more information about wrapper library names.

Step 6: Optional: Set the DB2_DJ_COMM environment variable

To improve performance when the Sybase data source is accessed, set the DB2_DJ_COMM environment variable. This variable determines whether a wrapper is loaded when the federated server initializes. Set the DB2_DJ_COMM environment variable to include the wrapper library that corresponds to the wrapper that you specified in the previous step; for example:

On DB2 for AIX servers:
> `db2set DB2_DJ_COMM='libctlib.a'`

On DB2 for Solaris servers:
> `db2set DB2_DJ_COMM='libctlib.so'`

Ensure that there are no spaces on either side of the equal sign (=).

Refer to the *DB2 SQL Reference* for more information about wrapper library names. Refer to the *Administration Guide* for information about the DB2_DJ_COMM environment variable.

Step 7: Create the server

Use the CREATE SERVER statement to define each Sybase server whose data sources you want to access; for example:

```
CREATE SERVER SYBSERVER TYPE SYBASE VERSION 12.0 WRAPPER CTLIB
OPTIONS (NODE 'sybnode', DBNAME'sybdb')
```

where:

SYBSERVER
Is a name that you assign to the Sybase server. This name must be unique.

SYBASE
Is the type of data source to which you are configuring access. Sybase is the only data source that is supported.

12.0   Is the version of Sybase that you are accessing. The supported versions are 10.0, 11.0, 11.1, 11.5, 11.9, and 12.0.

CTLIB Is the wrapper name that you specified in the CREATE WRAPPER statement.

'sybnode'
Is the name of the node where *SYBSERVER* resides. Obtain the node value from the interfaces file. This value is case-sensitive.

Although the name of the node is specified as an option, it is required for Sybase data sources. See the *DB2 SQL Reference* for information on additional options.

'sybdb' Is the name of the Sybase database that you want to access. Obtain this name from the Sybase server.

You can use the IGNORE_UDT server option with CTLIB and DBLIB protocols to specify whether the federated server should determine the built-in type that underlies a UDT without strong typing. This server option applies only to data sources accessed through the CTLIB and DBLIB protocols. Valid values are:

'Y'   Ignore the fact that UDTs are user-defined and determine what built-in types under lie them.

'N'   Do not ignore user-defined specifications of UDTs. This is the default setting.

When DB2 creates nicknames, it looks for and catalogs information about the objects (tables, views, stored procedures) that the nicknames point to. As it looks for the information, it might find that some objects have data types that it doesn't recognize (that is, data types that don't map to counterparts at the federated database). Such unrecognizable types can include:
- New built-in types
- UDTs with strong typing

- UDTs without strong typing. These are built-in types that the user has simply renamed. These types are supported only by certain data sources, such as Sybase and Microsoft SQL Server.

When the federated server finds data types that it does not recognize, it returns the error message, SQL3324N. However, it can make an exception to this practice. For data sources accessible through the CTLIB or DBLIB protocols, you can set the IGNORE_UDT server option so that when the federated database encounters an unrecognizable UDT without strong typing, the federated database determines what the UDT's underlying built-in type is. Then, if the federated database recognizes this built-in type, the federated database returns information about the built-in type to the catalog. To have the federated database determine the underlying built-in types of UDTs that do not have strong typing, set IGNORE_UDT to 'Y'.

Step 8: Optional: Set the CONNECTSTRING server option

Specify the timeout thresholds, the path and name of the interfaces file, and the packet size of the interfaces file. Sybase Open Client uses timeout thresholds to interrupt queries and responses that run for too long a period of time. You can set these thresholds in DB2 by using the CONNECTSTRING option of the CREATE SERVER OPTION DDL statement. Use the CONNECTSTRING option to specify:

- Timeout duration for SQL queries.
- Timeout duration for login response.
- Path and name of the interfaces file.
- Packet size.
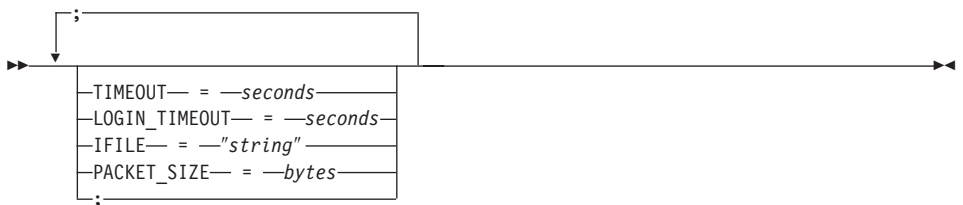
TIMEOUT

Specifies the number of seconds for DB2 Universal Database to wait for a response from Sybase Open Client for any SQL statement. The value of *seconds* is a positive whole number in DB2 Universal Database's integer range. The timeout value that you specify depends on which wrapper you are using. Windows NT, AIX, and Solaris servers are all able to utilize the DBLIB wrapper. The default value for the DBLIB wrapper is 0. On Windows NT, AIX, and Solaris servers the default value for DBLIB causes DB2 Universal Database to wait indefinitely for a response.

LOGIN_TIMEOUT

Specifies the number of seconds for DB2 Universal Database to wait for a response from Sybase Open Client to the login request. The default values are the same as for TIMEOUT.

IFILE

Specifies the path and name of the Sybase Open Client interfaces file. The path that is identified in *string* must be enclosed in double quotation marks ("). On Windows NT servers, the default is %DB2PATH%. On AIX and Solaris servers, the default value is `sqllib/interfaces` in the home directory of your DB2 Universal Database instance.

PACKET_SIZE

Specifies the packet size of the interfaces file in bytes. If the data source does not support the specified packet size, the connection will fail. Increasing the packet size when each record is very large (for example, when inserting rows into large tables) significantly increases performance. The byte size is a numeric value. See the Sybase reference manuals for more information.

Examples:

On Windows NT servers, to set the timeout value to 60 seconds and the interfaces file to `C:\etc\interfaces`, use:

```
CREATE SERVER OPTION connectstring FOR SERVER sybase1
SETTING 'TIMEOUT=60;LOGIN_TIMEOUT=5;IFILE="C:\etc\interfaces"'
```

On AIX and Solaris servers, set the timeout value to 60 seconds and the interfaces file to`/etc/interfaces`, use:

```
CREATE SERVER OPTION connectstring FOR SERVER sybase1
SETTING 'TIMEOUT=60;PACKET_SIZE=4096;IFILE="/etc/interfaces"'
```

Step 9: Create a user mapping

If a user ID or password on the federated server is different from a user ID or password on a Sybase data source, use the CREATE USER MAPPING statement to map the local user ID to the user ID and password defined at the Sybase data source; for example:

```
CREATE USER MAPPING FOR DB2USER SERVER SYBSERVER
OPTIONS ( REMOTE_AUTHID 'sybuser', REMOTE_PASSWORD 'day2night')
```

where:

DB2USER

Is the local user ID that you are mapping to a user ID defined at an Sybase data source.

SYBSERVER
> Is the name of the Sybase data source that you defined in the CREATE SERVER statement.

'sybuser'
> Is the user ID at the Sybase data source to which you are mapping *DB2USER*. This value is case sensitive.

'day2night'
> Is the password associated with *'sybuser'*. This value is case sensitive.

See the *DB2 SQL Reference* for more information on additional options.

Step 10: Create nicknames for tables and views

Assign a nickname for each view or table located at your Sybase data source. You will use these nicknames when you query the Sybase data source. Sybase nicknames are case sensitive. Enclose both the schema and table names in double quotation marks ("). The following example shows a CREATE NICKNAME statement:

```
CREATE NICKNAME SYBSALES FOR SYBSERVER."salesdata"."europe"
```

where:

SYBSALES
> Is a unique nickname for the Sybase table or view.

SYBSERVER."salesdata"."europe"
> Is a three-part identifier that follows this format:
>
> *data_source_name."remote_schema_name"."remote_table_name"*

Repeat this step for each table or view to which you want create nicknames. When you create the nickname, DB2 will use the connection to query the data source catalog. This query tests your connection to the data source. If the connection does not work, you receive an error message.

See the *DB2 SQL Reference* for more information about the CREATE NICKNAME statement. For more information about nicknames in general and to verify data type mappings, see the *DB2 Administration Guide*.

Specifying Sybase code pages

This step is necessary only when the DB2 federated server and the Sybase server are running different code pages. Data sources that are using the same code set as DB2 require no translation. The following table provides equivalent Sybase options for common National Language Support (NLS) code pages. Either your Sybase data sources must be configured to correspond to these equivalents, or the client code must be able to detect the mismatch and flag it as an error or map the data by using its own semantics. If no conversion table can be found from the source code page to the target code

page, DB2 issues an error message. Refer to your Sybase documentation for more information.

Table 2. Sybase Code Page Options

Code page	Equivalent Sybase option
850	cp850
897	sjis
819	iso_1
912	iso_2
1089	iso_6
813	iso_7
916	iso_8
920	iso_9

Accessing Microsoft SQL Server Data Sources using ODBC (new chapter)

Before you add Microsoft SQL Server data sources to a DB2 federated server, you need to install and configure the ODBC driver on the federated server. See the installation procedures in the documentation that comes with the ODBC driver for specific details on how install the ODBC driver.

To set up your federated server to access data stored in Microsoft SQL Server data sources, you need to:

1. Install and configure the ODBC driver on the federated server. See the installation procedures in the documentation that comes with the ODBC driver for specific details on how to install the ODBC driver.

   On DB2 for Windows NT servers:
   Configure a system DSN using the ODBC device manager. In the Windows ODBC Data Source Administrator, specify the SQL Server driver and proceed through the dialog to add a new System DSN. Specify "SQL Server Authentication using Login ID and password provided by the user."

   On DB2 for AIX servers:
   Install the threaded version of the libraries supplied by MERANT, specify the MERANT library directory as the first entry in the LIBPATH, and set up the .odbc.ini file.

2. Install DB2 Relational Connect Version 7.2. See "Installing DB2 Relational Connect" on page 57.

3. Add Microsoft SQL Server data sources to your federated server.

4. Specify the Microsoft SQL Server code pages. (Windows NT only)

This chapter discusses steps 3 and 4.

The instructions in this chapter apply to Windows NT and AIX platforms. The platform-specific differences are noted where they occur.

Adding Microsoft SQL Server Data Sources to a Federated Server

After you install the ODBC driver and DB2 Relational Connect, add Microsoft SQL Server data sources to your federated server using these steps:

1. Set the environment variables (AIX only).
2. Run the shell script (AIX only).
3. Optional: Set the DB2_DJ_COMM environment variable. (AIX only)
4. Recycle the DB2 instance (AIX only).
5. Create the wrapper.
6. Create the server.
7. Create a user mapping.
8. Create nicknames for the tables and views.
9. Optional: Obtain the ODBC traces.

These steps are explained in detail in the following sections.

Step 1: Set the environment variables (AIX only)

Set data source environment variables by modifying the db2dj.ini file and issuing the **db2set** command. The db2dj.ini file contains configuration information to connect to Microsoft SQL Server data sources. The **db2set** command updates the DB2 profile registry with your settings.

In a partitioned database system, you can use a single db2dj.ini file for all nodes in a particular instance, or you can use a unique db2dj.ini file for one or more nodes in a particular instance. A nonpartitioned database system can have only one db2dj.ini file per instance.

To set the environment variables:

1. Edit the db2dj.ini file located in $HOME/sqllib/cfg/, and set the following environment variables:

```
ODBCINI=$HOME/.odbc.ini
DJX_ODBC_LIBRARY_PATH=<path to the Merant driver>/lib
DB2ENVLIST=LIBPATH
```

Issue the **db2set** command to update the DB2 profile registry with your changes. The syntax of **db2set** is dependent upon your database system structure:

- If you are using the db2dj.ini file in a nonpartitioned database system, or if you are using the db2dj.ini file in a partitioned database system and you want the values in the db2dj.ini file to apply to the current node only, issue this command:

  ```
  db2set DB2_DJ_INI=<path to ini file>/db2dj.ini
  ```

- If you are using the db2dj.ini file in a partitioned database system and you want the values in the db2dj.ini file to apply to all nodes within this instance, issue this command:

  ```
  db2set -g DB2_DJ_INI=<path to ini file>/db2dj.ini
  ```

- If you are using the db2dj.ini file in a partitioned database system, and you want the values in the db2dj.ini file to apply to a specific node, issue this command:

  ```
  db2set -i INSTANCEX  3 DB2_DJ_INI=$HOME/sqllib/cfg/node3.ini
  ```

where:

INSTANCEX
      Is the name of the instance.

3   Is the node number as listed in the db2nodes.cfg file.

node3.ini
      Is the modified and renamed version of the db2dj.ini file.

To set the path to the client library, issue these commands:

```
db2set DB2LIBPATH=<path to the Merant client library>
db2set DB2ENVLIST=LIBPATH
```

Step 2: Run the shell script (AIX only)
The djxlink.sh shell script links the client libraries to the wrapper libraries. To run the shell script:

```
djxlink
```

Step 3: Optional: Set the DB2_DJ_COMM environment variable (AIX only)
If you find it takes an inordinate amount of time to access the Microsoft SQL Server data source, you can improve the performance by setting the DB2_DJ_COMM environment variable to load the wrapper when the federated server initializes rather than when you attempt to access the data source. Set the DB2_DJ_COMM environment variable to include the wrapper library that corresponds to the wrapper that you specified in Step 5. For example:

On DB2 for Windows NT servers:

```
db2set DB2_DJ_COMM=djxmssql3.dll
```

On DB2 for AIX servers:

```
db2set DB2_DJ_COMM=libmssql3.a
```

Ensure that there are no spaces on either side of the equal sign (=).

See the *DB2 SQL Reference* for more information about wrapper library names.

Step 4: Recycle the DB2 instance (AIX only)

To ensure that the environment variables are set in the program, recycle the DB2 instance. When you recycle the instance, you refresh the DB2 instance to accept the changes that you made. Recycle the DB2 instance by issuing the following commands:

```
db2stop
db2start
```

Step 5: Create the wrapper

DB2 Universal Database has two different protocols, called wrappers, that you can use to access Microsoft SQL Server data sources. Wrappers are the mechanism that federated servers use to communicate with and retrieve data from data sources. The wrapper that you use depends on the platform on which DB2 Universal Database is running. Use Table 3 as a guide to selecting the appropriate wrapper.

Table 3. ODBC drivers

ODBC driver	Platform	Wrapper Name
ODBC 3.0 (or higher) driver	Windows NT	DJXMSSQL3
MERANT DataDirect Connect ODBC 3.6 driver	AIX	MSSQLODBC3

Use the CREATE WRAPPER statement to specify the wrapper that will be used to access Microsoft SQL Server data sources. The following example shows a CREATE WRAPPER statement:

```
CREATE WRAPPER DJXMSSQL3
```

where DJXMSSQL3 is the default wrapper name used on a DB2 for Windows NT server (using the ODBC 3.0 driver). If you have a DB2 for AIX server, you would specify the MSSQLODBC3 wrapper name.

You can substitute the default wrapper name with a name that you choose. However, if you do so, you must include the LIBRARY parameter and the name of the wrapper library for your federated server platform in the CREATE WRAPPER statement. For example:

On DB2 for Windows NT servers:

```
CREATE WRAPPER wrapper_name LIBRARY 'djxmssql3.dll'
```

where *wrapper_name* is the name that you want to give the wrapper, and 'djxmssql3.dll' is the library name.

On DB2 for AIX servers:

```
CREATE WRAPPER wrapper_name LIBRARY 'libmssql3.a'
```

where *wrapper_name* is the name that you want to give the wrapper, and 'libdjxmssql.a' is the library name.

See the CREATE WRAPPER statement in the *DB2 SQL Reference* for more information about wrapper library names.

Step 6: Create the server

Use the CREATE SERVER statement to define each Microsoft SQL Server data source to which you want to connect. For example:

```
CREATE SERVER sqlserver TYPE MSSQLSERVER VERSION 7.0 WRAPPER djxmssql3
OPTIONS (NODE 'sqlnode', DBNAME 'database_name')
```

where:

sqlserver

Is a name that you assign to the Microsoft SQL Server server. This name must be unique.

MSSQLSERVER

Is the type of data source to which you are configuring access.

7.0   Is the version of Microsoft SQL Server that you are accessing. DB2 Universal Database supports versions 6.5 and 7.0 of Microsoft SQL Server.

DJXMSSQL3

Is the wrapper name that you defined in the CREATE WRAPPER statement.

'sqlnode'

Is the system DSN name that references the Microsoft SQL Server version of Microsoft SQL Server that you are accessing. This value is case sensitive. DB2 Universal Database supports versions 6.5 and 7.0 of Microsoft SQL Server.

Although the name of the node (System DSN name) is specified as an option in the CREATE SERVER statement, it is required for Microsoft SQL Server data sources. On Windows, obtain the DSN from the System DSN tab of the Windows ODBC Data Source Administrator tool. On AIX, obtain the DSN from the .odbc.ini file in the DB2 instance owners home directory.

See the *DB2 SQL Reference* for additional options that you can use with the CREATE WRAPPER statement.

'database_name'

Is the name of the database to which you are connecting.

Although the name of the database is specified as an option in the CREATE SERVER statement, it is required for Microsoft SQL Server data sources.

Step 7: Create a user mapping

If a user ID or password at the federated server is different from a user ID or password at a Microsoft SQL Server data source, use the CREATE USER MAPPING statement to map the local user ID to the user ID and password defined at the Microsoft SQL Server data source; for example:

```
CREATE USER MAPPING FOR db2user SERVER server_name
OPTIONS (REMOTE_AUTHID 'mssqluser', REMOTE_PASSWORD 'day2night')
```

where:

db2user
    Is the local user ID that you are mapping to a user ID defined at the Microsoft SQL Server data source.

server_name
    Is the name of the server that you defined in the CREATE SERVER statement.

'mssqluser'
    Is the login ID at the Microsoft SQL Server data source to which you are mapping *db2user*. This value is case sensitive.

'day2night'
    Is the password associated with *'mssqluser'*. This value is case sensitive.

See the *DB2 SQL Reference* for additional options that you can use with the CREATE USER MAPPING statement.

Step 8: Create nicknames for tables and views

Assign a nickname for each view or table located in your Microsoft SQL Server data source that you want to access. You will use these nicknames when you query the Microsoft SQL Server data source. Use the CREATE NICKNAME statement to assign a nickname. Nicknames are case sensitive. The following example shows a CREATE NICKNAME statement:

```
CREATE NICKNAME mssqlsales FOR server_name.salesdata.europe
```

where:

mssqlsales
    Is a unique nickname for the Microsoft SQL Server table or view.

server_name.salesdata.europe
    Is a three-part identifier that follows this format:

    data_source_server_name.remote_schema_name.remote_table_name

Double quotes are recommended for the *remote_schema_name* and *remote_table_name* portions of the nickname.

When you create a nickname, DB2 attempts to access the data source catalog tables (Microsoft SQL Server refers to these as system tables). This tests the connection to the data source. If the connection fails, you receive an error message.

Repeat this step for all database tables and views for which you want to create nicknames.

For more information about the CREATE NICKNAME statement, see the *DB2 SQL Reference*. For more information about nicknames in general, and to verify data type mappings see the *DB2 Administration Guide*.

Step 9: Optional: Obtain ODBC traces

If you are experiencing problems when accessing the data source, you can obtain ODBC tracing information to analyze and resolve these problems. To ensure the ODBC tracing works properly, use the trace tool provided by the ODBC Data Source Administrator. Activating tracing impacts your system performance, therefore you should turn off tracing once you have resolved the problems.

Reviewing Microsoft SQL Server code pages (Windows NT only)

Microsoft SQL Server supports many of the common National Language Support (NLS) code page options that DB2 UDB supports. Data sources that are using the same code set as DB2 require no translation. Table 3 lists the code pages that are supported by both DB2 Universal Database and Microsoft SQL Server.

Table 4. DB2 UDB and Microsoft SQL Server Code Page Options

Code page	Language supported
1252	ISO character set
850	Multilingual
437	U.S. English
874	Thai
932	Japanese
936	Chinese (simplified)
949	Korean
950	Chinese (traditional)
1250	Central European
1251	Cyrillic

Table 4. DB2 UDB and Microsoft SQL Server Code Page Options (continued)

Code page	Language supported
1253	Greek
1254	Turkish
1255	Hebrew
1256	Arabic

When the DB2 federated server and the Microsoft SQL Server are running different National Language Support (NLS) code pages either your Microsoft SQL Server data sources must be configured to correspond to these equivalents, or the client code must be able to detect the mismatch and flag it as an error or map the data by using its own semantics. If no conversion table can be found from the source code page to the target code page, DB2 issues an error message. Refer to your Microsoft SQL Server documentation for more information.

Accessing Informix Data Sources (new chapter)

Before you add Informix data sources to a DB2 federated server, you need to install and configure the Informix Client SDK software on the federated server. See the installation procedures in the documentation that comes with Informix database software for specific details on how to install the Client SDK software. As part of the installation, make sure that you include the Informix Client SDK libraries.

To set up your federated server to access data stored on Informix data sources, you need to:
1. Install DB2 Relational Connect. See "Installing DB2 Relational Connect" on page 57.
2. Apply the latest DB2 FixPak.
3. Add Informix data sources to your federated server.

This chapter discusses step 3.

The instructions in this chapter apply to AIX, Solaris Operating Environment, and HP-UX operating systems. Specific operating system differences are noted where they occur.

Adding Informix Data Sources to a Federated Server

To add a Informix data source to a federated server, you need to:
1. Set the environment variables and update the profile registry.
2. Link DB2 to the Informix client software.

3. Recycle the DB2 instance.
4. Create the Informix `sqlhosts` file.
5. Create the wrapper.
6. Optional: Set the DB2_DJ_COMM environment variable.
7. Create a server.
8. Create a user mapping.
9. Create nicknames for tables, views and Informix synonyms.

These steps are explained in detail in this section.

Step 1: Set the environment variables and update the profile registry
Set data source environment variables by modifying the db2dj.ini file and
issuing the **db2set** command. The db2dj.ini file contains configuration
information about the Informix client software installed on your federated
server. The **db2set** command updates the DB2 profile registry with your
settings.

In a partitioned database system, you can use a single db2dj.ini file for all
nodes in a particular instance, or you can use a unique db2dj.ini file for one
or more nodes in a particular instance. A nonpartitioned database system can
have only one db2dj.ini file per instance.

To set the environment variables:
1. Edit the db2dj.ini file located in the sqllib/cfg directory, and set the
   following environment variables:

   Note: You can create this file yourself if it is not already on the system.

   INFORMIXDIR

   Set the INFORMIXDIR environment variable to the path for the
   directory where the Informix Client SDK software is installed; for
   example:
   `INFORMIXDIR=/informix/csdk`

   INFORMIXSERVER

   This variable identifies the name of the default Informix server.
   `INFORMIXSERVER=inf93`

   Note: Although the Informix wrapper does not use the value of
   this variable, the Informix client requires that this variable be set.
   The wrapper uses the value of the **node** server option, which
   specifies the Informix database server that you want to access.

   INFORMIXSQLHOSTS

If you are using the default path for the Informix `sqlhosts` file
($INFORMIXDIR/etc/sqlhosts), you do not need to set this
variable. However, if you are using a path for the Informix
`sqlhosts` file other than the default, then you need to set this
variable to the full path name of the Informix `sqlhosts` file. For
example:

```
INFORMIXSQLHOSTS=/informix/csdk/etc/my_sqlhosts
```

2. Update the `.profile` file of the DB2 instance with the Informix
   environment variables. You can do this by issuing the following
   commands to set and export each variable:

```
PATH=$INFORMIXDIR/bin:$PATH
export PATH

INFORMIXDIR=<informix_client_path>
export INFORMIXDIR
```

where `informix_client_path` is the path on the federated server for the
directory where the Informix client is installed. Use double quotes (")
around the path if a name in the path contains a blank.

3. Execute the DB2 instance `.profile` by entering:

```
. .profile
```

4. Issue the **db2set** command to update the DB2 profile registry with your
   changes. The syntax of this command, **db2set**, is dependent upon your
   database system structure. This step is only necessary if you are using the
   db2dj.ini file in any of the following database system structures:

   If you are using the db2dj.ini file in a nonpartitioned database system, or
   if you want the db2dj.ini file to apply to the current node only, issue:

```
db2set DB2_DJ_INI=<path to sqllib>/sqllib/cfg/db2dj.ini
```

   Note: The pathnames in this section should be fully qualified. For
   example, `my_home/my_instance/sqllib/cfg/db2dj.ini`

   If you are using the db2dj.ini file in a partitioned database system, and
   you want the values in the db2dj.ini file to apply to all nodes within this
   instance, issue:

```
db2set -g DB2_DJ_INI=<path to sqllib>/sqllib/cfg/db2dj.ini
```

   If you are using the db2dj.ini file in a partitioned database system, and
   you want the values in the db2dj.ini file to apply to a specific node,
   issue:

```
db2set -i INSTANCEX 3 DB2_DJ_INI=sqllib/cfg/node3.ini
```

   where:

INSTANCEX
> Is the name of the instance.

3 Is the node number as listed in the db2nodes.cfg file.

node3.ini
> Is the modified and renamed version of the db2dj.ini file.

Step 2: Link DB2 to Informix client software

To enable access to Informix data sources, the DB2 federated server must be link-edited to the client libraries. The link-edit process creates a wrapper library for each data source with which the federated server will communicate. When you run the djxlinkInformix script you create the Informix wrapper library. To issue the djxlinkInformix script, type:

djxlinkInformix

Note:

> The djxlinkInformix script only creates the Informix wrapper library. There is another script, the djxlink script that attempts to create a wrapper library for every data source that DB2 Universal Database supports (Oracle, Microsoft SQL Server, etc.). If you only have the client software for some of the data sources installed, you will receive an error message for each of the missing data sources when you issue the djxlink script.

> You need UNIX Systems Administrator (root) authorization to run the djxlinkInformix and djxlink scripts.

> The djxlinkInformix and djxlink scripts write detailed error and warning messages to a specific file, depending on the operating system. For example, on AIX, the djxlinkInformix script writes to /usr/lpp/db2_07_01/lib/djxlinkInformix.out and the djxlink script writes to /usr/lpp/db2_07_01/lib/djxlink.out.

> The djxlinkInformix and djxlink scripts create the wrapper library in a specific directory, depending on the operating system. For example, on AIX, the libinformix.a wrapper library is created in the /usr/lpp/db2_07_01/lib directory.

> Check the permissions on the libinformix.a wrapper library after it is created to make sure that it can be read and executed by DB2 instance owners. If the DB2 instance owners are not in the System group, then permissions on the libinformix.a wrapper library will need to be -rwxr-xr-x root system...libinformix.a.

Step 3: Recycle the DB2 instance

To ensure that the environment variables are set in the program, recycle the DB2 instance. When you recycle the instance, you refresh the DB2 instance to accept the changes that you made.

Issue the following commands to recycle the DB2 instance:

On DB2 for AIX, Solaris Operating Environment, and HP-UX servers:

```
db2stop
db2start
```

Step 4: Create the Informix sqlhosts file

The file specifies the location of each Informix database server and the type of connection (protocol) for the database server. There are several ways to create this file. You can copy it from another system that has Informix Connect or Informix Client SDK connected to an Informix server. You can also configure the Informix Client SDK on the DB2 server to connect to an Informix server, which creates the sqlhosts file.

After the sqlhosts file is copied or created, the DB2 instance owner should use Informix dbaccess (if it is on the DB2 server) to connect to and query the Informix server. This will establish that the Informix Client SDK is able to connect to the Informix server before you try to configure DB2 Relational Connect to work with the Informix Client SDK.

For more information on setting up this file, refer to the Informix manual *Administrators Guide for Informix Dynamic Server*.

> **Warning:**
> If you do not define the Informix database server name in the `sqlhosts` file, then when you perform an operation that requires connecting to the Informix database server, you will receive an error.

Step 5: Create the wrapper

Use the CREATE WRAPPER statement to specify the wrapper that will be used to access Informix data sources. Wrappers are mechanisms that federated servers use to communicate with and retrieve data from data sources. The following example shows a CREATE WRAPPER statement:

```
CREATE WRAPPER informix
```

where **informix** is the wrapper_name; **informix** is the default wrapper name used with Informix Client SDK software.

You can substitute the default wrapper name with a name that you choose. However, if you do so, you must also include the LIBRARY parameter and

the name of the wrapper library for your federated server in the CREATE
WRAPPER statement. See the CREATE WRAPPER statement in the *DB2 SOL
Reference* for more information about wrapper library names.

The wrapper library names for Informix are:
- libinformix.a (AIX)
- libinformix.so (Solaris Operating Environment)
- libinformix.sl (HP-UX)

Step 6: Optional: Set the DB2_DJ_COMM environment variable

To improve performance when the Informix data source is accessed, set the
DB2_DJ_COMM environment variable on the Federated server. This variable
determines whether a wrapper is loaded when the Federated server initializes.
Set the DB2_DJ_COMM environment variable to include the wrapper library
that corresponds to the wrapper that you specified in the previous step. If you
are using the Korn shell or Bourne shell command line interfaces, use these
export commands:

On DB2 for AIX servers:

```
DB2_DJ_COMM='libinformix.a'
export DB2_DJ_COMM
```

On DB2 for Solaris Operating Environment servers:

```
DB2_DJ_COMM='libinformix.so'
export DB2_DJ_COMM
```

On DB2 for HP-UX servers:

```
DB2_DJ_COMM='libinformix.sl'
export DB2_DJ_COMM
```

Ensure that there are no spaces on either side of the equal sign (=).

If you are using the C shell command line interface, set the environment
variables using these commands:

```
setenv DB2_DJ_COMM 'libinformix.a' (DB2 for AIX servers)
setenv DB2_DJ_COMM 'libinformix.so' (DB2 for Solaris Operating Environment servers)
setenv DB2_DJ_COMM 'libinformix.sl' (DB2 for HP–UX servers)
```

Refer to the *DB2 SQL Reference* for more information about wrapper library
names and the DB2_DJ_COMM environment variable.

Step 7: Create the server

Use the CREATE SERVER statement to define each Informix server whose
data sources you want to access. The syntax for this statement is:

```
CREATE SERVER server_name TYPE server_type VERSION server_version
 WRAPPER wrapper_name
OPTIONS (NODE 'node_name', DBNAME 'database_name')
```

where:

server_name
Is a name you assign to the Informix database server. This name must be unique and not duplicate any other server_name defined in the federated database. The server_name must not be the same as the name of any table space in the federated database.

TYPE server_type
Specifies the type of data source to which you are configuring access.

> **Note:** For the Informix wrapper, the server_type must be informix.

VERSION server_version
Is the version of Informix database server that you want to access. The supported Informix versions are 5, 7, 8, and 9.

WRAPPER wrapper_name
Is the name you specified in the CREATE WRAPPER statement.

NODE 'node_name'
Is the name of the node where the **server_name** resides. The **node_name** must be defined in the Informix `sqlhosts` file (see step 4). Although the **node_name** is specified as an option in the CREATE SERVER SQL statement, it is required for Informix data sources. This value is case-sensitive. See the *DB2 SQL Reference* for information on additional options.

DBNAME 'database_name'
Is the name of the Informix database that you want to access.

The following is an example of the CREATE SERVER statement:
```
CREATE SERVER asia TYPE informix VERSION 9 WRAPPER informix
OPTIONS (NODE 'abc', DBNAME 'sales')
```

The FOLD_ID and FOLD_PW server options affect whether the wrapper folds the user ID and password to uppercase or lowercase before sending them to Informix. An example of the CREATE SERVER statement with the FOLD_ID and FOLD_PW server options is:
```
CREATE SERVER asia TYPE informix VERSION 9 WRAPPER informix
OPTIONS (NODE 'abc', DBNAME 'sales', FOLD_ID 'U', FOLD_PW 'U')
```

Step 8: Create a user mapping
If a user ID or password on the DB2 Federated server is different from a user ID or password on an Informix data source, use the CREATE USER MAPPING statement to map the local user ID to the user ID and password defined at the Informix data source; for example:
```
CREATE USER MAPPING FOR local_userid SERVER server_name
OPTIONS (REMOTE_AUTHID 'remode_userid', REMOTE_PASSWORD 'remote_password')
```

where:

local_userid
> Is the local user ID that you are mapping to a user ID defined at an Informix data source.

SERVER server_name
> Is the name of the Informix data source that you defined in the CREATE SERVER statement.

REMOTE_AUTHID 'remote_userid'
> Is the user ID at the Informix database server to which you are mapping **local_userid**. This value is case sensitive unless you set the FOLD_ID server option to 'U' or 'L' in the CREATE SERVER statement.

REMOTE_PASSWORD 'remote_password'
> Is the password associated with **remote_userid**. This value is case sensitive unless you set the FOLD_PW server option to 'U' or 'L' in the CREATE SERVER statement.

The following is an example of the CREATE USER MAPPING statement:

```
CREATE USER MAPPING FOR robert SERVER asia
OPTIONS (REMOTE_AUTHID 'bob', REMOTE_PASSWORD 'day2night')
```

You can use the DB2 special register **USER** to map the authorization ID of the person issuing the CREATE USER MAPPING statement to the data source authorization ID specified in the **REMOTE_AUTHID** user option. The following is an example of the CREATE USER MAPPING statement which includes the **USER** special register:

```
CREATE USER MAPPING FOR USER SERVER asia
OPTIONS (REMOTE_AUTHID 'bob', REMOTE_PASSWORD 'day2night')
```

See the *DB2 SQL Reference* for more information on additional options.

Step 9: Create nicknames for tables, views, and Informix synonyms
Assign a nickname for each table, view, or Informix synonym located at your Informix data source. Nicknames can be 128 characters in length. You will use these nicknames when you query the Informix data source. DB2 will fold the server, schema, and table names to uppercase unless you enclose them in double quotation marks ("). The following example shows a CREATE NICKNAME statement:

```
CREATE NICKNAME nickname FOR
   server_name."remote_schema_name"."remote_table_name"
```

where:

nickname
> Is a unique nickname used to identify the Informix table, view, or synonym.

server_name."**remote_schema_name**"."**remote_table_name**"
> Is a three-part identifier for the remote object.
> - **server_name** is the name you assigned to the Informix database server in the CREATE SERVER statement.
> - **remote_schema_name** is the name of the remote schema to which the table, view, or synonym belongs.
> - **remote_table_name** is the name of the remote table, view, or synonym which you want to access.

The following is an example of the CREATE NICKNAME statement:

```
CREATE NICKNAME salesjapan FOR asia."salesdata"."japan"
```

Repeat this step for each table or view to which you want create a nickname. When you create the nickname, DB2 will use the connection to query the data source catalog. This query tests your connection to the data source. If the connection does not work, you receive an error message.

See the *DB2 SQL Reference* for more information about the CREATE NICKNAME statement. For more information about nicknames in general and to verify data type mappings, see the *DB2 Administration Guide*.

Part 3. Administration

Administration Guide

Update Available

The Administration Guide was updated as part of FixPak 4. The latest PDF is available for download online at http://www.ibm.com/software/data/db2/udb/winos2unix/support. The information in these notes is in addition to the updated reference. All updated documentation is also available on CD. This CD can be ordered through DB2 service using the PTF number U478862. Information on contacting DB2 Service is available at http://www.ibm.com/cgi-bin/db2www/data/db2/udb/winos2unix/support/help.d2w/report.

Administration Guide: Planning

Chapter 8. Physical Database Design

Table Space Design Considerations

Optimizing Table Space Performance when Data is Place on Raid

DB2_PARALLEL_IO: DB2_PARALLEL_IO also affects table spaces with more than one container defined. If you do not set the registry variable, the I/O parallelism is equal to the number of containers in the table space. If you set the registry variable, the I/O parallelism is equal to the result of prefetch size divided by extent size. You might want to set the registry variable if the individual containers in the table space are striped across multiple physical disks.

For example, a table space has two containers and the prefetch size is four times the extent size. If the registry variable is not set, a prefetch request for this table space will be broken into two requests (each request will be for two extents). Provided that the prefetchers are available to do work, two prefetchers can be working on these requests in parallel. In the case where the registry variable is set, a prefetch request for this table space will be broken into four requests (one extent per request) with a possibility of four prefetchers servicing the requests in parallel.

In this example, if each of the two containers had a single disk dedicated to it, setting the registry variable for this table space might result in contention on those disks since two prefetchers will be accessing each of the two disks at once. However, if each of the two containers was striped across multiple disks, setting the registry variable would potentially allow access to four different disks at once.

Partitioning Keys

In the "Nodegroup Design Considerations" subsection of the "Designing Nodegroups" section , the following text from the "Partitioning Keys" sub-subsection stating the points to be considered when defining partitioning keys should be deleted only if DB2_UPDATE_PART_KEY=ON:

Note: If DB2_UPDATE_PART_KEY=OFF, then the restrictions still apply.

Note: In FixPak 3 and later, the default value is OFF.

- You cannot update the partitioning key column value for a row in the table.
- You can only delete or insert partitioning key column values.

Appendix D. Incompatibilities Between Releases

Error SQL30081N Not Returned When Lost Connection Is Detected

Applications that detect a lost connection to the database server by checking for error SQL30081N will no longer detect lost connections upon migration to DB2 Universal Database Version 6 or above.

Export Utility Requires FixPak 7 or Later to Properly Handle Identity Attributes

In order for the export utility to support all the identity attributes (for example, minvalue, maxvalue, cycle, order, remarks), you must run both your client and server at a minimum level of FixPak 7. If either the client or server is not at at least this level, the export utility will still function, but will be unable to interpret the attributes.

Appendix E. National Language Support (NLS)

Country/Region Code and Code Page Support

In the table of Supported Languages and Code Sets, code page 5488 is also known as GB 18030, and code page 1394 is also known as ShiftJIS X0213.

Connection of a UTF-8 (code page 1208) client to a non-Unicode database is not supported.

Import/Export/Load Considerations — Restrictions for Code Pages 1394 and 5488

Data in code pages 1394 (ShiftJIS X0213) and 5488 (GB 18030) can be moved into a Unicode database using the load or import utilities. The export utility can be used to move data from a Unicode database to a data file in code pages 1394 and 5488.

Only connections between a Unicode client and a Unicode server are supported, so you need to use either a Unicode client or set the DB2 registry variable DB2CODEPAGE to 1208 prior to using the load, import, or export utilities.

Conversion from code page 1394 or 5488 to Unicode may result in expansion. For example, a 2-byte character may be stored as two 16-bit Unicode characters in the GRAPHIC columns. You need to ensure the target columns in the Unicode database are wide enough to contain any expanded Unicode byte.

Datetime Values

String Representations of Datetime Values

Values whose data types are DATE, TIME, or TIMESTAMP are represented in an internal form that is transparent to the SQL user. Dates, times, and time stamps can also, however, be represented by strings, and these representations directly concern the SQL user because there are no constants or variables whose data types are DATE, TIME, or TIMESTAMP. Thus, to be retrieved, a datetime value must be assigned to a string variable. The string representation is normally the default format of datetime values associated with the country/region code of the client, unless overridden by specification of the DATETIME format option when the program is precompiled or bound to the database.

When a valid string representation of a datetime value is used in an operation with an internal datetime value, the string representation is converted to the internal form of the date, time, or time stamp before the operation is performed. Valid string representations of datetime values are defined in the following sections.

Note: Graphic string representations of datetime values are supported only in Unicode databases.

Date Strings

A string representation of a date is a string that starts with a digit and has a length of at least 8 characters. Trailing blanks may be included; leading zeros may be omitted from the month part and the day part of the date.

The table "Formats for String Representations of Dates" remains unchanged.

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 you choose to omit seconds, an implicit specification of 0 seconds is assumed. Thus, 13:30 is equivalent to 13:30:00.

The table "Formats for String Representations of Times" remains unchanged.

Time Stamp Strings

A string representation of a time stamp is a string that starts with a digit and has a length of at least 16 characters. The complete string representation of a time stamp has the form *yyyy-mm-dd-hh.mm.ss.nnnnnn*. Trailing blanks may be included; leading zeros may be omitted from the month, day, or hour part of the time stamp, and microseconds may be truncated or omitted entirely. If

you choose to omit any digit of the microseconds part, an implicit
specification of 0 is assumed. Thus, 1991-3-2-8.30.00 is equivalent to
1991-03-02-08.30.00.000000.

Character Set Considerations
Date and time stamp strings must contain only digits and delimiter symbols.

Date and Time Formats
The string representation of date and time formats is the default format of
datetime values associated with the country/region code of the application.
This default format can be overridden by specifying the DATETIME format
option when the program is precompiled or bound to the database.

Administration Guide: Implementation

New Method for Specifying DMS containers on Windows 2000 and Later Systems

DB2 now uses a new method to specify DMS raw table space containers on Windows 2000 and later systems. Each basic disk partition or dynamic volume is assigned a globally unique identifier (GUID) at creation time. This GUID can be used as a device identifier when specifying the containers in a tablespace definition. Because the GUIDs are unique across the system, a multi-node configuration has unique GUID for each node, even if the disk partition definitions are the same.

A tool called db2listvolumes.exe has been provided to help display the GUIDs for all the disk volumes defined on a Windows system. The tool creates two files in the current directory that you run it. One file, volumes.xml, contains information about each disk volume. It is designed for easy viewing in any XML-enabled browser. The other file, tablespace.ddl, contains the required syntax for specifying the table space containers. Before you use tablespace.ddl, you must update it to include the remaining information needed for a table space definition.

The db2listvolumes tool does not require any command-line arguments.

Example for Extending Control Center

The example shown in the Extending the Control Center appendix is not correct and will not work. Use the following information to work with the Java example instead:

The sample program PluginEx.java is located in the samples/java subdirectory. PluginEx.java is installed with the DB2 Application Development client. To compile PluginEx.java, the following must be included in your *classpath*:

- On Windows platforms use:
  - DRIVE: \sqllib\java\swingall.jar
  - DRIVE: \sqllib\cc\com.jar
  - DRIVE: \sqllib\cc

  where DRIVE represents the drive on which DB2 is installed.
- On UNIX platforms use:
  - /u/db2inst1/sqllib/java/swingall.jar

– /u/db2inst1/sqllib/cc

where /u/db2inst1 represents the directory in which DB2 is installed.

Create the db2plug.zip to include all the classes generated from compiling PluginEx.java. The file should not be compressed. For example, issue the following:

```
zip -r0 db2plug.zip PluginEx*.class
```

This command places all the class files into the db2plug.zip file and preserves the relative path information.

Follow the instructions in the PluginEx.java file to compile and run the example.

The CCObject interface includes more static constants than are listed in the Extending the Control Center appendix of the Administration Guide. Below are the Java interfaces for extending the Control Center (CCExtension, CCObject, CCM enuAction, CCToolBarAction). These interfaces are listed here for reference only.

CCExtension::
```
// Licensed Materials -- Property of IBM
//
// (c) Copyright International Business Machines Corporation, 1999.
//     All Rights Reserved.
//
// US Government Users Restricted Rights -
// Use, duplication or disclosure restricted by
// GSA ADP Schedule Contract with IBM Corp.
//


package com.ibm.db2.tools.cc.navigator;


/**
 * The CCExtension interface allows users to extend the Control Center user
 * interface by adding new toolbar buttons, new menu items and
 * remove some predefined set of existing menu actions.
 *
 * To do so, create a java file which imports the
 * com.ibm.db2.tools.cc.navigator package and implements this interface.
 * The new file provides the implementation of the getObjects() and
 * getToolbarActions() function.
 *
 * The getObjects() function returns an array of CCObjects which defines
 * the existing
 * objects which the user would like to add new menu actions or remove
 * the alter or configure menu actions.
 *
 * The getToolbarActions() function returns an array of CCToolbarActions
 * which is added to the Control Center main toolbar.
```

```
                     *
                     * A single CCExtension subclass file or multiple CCExtension subclass
                     * files can be used to define the Control Center extensions.  In order
                     * for the Control Center to make use of these extensions, use the
                     * following setup procedures:
                     * (1) Create a "db2plug.zip" file which contains all the CCExtension
                     *     subclass files.  The files should not be compressed. For example,
                     *     if the CCExtension files are in the plugin package and they are
                     *     located in the plugin directory, issue
                     *        zip -r0 db2plug.zip plugin\*.class
                     *     This command will put all the plugin package class files into the
                     *     db2plug.zip file and preserve their relative path information.
                     * (2) To run WEBCC as an applet, put the db2plug.zip file in where the
                     *     <codebase> tag points to in the WEBCC html file.
                     *     To run the Control Center as an application, put
                     *     the db2plug.zip in a directory pointed to by the CLASSPATH
                     *     envirnoment variable and where the Control Center is run.
                     *
                     * For browsers that support multiple archives, just add "db2plug.zip"
                     * to the archive list of the WEBCC html page. Otherwise, all the
                     * CCExtension, CCObject, CCToolbarAction, CCMenuAction subclass files
                     * will have to be in their relative path depending on which package
                     * they belong to.
                     */

                    public interface CCExtension
                    {
                       /**
                        * Get an array of CCObject subclass objects which define
                        * a list of objects to be overrided in the
                        * Control Center
                        * @return CCObject[] CCObject subclass objects array
                        */
                       public CCObject[] getObjects();

                       /**
                        * Get an array of CCToolbarAction subclass objects which represent
                        * a list of buttons to be added to the Control Center
                        * main toolbar.
                        * @return CCToolbarAction[] CCToolbarAction subclass objects array
                        */
                       public CCToolbarAction[] getToolbarActions();
                    }
```

CCObject:

```
CCObject:
//
//   Licensed Materials -- Property of IBM
//
//   (c) Copyright International Business Machines Corporation, 1999.
//       All Rights Reserved.
//
//   US Government Users Restricted Rights -
```

```
//  Use, duplication or disclosure restricted by
//  GSA ADP Schedule Contract with IBM Corp.
//

package com.ibm.db2.tools.cc.navigator;

/**
 * The CCObject interface allows users to define a new object to be
 * inserted into the Control Center tree or changing the behavior of the
 * menu actions of an existing object.
 */
public interface CCObject
{
   /**
    * The following static constants defines a list of object type
    * available to be added to the Control Center tree.
    */
   public static final int UDB_SYSTEMS_FOLDER                          = 0;
   public static final int UDB_SYSTEM                                  = 1;
   public static final int UDB_INSTANCES_FOLDER                        = 2;
   public static final int UDB_INSTANCE                                = 3;
   public static final int UDB_DATABASES_FOLDER                        = 4;
   public static final int UDB_DATABASE                                = 5;
   public static final int UDB_TABLES_FOLDER                           = 6;
   public static final int UDB_TABLE                                   = 7;
   public static final int UDB_TABLESPACES_FOLDER                      = 8;
   public static final int UDB_TABLESPACE                              = 9;
   public static final int UDB_VIEWS_FOLDER                            = 10;
   public static final int UDB_VIEW                                    = 11;
   public static final int UDB_ALIASES_FOLDER                          = 12;
   public static final int UDB_ALIAS                                   = 13;
   public static final int UDB_TRIGGERS_FOLDER                         = 14;
   public static final int UDB_TRIGGER                                 = 15;
   public static final int UDB_SCHEMAS_FOLDER                          = 16;
   public static final int UDB_SCHEMA                                  = 17;
   public static final int UDB_INDEXES_FOLDER                          = 18;
   public static final int UDB_INDEX                                   = 19;
   public static final int UDB_CONNECTIONS_FOLDER                      = 20;
   public static final int UDB_CONNECTION                              = 21;
   public static final int UDB_REPLICATION_SOURCES_FOLDER              = 22;
   public static final int UDB_REPLICATION_SOURCE                      = 23;
   public static final int UDB_REPLICATION_SUBSCRIPTIONS_FOLDER        = 24;
   public static final int UDB_REPLICATION_SUBSCRIPTION                = 25;
   public static final int UDB_BUFFERPOOLS_FOLDER                      = 26;
   public static final int UDB_BUFFERPOOL                              = 27;
   public static final int UDB_APPLICATION_OBJECTS_FOLDER              = 28;
   public static final int UDB_USER_DEFINED_DISTINCT_DATATYPES_FOLDER  = 29;
   public static final int UDB_USER_DEFINED_DISTINCT_DATATYPE          = 30;
   public static final int UDB_USER_DEFINED_DISTINCT_FUNCTIONS_FOLDER  = 31;
   public static final int UDB_USER_DEFINED_DISTINCT_FUNCTION          = 32;
   public static final int UDB_PACKAGES_FOLDER                         = 33;
   public static final int UDB_PACKAGE                                 = 34;
   public static final int UDB_STORE_PROCEDURES_FOLDER                 = 35;
   public static final int UDB_STORE_PROCEDURE                         = 36;
   public static final int UDB_USER_AND_GROUP_OBJECTS_FOLDER           = 37;
```

```
public static final int UDB_DB_USERS_FOLDER                    = 38;
public static final int UDB_DB_USER                            = 39;
public static final int UDB_DB_GROUPS_FOLDER                   = 40;
public static final int UDB_DB_GROUP                           = 41;
public static final int UDB_DRDA_TABLES_FOLDER                 = 42;
public static final int UDB_DRDA_TABLE                         = 43;
public static final int UDB_NODEGROUPS_FOLDER                  = 44;
public static final int UDB_NODEGROUP                          = 45;


public static final int S390_SUBSYSTEMS_FOLDER                 = 46;
public static final int S390_SUBSYSTEM                         = 47;
public static final int S390_BUFFERPOOLS_FOLDER                = 48;
public static final int S390_BUFFERPOOL                        = 49;
public static final int S390_VIEWS_FOLDER                      = 50;
public static final int S390_VIEW                              = 51;
public static final int S390_DATABASES_FOLDER                  = 52;
public static final int S390_DATABASE                         = 53;
public static final int S390_TABLESPACES_FOLDER               = 54;
public static final int S390_TABLESPACE                       = 55;
public static final int S390_TABLES_FOLDER                    = 56;
public static final int S390_TABLE                           = 57;
public static final int S390_INDEXS_FOLDER                    = 58;
public static final int S390_INDEX                           = 59;
public static final int S390_STORAGE_GROUPS_FOLDER           = 60;
public static final int S390_STORAGE_GROUP                    = 61;
public static final int S390_ALIASES_FOLDER                   = 62;
public static final int S390_ALIAS                           = 63;
public static final int S390_SYNONYMS_FOLDER                  = 64;
public static final int S390_SYNONYM                         = 65;
public static final int S390_APPLICATION_OBJECTS_FOLDER       = 66;
public static final int S390_COLLECTIONS_FOLDER              = 67;
public static final int S390_COLLECTION                      = 68;
public static final int S390_PACKAGES_FOLDER                 = 69;
public static final int S390_PACKAGE                         = 70;
public static final int S390_PLANS_FOLDER                    = 71;
public static final int S390_PLAN                            = 72;
public static final int S390_PROCEDURES_FOLDER               = 73;
public static final int S390_PROCEDURE                       = 74;
public static final int S390_DB_USERS_FOLDER                 = 75;
public static final int S390_DB_USER                        = 76;
public static final int S390_LOCATIONS_FOLDER               = 77;
public static final int S390_LOCATION                       = 78;
public static final int S390_DISTINCT_TYPES_FOLDER         = 79;
public static final int S390_DISTINCT_TYPE                  = 80;
public static final int S390_USER_DEFINED_FUNCTIONS_FOLDER  = 81;
public static final int S390_USER_DEFINED_FUNCTION         = 82;
public static final int S390_TRIGGERS_FOLDER               = 83;
public static final int S390_TRIGGER                       = 84;
public static final int S390_SCHEMAS_FOLDER                = 85;
public static final int S390_SCHEMA                        = 86;
public static final int S390_CATALOG_TABLES_FOLDER        = 87;
public static final int S390_CATALOG_TABLE                 = 88;
public static final int DCS_GATEWAY_CONNECTIONS_FOLDER     = 89;
public static final int DCS_GATEWAY_CONNECTION             = 90;
public static final int S390_UTILITY_OBJECTS_FOLDER       = 91;
```

```
public static final int S390_DATASET_TEMPLATES_FOLDER            = 92;
public static final int S390_DATASET_TEMPLATE                    = 93;
public static final int S390_UTILITY_LISTS_FOLDER                = 94;
public static final int S390_UTILITY_LIST                        = 95;
public static final int S390_UTILITY_PROCEDURES_FOLDER           = 96;
public static final int S390_UTILITY_PROCEDURE                   = 97;
/**
 * Total number of object types
 */
public static final int NUM_OBJECT_TYPES                         = 98;


/**
 * Get the name of these object
 *
 * The function returns the name of this object. This name
 * can be of three types:
 * (1) Fully qualified name
 *     Syntax: xxxxx-yyyyy-zzzzz
 *             where xxxxx-yyyyy is the fully quality name of the parent
 *             object and zzzzz is the name of the new object.
 *     Note: Parent and child object name is separated by '-' character.
 *     If a schema name is required to identify object, the fully
 *     qualified name is represented by xxxxx-yyyyy-wwwww.zzzzz
 *     where wwwww is the schema name.
 *     Only the behavior of the object that match this fully
 *     quality name will be affected.
 * (2) Parent fully qualified name
 *     Syntax: xxxxx-yyyyy
 *             where xxxxx-yyyyy is the fully qualified name of the
 *             parent object.
 *     When the object type is folder (ie. DATABASES_FOLDER), the
 *     getName() should only return the fully qualified name of the
 *     folder's parent.
 *     Only the behavior of the object that match this name
 *     and the specific type return by the getType() function will be
 *     affected.
 * (3) null
 *     Syntax: null
 *     If null is return, the CCMenuActions returns by the
 *     getMenuActions() call will be applied to all objects of type
 *     returns by the getType() call.
 * @return String object name
 */
public String getName();

/**
 * Get the type of this object
 * @return int return one of the static type constants defined in this
 * interface
 */
public int getType();

/**
 * Get the CCMenu Action array which defines the list of menu actions
 *  to be created for object
```

```
                          * return CCMenuAction[] CCMenuAction array
                          */
                        public CCMenuAction[] getMenuActions();

                        /**
                         * Check if this object is editable.  If not, the Alter related menu
                         * items will be removed from the object's popup menu
                         * return boolean If false, the Alter menu item will be remove from the
                         * object's popup menu.
                         * Return true if you do not wish to modify current Alter menu item
                         * behaviour.
                         */
                        public boolean isEditable();

                        /**
                         * Check if this object is configurable.  If not, the configuration
                         * related menu items will be removed from the object's popup menu
                         * return boolean If false, the Configuration related menu item will be
                         *  removed from the object's popup menu.
                         * Return true if you do not wish to modify current Configuration
                         * behaviour.
                         */
                        public boolean isConfigurable();
                    }
```

CCMenuAction::

```
//
//  Licensed Materials -- Property of IBM
//
//  (c) Copyright International Business Machines Corporation, 1999.
//      All Rights Reserved.
//
//  US Government Users Restricted Rights -
//  Use, duplication or disclosure restricted by
//  GSA ADP Schedule Contract with IBM Corp.
//

package com.ibm.db2.tools.cc.navigator;
import java.awt.event.*;
import javax.swing.*;

/**
 * The CCMenuAction class allows users to define a new menu item to be added
 * to a Control Center object.  The new menu item will be added at the end of
 * an object's popup menu.
 *
 * Note: If the object has a Control Center Refresh and/or
 * Filter menu item, the new menu item will be inserted before the Refresh
 * and Filter menu. The Control Center Refresh and Filter menu items are
 * always  at the end of the popup menu.
 */
public interface CCMenuAction
```

```
{
   /**
    * Get the name of this action
    * @return String Name text on the menu item
    */
   public String getMenuText();

   /**
    * Invoked when an action occurs.
    * @param e Action event
    */
   public void actionPerformed(ActionEvent e);
}
```

CCToolBarAction:

```
//  Licensed Materials -- Property of IBM
//
//  (c) Copyright International Business Machines Corporation, 1999.
//      All Rights Reserved.
//
//  US Government Users Restricted Rights -
//  Use, duplication or disclosure restricted by
//  GSA ADP Schedule Contract with IBM Corp.
//

package com.ibm.db2.tools.cc.navigator;
import java.awt.event.*;
import javax.swing.*;

/**
 * The CCToolbarAction interface class allows users to define a new action
 *  to be added to the Control Center toolbar.
 */
public interface CCToolbarAction
{
   /**
    * Get the name of this action
    * @return String Name text on the menu item, or toolbar button hover help
    */
   public String getHoverHelpText();

   /**
    * Get the icon for the toolbar button
    * Any toolbar CCAction should override this function and return
    * a valid ImageIcon object. Otherwise, the button will have no icon.
    * @return ImageIcon Icon to be displayed
    */
   public ImageIcon getIcon();

   /**
    * Invoked when an action occurs.
```

```
 * @param e Action event
 */
public void actionPerformed(ActionEvent e);
}
```

Administration Guide: Performance

System Temporary Table Schemas

The schema for a system temporary table is determined by the application and authorization ID that create it. When this data is available, the schema in which the table is created is <AUTHID><APPLID>. Under some circumstances, the tables are created using only one of these IDs to determine the schema, and sometimes, none. This can result in a tables such as AUTHID.TEMPTABLENAME, or .TEMPTABLENAME. You can view the schema information for these tables by using the GET SNAPSHOT command. For information on this command, please refer to the Command Reference

Chapter 8. Operational Performance

Block- Based Buffer Pool

This feature is only supported on the Sun Solaris Operating Environment.

Due to I/O overhead, prefetching pages from disk is an expensive operation. DB2's prefetching significantly improves throughput when processing can be overlapped with I/O. Most platforms provide high performance primitives to read contiguous pages from disk into discontiguous portions of memory. These primitives are usually called "scattered read" or "vectored I/O". On some platforms, the performance of these primitives cannot compete with doing I/O in large block sizes. By default, the buffer pools are page-based. That is, contiguous pages on disk are prefetched into discontiguous pages in memory. Prefetching performance can be further enhanced on these platforms if pages can be read from disk into contiguous pages in a buffer pool. A registry variable, DB2_BLOCK_BASED_BP, allows you to create a section in the buffer pool that holds sets of contiguous pages. These sets of contiguous pages are referred to as "blocks". By setting this registry variable, a sequential prefetch will read the pages from disk directly into these blocks instead of reading each page individually. This will improve I/O performance. For more information on this registry variable, see the 'Registry and Environment Variables' section of the Administration Guide.

Multiple table spaces of different extent sizes can be bound to a buffer pool of the same block size. There is a close relationship between extent sizes and block sizes even though they deal with separate concepts. An extent is the granularity at which table spaces are striped across multiple containers. A block is the only granularity at which I/O servers doing sequential prefetch requests will consider doing block-based I/O.

107

Individual sequential prefetch requests use extent-size pages. When such a prefetch request is received, the I/O server determines the cost and benefit of doing each request as a block-based I/O (if there is a block-based area in the buffer pool) instead of the page-based I/O using the scattered read method. The benefit of doing any I/O as block-based I/O is the performance benefit from reading from contiguous disk into contiguous memory. The cost is the amount of wasted buffer pool memory that can result from using this method.

Buffer pool memory can be wasted for two reasons when doing block-based I/O:
- The number of pages in the prefetch request contains fewer pages than the number of pages in a block. That is, the extent size is smaller than the block size.
- Some of the pages requested as part of the prefetch request are already in the page area of the buffer pool.

Note: Each block in the block-based area of a buffer pool cannot be subdivided. The pages within the block must all be contiguous. As a result, there is a possibility of wasted space.

The I/O server allows for some wasted pages within each block in order to gain the benefit of doing block-based I/O. However, when too much of a block is wasted, the I/O server will revert to using page-based prefetching into the page area of the buffer pool. As a result, some of the I/O done during prefetching will not be block-based. This is not an optimal condition.

For optimal performance, you should have table spaces of the same extent size bound to a buffer pool of the same block size. Good performance can still be achieved if the extent size of some table spaces is greater than the block size of the buffer pool they are bound to. It is not advisable to bind table spaces to a buffer pool when the extent size is less than the block size.

Note: The block area of a buffer pool is only used for sequential prefetching. If there is little or no sequential prefetching involved on your system, then the block area will be a wasted portion of the buffer pool.

Both AWE and block-based support cannot be setup for a buffer pool at the same time. If both the DB2_AWE and DB2_BLOCK_BASED_BP registry variables refer to the same buffer pool, precedence will be given to AWE. Block-based support will be disabled in this case and will only be re-enabled once AWE is disabled.

A buffer pool that is using extended storage does not support block-based I/O.

Block-based Buffer Pool Examples

Before working with any of the examples, you will need to know the identifiers for the buffer pools on your system. The ID of the buffer pool can be seen in the BUFFERPOOLID column of the SYSCAT.BUFFERPOOLS system catalog view.

Scenario 1

You have a buffer pool with an ID of 4 that has 1000 pages. You wish to create a block area which is made up of 700 pages where each block contains 32 pages. You must run the following:

```
db2set DB2_BLOCK_BASED_BP=4,700,32
```

When the database is started, the buffer pool with ID 4 is created with a block area of 672 pages and a page area of 328 pages. In this example, 32 cannot be evenly divided into 700. This means that the block area size specified had to be reduced to the nearest block size boundary using the following formula:

```
        ((block area size))
    FLOOR(-----------------) X block size
        ( (block size)    )
        (       700       )
  = FLOOR(-----------------) X 32
        (        32       )
  = 21 x 32
  = 672
```

Scenario 2

You have a buffer pool with an ID of 11 that has 3000 pages. You wish to create a block area which is made up of 2700 pages. You must run the following:

```
db2set DB2_BLOCK_BASED_BP=11,2700
```

When the database is started, the buffer pool with ID 11 is created with a block area of 2688 pages and a page area of 312 pages. With no value explicitly given for the block size, the default value of 32 is used. In this example, 32 cannot be evenly divided into 2700. This means that the block area size specified had to be reduced to the nearest block size boundary using the following formula:

```
        ((block area size))
    FLOOR(-----------------) X block size
        ( (block size)    )
        (      2700       )
  = FLOOR(-----------------) X 32
        (        32       )
  = 84 x 32
  = 2688
```

Chapter 10. Scaling Your Configuration Through Adding Processors

Problems When Adding Nodes to a Partitioned Database

When adding nodes to a partitioned database that has one or more system temporary table spaces with a page size that is different from the default page size (4 KB), you may encounter the error message: "SQL6073N Add Node operation failed" and an SQLCODE. This occurs because only the IBMDEFAULTBP buffer pool exists with a page size of 4 KB when the node is created.

For example, you can use the **db2start** command to add a node to the current partitioned database:

```
DB2START NODENUM 2 ADDNODE HOSTNAME newhost PORT 2
```

If the partitioned database has system temporary table spaces with the default page size, the following message is returned:

```
SQL6075W The Start Database Manager operation successfully added the node.
The node is not active until all nodes are stopped and started again.
```

However, if the partitioned database has system temporary table spaces that are not the default page size, the returned message is:

```
SQL6073N Add Node operation failed. SQLCODE = "<-902>"
```

In a similar example, you can use the ADD NODE command after manually updating the db2nodes.cfg file with the new node description. After editing the file and running the ADD NODE command with a partitioned database that has system temporary table spaces with the default page size, the following message is returned:

```
DB20000I The ADD NODE command completed successfully.
```

However, if the partitioned database has system temporary table spaces that are not the default page size, the returned message is:

```
SQL6073N Add Node operation failed. SQLCODE = "<-902>"
```

One way to prevent the problems outlined above is to run:

```
DB2SET DB2_HIDDENBP=16
```

before issuing **db2start** or the ADD NODE command. This registry variable enables DB2 to allocate hidden buffer pools of 16 pages each using a page size different from the default. This enables the ADD NODE operation to complete successfully.

Another way to prevent these problems is to specify the WITHOUT TABLESPACES clause on the ADD NODE or the **db2start** command. After doing this, you will have to create the buffer pools using the CREATE

BUFFERPOOL statement, and associate the system temporary table spaces to the buffer pool using the ALTER TABLESPACE statement.

When adding nodes to an existing nodegroup that has one or more table spaces with a page size that is different from the default page size (4 KB), you may encounter the error message: "SQL0647N Bufferpool "" is currently not active.". This occurs because the non-default page size buffer pools created on the new node have not been activated for the table spaces.

For example, you can use the ALTER NODEGROUP statement to add a node to a nodegroup:

```
DB2START
CONNECT TO mpp1
ALTER NODEGROUP ng1 ADD NODE (2)
```

If the nodegroup has table spaces with the default page size, the following message is returned:

```
SQL1759W Redistribute nodegroup is required to change data positioning for
objects in nodegroup "<ng1>" to include some added nodes or exclude
some drop nodes.
```

However, if the nodegroup has table spaces that are not the default page size, the returned message is:

```
SQL0647N Bufferpool "" is currently not active.
```

One way to prevent this problem is to create buffer pools for each page size and then to reconnect to the database before issuing the ALTER NODEGROUP statement:

```
DB2START
CONNECT TO mpp1
CREATE BUFFERPOOL bp1 SIZE 1000 PAGESIZE 8192
CONNECT RESET
CONNECT TO mpp1
ALTER NODEGROUP ng1 ADD NODE (2)
```

A second way to prevent the problem is to run:

```
DB2SET DB2_HIDDENBP=16
```

before issuing the **db2start** command, and the CONNECT and ALTER NODEGROUP statements.

Another problem can occur when the ALTER TABLESPACE statement is used to add a table space to a node. For example:

```
DB2START
CONNECT TO mpp1
ALTER NODEGROUP ng1 ADD NODE (2) WITHOUT TABLESPACES
ALTER TABLESPACE ts1 ADD ('ts1') ON NODE (2)
```

This series of commands and statements generates the error message
SQL0647N (not the expected message SQL1759W).

To complete this change correctly, you should reconnect to the database after
the ALTER NODEGROUP... WITHOUT TABLESPACES statement.

```
DB2START
CONNECT TO mpp1
ALTER NODEGROUP ng1 ADD NODE (2) WITHOUT TABLESPACES
CONNECT RESET
CONNECT TO mpp1
ALTER TABLESPACE ts1 ADD ('ts1') ON NODE (2)
```

Another way to prevent the problem is to run:

```
DB2SET DB2_HIDDENBP=16
```

before issuing the **db2start** command, and the CONNECT, ALTER
NODEGROUP, and ALTER TABLESPACE statements.

Chapter 13. Configuring DB2

Log Archive Completion Now Checked More Frequently

To improve recovery time by avoiding unnecessary log archive requests, the
database server now checks for log archive completion both when a new log
file is created and when the first active log changes.

Correction to Collating Information (collate_info) Section

The documentation for collating information incorrectly states that the
collate_info parameter can only be displayed by using the GET DATABASE
CONFIGURATION API. This is incorrect. You cannot use the GET DATABASE
CONFIGURATION API to display the collate_info parameter. Instead, you
must use the db2CfgGetAPI.

DB2 Registry and Environment Variables

Corrections to Performance Variables

Table 5. Performance Variables

Variable Name	Operating System	Values
Description		
DB2_BINSORT	All	Default=YES
		Values: YES or NO

Table 5. Performance Variables (continued)

Variable Name	Operating System	Values
Description		
Enables a new sort algorithm that reduces the CPU time and elapsed time of sorts. This new algorithm extends the extremely efficient integer sorting technique of DB2 UDB to all sort datatypes such as BIGINT, CHAR, VARCHAR, FLOAT, and DECIMAL, as well as combinations of these datatypes. To enable this new algorithm, use the following command: `db2set DB2_BINSORT = yes`		
DB2_BLOCK_BASED_BP	Solaris Operating Environment	Default=None Values: dependant on parameters
Specifies the values needed to create a block area within a buffer pool. The ID of the buffer pool is needed and can be seen in the BUFFERPOOLID column of the SYSCAT.BUFFERPOOLS system catalog view. The number of pages to be allocated in the buffer pool to block-based I/O must be given. The number of pages to include in a block is optional, with a default value of 32. The format for the use of this registry variable is: `DB2_BLOCK_BASED_BP=BUFFER POOL ID,BLOCK AREA SIZE,[BLOCK SIZE];...` Multiple buffer pools can be defined as block-based using the same variable with a semi-colon separating the entries. The value for BLOCK SIZE can range from 2 to 256. If no BLOCK SIZE is given, the default used is 32. If the BLOCK AREA SIZE specified is larger than 98% of the total buffer pool size, then the buffer pool will not be made block-based. It is a good idea to always have some portion of the buffer pool in the page-based area of the buffer pool because there is a possibility of individual pages being required even if the majority of the I/O on the system is sequential prefetching. If the value specified for BLOCK AREA SIZE is not a multiple of BLOCK SIZE, it is reduced to the nearest block size boundary. For more information on block-based I/O, see "Block- Based Buffer Pool" on page 107.		
DB2_NO_FORK_CHECK	UNIX	Default=OFF Values: ON or OFF
When this variable is "ON", the client process will not protect itself against an application making a copy of the process to be run (called forking). When forking occurs, the results are unpredictable. The results could range from no effect, to some bad results, to some error code being returned, to a trap in the application. If you are certain that your application does not fork and you want better performance, you should change the value of this variable to "ON".		
DB2_MINIMIZE_LIST_PREFETCH	All	Default=NO Values: YES or NO

Table 5. Performance Variables (continued)

Variable Name	Operating System	Values
Description		
List prefetch is a special table access method that involves retrieving the qualifying RIDs from the index, sorting them by page number and then prefetching the data pages. Sometimes the optimizer does not have accurate information to determine if list prefetch is a good access method. This might occur when predicate selectivities contain parameter markers or host variables that prevent the optimizer from using catalog statistics to determine the selectivity. This registry variable will prevent the optimizer from considering list prefetch in such situations.		
DB2_INLIST_TO_NLJN	All	Default=NO Values: YES or NO

In some situations, the SQL compiler can rewrite an IN list predicate to a join. For example, the following query:
 	
 ```\nSELECT *\n FROM EMPLOYEE\n WHERE DEPTNO IN ('D11', 'D21', 'E21')\n```
 could be written as:
 ```\nSELECT *\n FROM EMPLOYEE, (VALUES 'D11', 'D21', 'E21') AS V(DNO)\n WHERE DEPTNO = V.DNO\n```
 This revision might provide better performance if there is an index on DEPTNO. The list of values would be accessed first and joined to EMPLOYEE with a nested loop join using the index to apply the join predicate.
 	
 Sometimes the optimizer does not have accurate information to determine the best join method for the rewritten version of the query. This can occur if the IN list contains parameter markers or host variables which prevent the optimizer from using catalog statistics to determine the selectivity. This registry variable will cause the optimizer to favor nested loop joins to join the list of values, using the table that contributes the IN list as the inner table in the join.		

New Parameters for Registry Variable DB2BPVARS

The registry variable DB2BPVARS supports two new parameters: NUMPREFETCHQUEUES and PREFETCHQUEUESIZE. These parameters are applicable to all platforms and can be used to improve buffer-pool data prefetching. For example, consider sequential prefetching in which the desired PREFETCHSIZE is divided into PREFETCHSIZE/EXTENTSIZE prefetch requests. In this case, requests are placed on prefetch queues from which I/O servers are dispatched to perform asynchronous I/O. By default, DB2 maintains one queue of size max(100 , 2*NUM_IOSERVERS) for each database partition. In some environments, performance improves with either more queues, queues of a different size, or both. The number of prefetch

queues should be at most one half of the number of I/O servers. When you set these parameters, consider other parameters such as PREFETCHSIZE, EXTENTSIZE, NUM_IOSERVERS, buffer-pool size, and DB2_BLOCK_BASED_BP, as well as workload characteristics such as the number of current users.

If you think the default values are too small for your environment, first increase the values only slightly. For example, you might set NUMPREFETCHQUEUES=4 and PREFETCHQUEUESIZE=200. Make changes to these parameters in a controlled manner so that you can monitor and evaluate the effects of the change.

Table 6. Summary of New Parameters

Parameter name	Default value	Valid range
NUMPREFETCHQUEUES	1	1 to NUM_IOSERVERS
		if set to less than 1, adjusted to 1
		if set to greater than NUM_IOSERVERS, adjusted to NUM_IOSERVERS
PREFETCHQUEUESIZE	max(100,2*NUM_IOSERVERS)	1 to 32767
		if set to less than 1, adjusted to default
		if set to greater than 32767, adjusted to 32767

Corrections and Additions to Miscellaneous Registry Variables

The DB2_NEWLOGPATH2 registry variable is available for *all* operating systems. A new variable, DB2_ROLLFORWARD_NORETRIEVE, has been introduced. The correct information for both variables appears below.

Table 7. Miscellaneous Variables

Variable Name	Operating System	Values
Description		
DB2_NEWLOGPATH2	ALL	Default=NO
		Values: YES or NO
This parameter allows you to specify whether a secondary path should be used to implement dual logging. The path used is generated by appending a "2" to the current value of the *logpath* database configuration parameter.		
DB2_ROLLFORWARD_NORETRIEVE	ALL	Default=(not set)
		Values: YES or NO

| *Table 7. Miscellaneous Variables (continued)*

Variable Name	Operating System	Values
Description		
If the database configuration parameter USEREXIT is enabled, log files are automatically retrieved from the archive during rollforward operations. The DB2_ROLLFORWARD_NORETRIEVE variable lets you specify that rollforward operations should not retrieve log files from the archive. This variable is disabled by default. Set this variable to YES if you do not want rollforward to retrieve log files automatically. For example, set the variable to YES in a hot-standby setup when want to keep log records created by a bad application from corrupting the backup system.		

Corrections and Additions to General Registry Variables

A new variable, DB2_REDUCED_OPTIMIZATION, has been introduced.

Table 8. General Registry Variable

Variable Name	Operating System	Values
Description		
DB2_REDUCED_OPTIMIZATION	ALL	Default=NO
		Values: YES, NO, or any integer

Table 8. General Registry Variable (continued)

Variable Name	Operating System	Values
Description		

This registry variable lets you disable some of the optimization techniques used at specific optimization levels. If you reduce the number of optimization techniques used, you also reduce time and resource use during optimization.
Note: Although optimization time and resource use might be reduced, the risk of producing a less-than-optimal data access plan is increased.

- If set to NO

  The optimizer does not change its optimization techniques.

- If set to YES

  If the optimization level is 5 (the default) or lower, the optimizer disables some optimization techniques that might consume significant prepare time and resources but that do not usually produce a better access plan.

  If the optimization level is exactly 5, the optimizer scales back or disables some additional techniques, which might further reduce optimization time and resource use, but also further increase the risk of a less-than-optimal access plan. For optimization levels lower than 5, some of these techniques might not be in effect in any case. If they are, however, they remain in effect.

- If set to any integer

  The effect is the same as if the value is set to YES, with the following additional behavior for dynamically prepared queries optimized at level 5: If the total number of joins in any query block exceeds the setting, then the optimizer switches to greedy join enumeration instead of disabling additional optimization techniques as described above for optimization level 5, which implies that the query will be optimized at a level similar to optimization level 2.

  For information about greedy and dynamic join enumeration, see "Search Strategies for Selecting Optimal Join" in *Administration Guide: Performance*.

Note that the dynamic optimization reduction at optimization level 5, as described in "Adjusting the Optimization Class" in *Administration Guide: Performance*, takes precedence over the behavior described for optimization level of exactly 5 when DB2_REDUCED_OPTIMIZATION is set to YES as well as over the behavior described for the integer setting.

Administering Satellites Guide and Reference

Setting up Version 7.2 DB2 Personal Edition and DB2 Workgroup Edition as Satellites

The sections that follow describe how to set up Windows-based Version 7.2 DB2 Personal Edition and DB2 Workgroup Edition systems so that they can be used as fully functional satellites in a satellite environment. For information about the terms and concepts used in the information that follows, refer to the *Administering Satellites Guide and Reference*. You can find this book at the following URL: `http://www.ibm.com/cgi-bin/db2www/data/db2/udb/winos2unix/support/v6pubs.d2w/en_main`

For Technotes that supplement the information in the *Administering Satellites Guide and Reference*, refer to the following URL: http://www.ibm.com/cgi-bin/db2www/data/db2/udb/winos2unix/support/index.d2w/report

Prerequisites

To set up either DB2 Personal Edition or DB2 Workgroup Edition as satellites, you require the following:

1. A DB2 control server

    The DB2 control server is a DB2 Enterprise Edition system that runs on Windows NT or AIX, and has the Control Server component installed. The DB2 Enterprise Edition system that you use must be at Version 6 with FixPak 2 or higher, or Version 7 at any FixPak level.

    *   If you have a Version 6 Enterprise Edition system that you want to use as the DB2 control server, see "Installing FixPak 2 or Higher on a Version 6 Enterprise Edition System" on page 123.

    *   If you are using Version 7 and do not have the Control Server component installed, install this component, re-install any FixPaks that you have already installed, then create the DB2 control server instance and satellite control database. Refer to the *Administering Satellites Guide and Reference* for instructions on creating these objects.

    Note: If you are installing a Version 7.2 Enterprise Edition system on Windows NT for use as the DB2 control server, and you want to perform a response file installation, see the Technote entitled *DB2 Control Server Response File Keywords* for information about the keywords to specify in the response file.

2. The DB2 control server instance and the satellite control database

    The DB2 control server instance is typically called DB2CTLSV, and the satellite control database is called SATCTLDB. The DB2 control server

119

instance and the satellite control database are on the Enterprise Edition system, and, on Windows NT, are automatically created when you install DB2 with the Control Server component. If you install DB2 on AIX, see the *Administering Satellites Guide and Reference* for information about creating the DB2 control server instance and the satellite control database.

3. The Satellite Administration Center

   The Satellite Administration Center is the set of GUI tools that you use to set up and administer the satellite environment. You access this set of tools from the Control Center. For more information about the Satellite Administration Center and the satellite environment, see the *Administering Satellites Guide and Reference*, and the online help that is available from the Satellite Administration Center. If you are running a Version 6 Control Center, see "Upgrading a Version 6 Control Center and Satellite Administration Center" on page 132.

   If you have not already used the Satellite Administration Center to set up the satellite environment and to create the object that represents the new satellite in the Satellite Administration Center, you should do so before installing the satellite. For more information, see the description of how to set up and test a satellite environment in the *Administering Satellites Guide and Reference*.

4. A Version 7.2 Personal Edition or Workgroup Edition system that you want to use as a satellite.

Installation Considerations
When you install either DB2 Personal Edition or DB2 Workgroup Edition, you do not have to select any special component to enable either system to synchronize. If you intend to perform a response file installation, see "Performing a Response File Installation" for the keywords that you should specify when installing the Version 7.2 system. If you are performing an interactive installation of your Version 7.2 system, see "Configuring the Version 7.2 System for Synchronization" on page 122 after you finish installing DB2 for values that you must set at the Version 7.2 system to enable it to synchronize.

Performing a Response File Installation: If you are performing a response file installation of Version 7.2 DB2 Personal Edition or DB2 Workgroup Edition, you can set the following keywords in the response file.

If you decide to not specify one or more of these keywords during the response file installation, see "Configuring the Version 7.2 System for Synchronization" on page 122 for additional steps that you must perform after installing DB2 to enable the Version 7.2 system to synchronize. You can also use the instructions in this section if you want to change any values that were specified during the response file installation.

db2.db2satelliteid

Sets the satellite ID on the system.

> **Note:** If you do not specify this keyword, the satellite ID is automatically set to the user ID that was used to install DB2. If you want to use this user ID as the satellite ID, you do not have to specify a value for this keyword.

db2.db2satelliteappver

Sets the application version on the system.

> **Note:** If you do not specify this keyword, the application version on the satellite is automatically set to V1R0M00. If you want to use this value as the application version, you do not have to specify a value for this keyword.

db2.satctldb_username

Sets the user name to be used for the system to connect to the satellite control database.

db2.satctldb_password

Sets the password that the user name passes to the DB2 control server when the user name connects to the satellite control database.

After you complete the response file installation, the Version 7.2 system is ready to synchronize. You should issue the **db2sync -t** command on the satellite to verify that the values specified on the satellite are correct, and that the satellite can connect to the satellite control database.

For additional information about performing a response file installation, refer to the *Administering Satellites Guide and Reference*.

Notes:

1. In Version 7, user IDs and passwords are required for the creation of all services on Windows NT and Windows 2000. These user IDs and passwords are specified in the response file by keyword pairs. The first keyword pair found in the response file becomes the default user ID and password for all services, unless you provide an override for a service by specifying the specific keyword pair for that service.

   In Version 6, the **admin.userid** and the **admin.password** keywords could be specified during a response file installation of DB2 Satellite Edition to specify the user ID and password that would be used by the Remote Command Service. For Version 7.2 Personal Edition and Workgroup Edition, if you specify these keywords, they are used for the DB2DAS00 instance on the Version 7.2 system. For a DB2 Version 7.2 system, the Remote Command Service will use the user ID and password that is used

by the DB2 instance on the system. If you do not specify values for
db2.userid and **db2.password**, the defaulting rule described above applies.

2. In Version 6, you could create a database when installing DB2 Satellite
   Edition using a response file installation. You cannot create a database
   during a response file installation on the Version 7.2 Personal Edition or
   Workgroup Edition system that you intend to use as a satellite. The
   following keywords (which are described in the *Administering Satellites
   Guide and Reference*), are not supported:
   - **db2.userdb_name**
   - **db2.userdb_recoverable**
   - **db2.userdb_rep_src**

Configuring the Version 7.2 System for Synchronization

If you install the Version 7.2 system interactively, several values must be set
on the DB2 Personal Edition or DB2 Workgroup Edition system after installing
DB2 before the system can synchronize.

Note: You can execute an operating system script on the system to set all
values at the satellite except for the user ID and password that the
satellite uses to connect to the satellite control database (see step 4 on
page 123).

1. Set the satellite ID by using the **db2set** command.

   If you install DB2 Personal Edition or DB2 Workgroup Edition
   interactively, the satellite ID is automatically set to the user ID that was
   used to install DB2. If you want to use this user ID as the satellite ID, you
   do not have to perform this step. For information about setting the satellite
   ID, see the *Administering Satellites Guide and Reference*.

2. Set the application version on the satellite by using the **db2sync -s**
   command.

   If you install DB2 Personal Edition or DB2 Workgroup Edition
   interactively, the application version on the satellite is automatically set to
   V1R0M00. If you want to use this value as the application version, you do
   not have to perform this step.

   You can use the **db2sync -g** command on the satellite to view the current
   setting of the application version. If you want to change this value, issue
   the **db2sync -s** command. You are prompted to provide a new value for
   the application version. For more information about setting the application
   version, see the *Administering Satellites Guide and Reference*.

3. Issue the **catalog node** and **catalog database** commands on the satellite to
   catalog the DB2 control server instance and the satellite control database,
   SATCTLDB, at the satellite.

   You can also use the **db2sync -t** command on the satellite to open the DB2
   Synchronizer application in test mode. If the SATCTLDB database is not

cataloged at the satellite when you issue the command, the Catalog Control Database window opens. You can either use the DB2 discovery feature that is available from the Catalog Control Database window to catalog the DB2 control server and the SATCTLDB database, or you can type the hostname and server name in this window. You will also be prompted to specify the user ID and password that the satellite will use to connect to the satellite control database, as described in step 4.

> **Note:** After you install Version 7.2 DB2 Personal Edition or DB2 Workgroup Edition interactively, the DB2 Synchronizer does not start automatically in test mode (as was the case for Version 6 DB2 Satellite Edition).

4. Issue the **db2sync -t** command on the satellite to:
   - Specify the user ID and the password that the satellite will use to connect to the satellite control database

     If synchronization credentials are not already stored at the satellite, the Connect to Control Database window opens. You must use this window to specify the user ID and password the satellite will use to connect to the satellite control database.
   - Verify the values that are set on the satellite are correct
   - Verify that the satellite can connect to the satellite control database

After you complete these configuration tasks, the Version 7.2 system is ready to synchronize.

Installing FixPak 2 or Higher on a Version 6 Enterprise Edition System

For a Version 6 Enterprise Edition system to be used as a DB2 control server, the system must be at FixPak 2 or higher.

The sections that follow describe the tasks that you must perform to upgrade a Version 6 Enterprise Edition system on Windows NT or AIX for use as a DB2 control server. If you are using a Version 6 Control Center, also perform the steps in "Upgrading a Version 6 Control Center and Satellite Administration Center" on page 132 to verify that you have the correct level of the Control Center and the Satellite Administration Center to administer the satellite environment.

Upgrading Version 6 DB2 Enterprise Edition for Use as the DB2 Control Server

For a Version 6 DB2 Enterprise Edition system to be used as the DB2 control server, it must be installed with the Control Server component, and DB2 Enterprise Edition should be at the FixPak 2 service level, or higher. Depending on whether the DB2 control server component is installed, and the service level of DB2 Enterprise Edition, you will have to perform one of the following tasks:

- Install the DB2 control server component to an existing DB2 Enterprise Edition V6.1 system and install FixPak 2 or higher. Then update the satellite control database (SATCTLDB) on the system.
- Upgrade an already installed DB2 control server to the FixPak 2 level or higher.

Use the information that follows to identify which of the two preceding tasks you need to perform, and the steps that apply to your situation. The following is a summary of the steps that you will perform.

1. First, assess the current state of your DB2 Enterprise Edition installation. You will determine whether the Control Server component is installed, and the service level of DB2.
2. Second, based on the state information that you obtain, you will determine what needs to be done.
3. Third, you will perform the necessary steps to upgrade DB2 Enterprise Edition.

The DB2 control server can only run on DB2 Enterprise Edition for Windows NT and AIX. Continue with the instructions that are appropriate for your platform:
- "Upgrading DB2 Enterprise Edition on Windows NT"
- "Upgrading DB2 Enterprise Edition on AIX" on page 128

Upgrading DB2 Enterprise Edition on Windows NT: Use the information in the sections that follow to determine the current service level of your Version 6 DB2 Enterprise Edition system, and the steps that you need to perform to update the system to the FixPak 2 service level or higher. You will need to perform the steps of one or more of the following sections:
- "Assessing DB2 Enterprise Edition on Windows NT"
- "Determining What Needs to Be Done" on page 125
- "Installing the Control Server Component on Windows NT" on page 126
- "Installing FixPak 2 or Higher on Windows NT" on page 127
- "Upgrading the SATCTLDB on Windows NT" on page 127

Assessing DB2 Enterprise Edition on Windows NT: If you have DB2 Enterprise Edition installed on Windows NT, perform the following steps:

1. Check whether the Control Server component is installed. Use the Registry Editor to display the list of installed components:
   a. Enter regedit at a command prompt.
   b. Under the HKEY_LOCAL_MACHINE\SOFTWARE\IBM\DB2\Components registry key, check whether the Control Server is listed. If it is not listed, the control server is not installed.

2. Determine the service level of DB2 Enterprise Edition. Issue the db2level command from a command prompt. Use the table that follows to interpret the output:

Values of Key Fields in the db2level output			Your DB2 system is at:
Release	Level	Informational Tokens	
SQL06010	01010104	db2_v6, n990616	Version 6.1 base
SQL06010	01020104	DB2 V6.1.0.1, n990824, WR21136	Version 6.1 plus FixPak 1
SQL06010	01030104	DB2 V6.1.0.6, s991030, WR21163 *or* DB2 V6.1.0.9, s000101, WR21173	Version 6.1 plus FixPak 2

Note: If the level is greater than 01030104, your system is at a higher FixPak than FixPak 2.

3. Record the information that you find, and continue at "Determining What Needs to Be Done".

Determining What Needs to Be Done: Using the information that you have gathered, find the row in the following table that applies to your situation, and follow the steps that are required to prepare your DB2 Enterprise Edition system to support the DB2 control server at the FixPak 2 level or higher.

Sections that follow the table provide instructions for performing the required steps. Consider checking off each step as you perform it. Only perform the steps that apply to your situation.

Control Server Component Installed	Service Level of DB2 Enterprise Edition System	Steps required to prepare your DB2 Enterprise Edition system
No	Version 6.1 base, or Version 6.1 plus FixPak 1, or Version 6.1 plus FixPak 2 or higher	Perform the following steps: 1. "Installing the Control Server Component on Windows NT" on page 126 2. "Installing FixPak 2 or Higher on Windows NT" on page 127 3. "Upgrading the SATCTLDB on Windows NT" on page 127
Yes	Version 6.1 base, or Version 6.1 plus FixPak 1	Perform the following steps: 1. "Installing FixPak 2 or Higher on Windows NT" on page 127 2. "Upgrading the SATCTLDB on Windows NT" on page 127

Control Server Component Installed	Service Level of DB2 Enterprise Edition System	Steps required to prepare your DB2 Enterprise Edition system
Yes	Version 6.1, plus FixPak 2 or higher	Perform the following step: 1. "Upgrading the SATCTLDB on Windows NT" on page 127

Installing the Control Server Component on Windows NT: To install the Control Server component on Windows NT:

1. Ensure that all database activity on the system is complete before proceeding.
2. Insert the DB2 Universal Database Enterprise Edition Version 6.1 CD in the CD drive.

   If the installation program does not start automatically, run the setup command in the root of the CD to start the installation process.
3. When prompted, shut down all the processes that are using DB2.
4. On the Welcome window, select **Next**.
5. On the Select Products window, ensure that DB2 Enterprise Edition is selected.
6. On the Select Installation Type panel, click **Custom**.
7. On the Select Components panel, ensure that the Control Server component is selected, and click **Next**.

   Note: If you select other components that are not already installed on your system, these components will be installed too. You cannot alter the drive or directory in which DB2 is installed.
8. On the Configure DB2 Services panels, you can modify the protocol values and the start-up options for the Control Server instance, or take the default values. Either modify the defaults and click **Next**, or click **Next** to use the defaults.
9. Click **Next** on the Start Copy files window to begin the installation process.
10. When the file copying process is complete, you have the option of rebooting your system. You should reboot now. The changes made to the system for the Control Server do not take effect until the system is rebooted.

When the installation process is complete and you have rebooted the system, the satellite control database (SATCTLDB) that was created as part of the Control Server installation must be cataloged in the DB2 instance if you want to use the Control Center and Satellite Administration Center locally on the system. To catalog the SATCTLDB database:

1. Open a DB2 Command Window by selecting **Start>Programs>DB2 for Windows NT>Command Window**
2. Ensure that you are in the db2 instance.

   Issue the set command and check the value of db2instance. If the value is not db2, issue the following command:

   ```
   set db2instance=db2
   ```
3. Catalog the db2ctlsv instance by entering the following command:

   ```
   db2 catalog local node db2ctlsv instance db2ctlsv
   ```
4. Catalog the SATCTLDB database by entering the following command

   ```
   db2 catalog database satctldb at node db2ctlsv
   ```
5. Commit the cataloging actions by entering the following command:

   ```
   db2 terminate
   ```
6. Close the DB2 Command Window.

Installing FixPak 2 or Higher on Windows NT: To upgrade an existing Version 6 DB2 Enterprise Edition system on Windows NT to FixPak 2 or higher, either:

- Download the latest FixPak for DB2 Enterprise Edition for Windows NT V6.1 from the Web, along with its accompanying readme. The FixPak can be downloaded by following the instructions at URL:

  ```
  http://www.ibm.com/software/data/db2/db2tech/version61.html
  ```

  Install the FixPak following the instructions in the readme.txt file.
- Use a DB2 Universal Database, Version 6.1 FixPak for Windows NT CD that is at FixPak 2 level or higher, and follow the instructions in the readme.txt file in the WINNT95 directory on the CD to complete the installation.

Upgrading the SATCTLDB on Windows NT: To upgrade the SATCTLDB database on Windows NT

1. Determine the level of the SATCTLDB database:
   a. Log on with a user ID that has local administrative authority on the Windows NT system.
   b. Open a DB2 Command Window by selecting **Start>Programs>DB2 for Windows NT>Command Window**.
   c. Connect to the SATCTLDB by entering the following command

      ```
      db2 connect to satctldb
      ```
   d. Determine if the trigger I_BATCHSTEP_TRGSCR exists in the database by issuing the following query:

      ```
      db2 select name from sysibm.systriggers
                        where name='I_BATCHSTEP_TRGSCR'
      ```

      Record the number of rows that are returned.
   e. Enter the following command to close the connection to the database:

```
db2 connect reset
```

If step 1d on page 127 returned one row, the database is at the correct level. In this situation, skip step 2, and continue at step 3. If zero (0) rows are returned, the database is not at the correct level, and must be upgraded, as described in step 2, before you can perform step 3.

2. To upgrade the SATCTLDB database, perform the following steps. Enter all commands in the DB2 Command Window:

   a. Switch to the directory <db2path>\misc, where <db2path> is the install drive and path, for example c:\sqllib.

   b. Ensure that you are in the db2ctlsv instance.

      Issue the set command and check the value of db2instance. If the value is not db2ctlsv, issue the following command:
      ```
      set db2instance=db2ctlsv
      ```

   c. Drop the SATCTLDB database by entering the following command:
      ```
      db2 drop database satctldb
      ```

   d. Create the new SATCTLDB database by entering the following command:
      ```
      db2 -tf satctldb.ddl  -z satctldb.log
      ```

   e. Issue the following command:
      ```
      db2 terminate
      ```

3. Bind the db2satcs.dll stored procedure to the SATCTLDB database. Perform the following steps:

   a. Connect to the SATCTLDB database by entering the following command
      ```
      db2 connect to satctldb
      ```

   b. Switch to the directory <db2path>\bnd, where <db2path> is the install drive and path, for example c:\sqllib.

   c. Issue the bind command, as follows:
      ```
      db2 bind db2satcs.bnd
      ```

4. Enter the following command to close the connection to the database:
   ```
   db2 connect reset
   ```

5. Close the DB2 Command Window.

Upgrading DB2 Enterprise Edition on AIX: Use the information in the sections that follow to determine the current service level of your Version 6 DB2 Enterprise Edition system, and the steps that you need to perform to update the system to the FixPak 2 service level, or higher. You will need to perform the steps of one or more of the following sections:

• "Assessing DB2 Enterprise Edition on AIX" on page 129

- "Determining What Needs to Be Done"
- "Installing the Control Server Component on AIX" on page 130
- "Installing FixPak 2 or Higher on AIX" on page 131
- "Upgrading the SATCTLDB Database on AIX" on page 131

Assessing DB2 Enterprise Edition on AIX: If you have Version 6 DB2 Enterprise Edition installed on AIX, perform the following steps:

1. Check whether the Control Server component is installed. Enter the following command:

   ```
   lslpp -l | grep db2_06_01.ctsr
   ```

   If no data is returned, the Control Server component is not installed.

2. Determine the service level of the DB2 Enterprise Edition. Log on as a DB2 instance owner, and issue the db2level command. Use the table that follows to interpret the output:

Values of Key Fields in the db2level output			Your DB2 system is at:
Release	Level	Informational Tokens	
SQL06010	01010104	db2_v6, n990616	Version 6.1 base
SQL06010	01020104	DB2 V6.1.0.1, n990824, U465423	Version 6.1 plus FixPak 1
SQL06010	01030104	DB2 V6.1.0.6, s991030, U468276 *or* DB2 V6.1.0.9, s000101, U469453	Version 6.1 plus FixPak 2

> **Note:** If the level is greater than 01030104, your system is at a higher FixPak than FixPak 2.

3. Record the information that you find, and continue at "Determining What Needs to Be Done".

Determining What Needs to Be Done: Using the information that you have gathered, find the row in the following table that applies to your situation, and follow the steps that are required to prepare your Version 6 DB2 Enterprise Edition system to support the DB2 control server at the FixPak 2 level.

Sections that follow the table provide instructions for performing the required steps. Consider checking off each step as you perform it. Only perform the steps that apply to your situation.

Control Server Component Installed	Service Level of DB2 Enterprise Edition System	Steps required to prepare your DB2 Enterprise Edition system
No	Version 6.1 base, or Version 6.1 plus FixPak 1, or Version 6.1 plus FixPak 2 or higher	Perform the following steps: 1. "Installing the Control Server Component on AIX" 2. "Installing FixPak 2 or Higher on AIX" on page 131 3. "Upgrading the SATCTLDB Database on AIX" on page 131
Yes	Version 6.1 base, or Version 6.1 plus FixPak 1	Perform the following steps: 1. "Installing FixPak 2 or Higher on AIX" on page 131 2. "Upgrading the SATCTLDB Database on AIX" on page 131
Yes	Version 6.1, plus FixPak 2 or higher	Perform the following step: 1. "Upgrading the SATCTLDB Database on AIX" on page 131

Installing the Control Server Component on AIX: To install the Control Server component on AIX

1. Log on as a user with root authority.
2. Insert the DB2 Universal Database Enterprise Edition Version 6.1 CD in the CD drive.
3. Change to the directory where the CD is mounted, for example, cd /cdrom.
4. Type the following command to start the DB2 installer:

   ./db2setup
5. When the DB2 Installer window opens, use the tab key to select the **Install** option, and press Enter.
6. Locate the Enterprise Edition line and use the tab key to select the **Customize** option beside it. Press Enter.
7. Select the DB2 Control Server component, tab to OK, and press Enter.
8. Follow the instructions on the remaining windows to complete the installation of the DB2 Control Server component.

When the installation process is complete, create the DB2CTLSV instance and the SATCTLDB database. To perform these tasks, follow the detailed instructions in "Setting up the DB2 Control Server on AIX" in Chapter 13 of the *Administering Satellites Guide and Reference*.

Installing FixPak 2 or Higher on AIX: To upgrade an existing DB2 Enterprise Edition system AIX to FixPak 2 or higher, either:

- Download the latest FixPak for DB2 Enterprise Edition for AIX V6.1 from the Web, along with its accompanying FixPak readme. The FixPak can be downloaded by following the instructions at URL:

      http://www.ibm.com/software/data/db2/db2tech/version61.html

  Install the FixPak following the instructions in the FixPak readme file.
- Use a DB2 Universal Database, Version 6.1 FixPak for AIX CD that is at FixPak 2 level or higher, and follow the instructions in the readme directory on the CD to complete the installation.

Ensure that you have updated the DB2CTLSV instance by running the `db2iupdt` command as instructed in the FixPak readme file.

Upgrading the SATCTLDB Database on AIX: To upgrade the SATCTLDB database on AIX:

1. Determine the level of the SATCTLDB database:
   a. Log in as `db2ctlsv`.
   b. Ensure that the database server has been started. If the server is not started, issue the `db2start` command.
   c. Connect to the SATCTLDB database by entering the following command:

          db2 connect to satctldb
   d. Determine if the trigger I_BATCHSTEP_TRGSCR exists in the database by issuing the following query:

          db2 "select name from sysibm.systriggers
                          where name='I_BATCHSTEP_TRGSCR'"

      Record the number of rows that are returned.
   e. Enter the following command to close the connection to the database:

          db2 connect reset

      If step 1d returned one row, the database is at the correct level. In this situation, skip step 2, and continue at step 3 on page 132. If zero (0) rows are returned, the database is not at the correct level, and must be upgraded, as described in step 2, before you can perform step 3 on page 132.
2. To upgrade the SATCTLDB database to the FixPak 2 level, perform the following steps. Enter all commands in the DB2 Command Window:
   a. Switch to the `$HOME/sqllib/misc` directory.
   b. Drop the SATCTLDB database by entering the following command:

          db2 drop database satctldb

c. Create the new SATCTLDB database by entering the following command:

```
db2 -tf satctldb.ddl  -z $HOME/satctldb.log
```

d. Issue the following command:

```
db2 terminate
```

3. Bind the db2satcs.dll stored procedure to the SATCTLDB database. Perform the following steps:

a. Connect to the SATCTLDB database by entering the following command

```
db2 connect to satctldb
```

b. Switch to the directory $HOME/sqllib/bnd.

c. Issue the bind command, as follows:

```
db2 bind db2satcs.bnd
```

4. Enter the following command to close the connection to the database:

```
db2 connect reset
```

Upgrading a Version 6 Control Center and Satellite Administration Center

To use a Version 6 Control Center and Satellite Administration Center with a Version 6 DB2 control server and satellite control database (SATCTLDB) that have been upgraded to FixPak 2 or higher, the tools must also be upgraded to FixPak 2 or higher.

If the Control Center and the Satellite Administration Center are running on the same system as the DB2 control server, they were upgraded when the DB2 Enterprise Edition system was upgraded to FixPak 2. However, if you run these tools on another system, you must upgrade this system to the FixPak 2 level or higher.

To upgrade this system to FixPak 2 or higher:

- Download the latest FixPak for your product at the V6.1 level from the Web, along with its accompanying readme. FixPaks can be downloaded by following the instructions at URL:

  ```
  http://www.ibm.com/software/data/db2/db2tech/version61.html
  ```

  Install the FixPak following the instructions in the readme file.

- Use a DB2 Universal Database, Version 6.1 FixPak CD for the operating system that you are running that is at FixPak 2 level or higher, and follow the instructions in the readme to complete the installation.

Command Reference

Update Available

The Command Reference was updated as part of FixPak 4. The latest PDF is available for download online at http://www.ibm.com/software/data/db2/udb/winos2unix/support. The information in these notes is in addition to the updated reference. All updated documentation is also available on CD. This CD can be ordered through DB2 service using the PTF number U478862. Information on contacting DB2 Service is available at http://www.ibm.com/cgi-bin/db2www/data/db2/udb/winos2unix/support/help.d2w/report.

db2updv7 - Update Database to Version 7 Current Fix Level

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

- Enables the use of the new built-in functions: ABS, DECRYPT_BIN, DECRYPT_CHAR, ENCRYPT, GETHINT, MULTIPLY_ALT, and ROUND.
- Enables the use of the new built-in functions for Unicode databases: DATE(vargraphic), TIME(vargraphic), TIMESTAMP(vargraphic), GRAPHIC(datetime-expression), GRAPHIC(date-expression), GRAPHIC(time-expression), and VARGRAPHIC(datetime-expression).
- Enables the use of the new built-in procedures (GET_ROUTINE_SAR and PUT_ROUTINE_SAR).
- Adds or applies corrections to WEEK_ISO and DAYOFWEEK_ISO functions on Windows and OS/2 databases.
- Applies a correction to table packed descriptors for tables migrated from Version 2 to Version 6.
- Creates the view SYSCAT.SEQUENCES.
- Creates the system objects needed in order to use a DB2 version 8 client to connect to a DB2 version 7 server.

Authorization
> *sysadm*

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

Command Syntax

```
      ►►──db2updv7──-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 FixPak, update the system catalog in the sample database by issuing the following command:

```
db2updv7 -d sample
```

Usage Notes
This tool can only be used on a database running DB2 Version 7.1 or Version 7.2 with at least FixPak 2 installed. If the command is issued more than once, no errors are reported and each of the catalog updates is applied only once.

To enable the new built-in functions, all applications must disconnect from this database and the database must be deactivated if it has been activated.

Additional Context for ARCHIVE LOG Usage Note

The usage notes for ARCHIVE LOG currently state that using this command will cause a database to lose a portion of its log sequence number (LSN) space, and thereby hasten the exhaustion of valid LSNs. To put this space usage into context, if you have a log file size of 100MB and run ARCHIVE LOG every five minutes, it will still take approximately 40 years to exhaust the valid LSNs. Under most operating conditions, you will not experience an impact.

REBIND

The syntax diagram for the REBIND command should appear as follows:

Command Syntax

```
►►──REBIND──────────────package-name──RESOLVE───┬──ANY──────────┬──────────────►◄
              └─PACKAGE─┘                        └─CONSERVATIVE──┘
```
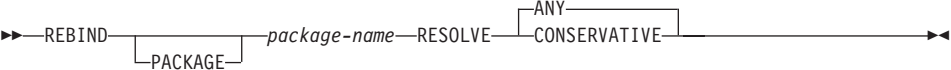
RUNSTATS

In the documentation for the RUNSTATS command, the last paragraph of the "Usage Notes" has incorrect information.

The last paragraph currently states what happens when inconsistencies are found when running the RUNSTATS command with and without distribution or index statistics. The statistics that are dropped or retained are not stated correctly. What follows are the correct statements about what happens.

If you issue RUNSTATS on a table, then previously collected distribution statistics are dropped. If you issue RUNSTATS on indexes only, then previously collected distribution statistics are retained.

db2inidb - Initialize a Mirrored Database

The description of the RELOCATE USING *configfile* parameter should appear as follows:

Specifies that the database files are to be relocated based on the information listed in the configuration file prior to initializing the database as a snapshot, standby or mirror.

Note: For information on the format of the configuration file, see the *Data Movement Utilities Guide and Reference*.

Usage Information

If the RELOCATE USING *configfile* parameter is specified and the database is relocated successfully, then the configuration file is copied into the database directory and renamed db2path.cfg. During any subsequent crash recoveries or rollfoward recoveries, this configuration file is used to dynamically rename the container paths during the log file processing.

If you initialize a snapshot or a mirror database, then the configuration file is removed automatically after the recovery completes. If you initialize a standby database, then the configuration file is not only removed after the recovery completes but is also removed if you cancel the recovery process.

If you are working with a standby database that you are keeping in the pending state so that you can continually roll it forward, and you add new containers to the original database, then you can manually update the db2path.cfg file to indicate where the containers should be stored for the standby database. If you do not specify a location for the new containers, then DB2 will attempt to store them in the same location as the originals.

db2relocatedb (new command)

db2relocatedb - Relocate Database

Renames a database, or relocates a database or part of a database (e.g., container, log directory) as specified in the configuration file provided by the user. This tool makes the necessary changes to the DB2 instance and database support files.

Authorization
None

Required Connection
None

Command Syntax

```
►►──db2relocatedb──-f──configFilename────────────────────────────────────────►◄
```

Command Parameters

-f configFilename

Specifies the name of the file containing configuration information necessary for relocating the database. This can be a relative or absolute filename. The format of the configuration file is:

```
DB_NAME=oldName,newName
DB_PATH=oldPath,newPath
INSTANCE=oldInst,newInst
NODENUM=nodeNumber
LOG_DIR=oldDirPath,newDirPath
CONT_PATH=oldContPath1,newContPath1
CONT_PATH=oldContPath2,newContPath2
...
```

Where:

DB_NAME

Specifies the name of the database being relocated. If the database name is being changed, both the old name and the new name must be specified. This is a required field.

DB_PATH

Specifies the path of the database being relocated. This is the path where the database was originally created. If the database path is changing, both the old path and new path must be specified. This is a required field.

INSTANCE

Specifies the instance where the database exists. If the database is being moved to a new instance, both the old instance and new instance must be specified. This is a required field.

NODENUM

Specifies the node number for the database node being changed. The default is 0.

LOG_DIR

Specifies a change in the location of the log path. If the log path is being changed, then both the old path and new path must be specified. This specification is optional if the log path resides under the database path, in which case the path is updated automatically.

CONT_PATH

Specifies a change in the location of table space containers. Both the old and new container path must be specified. Multiple CONT_PATH lines can be provided if there are multiple container path changes to be made. This specification is optional if the container paths reside under the database path, in which case the paths are updated automatically.

Note: Blank lines or lines beginning with a comment character (#) will be ignored.

Examples
Example 1

To change the name of the database TESTDB to PRODDB in the instance DB2INST1 that resides on the path /home/db2inst1, create the following configuration file:

```
DB_NAME=TESTDB,PRODDB
DB_PATH=/home/db2inst1
INSTANCE=db2inst1
NODENUM=0
```

Save the configuration file as relocate.cfg and use the following command to make the changes to the database files:

```
db2relocatedb -f relocate.cfg
```

Example 2

To move the database DATAB1 from the instance JSMITH on the path /dbpath to the instance PRODINST do the following:

1. Move the files in the directory /dbpath/jsmith to /dbpath/prodinst.
2. Use the following configuration file with the **db2relocatedb** command to make the changes to the database files:

```
DB_NAME=DATAB1
DB_PATH=/dbpath
INSTANCE=jsmith,prodinst
NODENUM=0
```

Example 3

The database PRODDB exists in the instance INST1 on the path
/databases/PRODDB. The location of two tablespace containers needs to be
changed as follows:
- SMS container /data/SMS1 needs to be moved to /DATA/NewSMS1.
- DMS container /data/DMS1 needs to be moved to /DATA/DMS1.

After the physical directories and files have been moved to the new locations,
the following configuration file can be used with the **db2relocatedb** command
to make changes to the database files so that they recognize the new locations:
```
DB_NAME=PRODDB
DB_PATH=/databases/PRODDB
INSTANCE=inst1
NODENUM=0
CONT_PATH=/data/SMS1,/DATA/NewSMS1
CONT_PATH=/data/DMS1,/DATA/DMS1
```

Example 4

The database TESTDB exists in the instance DB2INST1 and was created on the
path /databases/TESTDB. Table spaces were then created with the following
containers:
```
TS1
TS2_Cont0
TS2_Cont1
/databases/TESTDB/TS3_Cont0
/databases/TESTDB/TS4/Cont0
/Data/TS5_Cont0
/dev/rTS5_Cont1
```

TESTDB is to be moved to a new system. The instance on the new system will
be NEWINST and the location of the database will be /DB2.

When moving the database, all of the files that exist in the
/databases/TESTDB/db2inst1 directory must be moved to the /DB2/newinst
directory. This means that the first 5 containers will be relocated as part of this
move. (The first 3 are relative to the database directory and the next 2 are
relative to the database path.) Since these containers are located within the
database directory or database path, they do not need to be listed in the
configuration file. If the 2 remaining containers are to be moved to different
locations on the new system, they must be listed in the configuration file.

After the physical directories and files have been moved to their new locations, the following configuration file can be used with **db2relocatedb** to make changes to the database files so that they recognize the new locations:

```
DB_NAME=TESTDB
DB_PATH=/databases/TESTDB,./DB2
INSTANCE=db2inst1,newinst
NODENUM=0
CONT_PATH=/Data/TS5_Cont0,./DB2/TESTDB/TS5_Cont0
CONT_PATH=/dev/rTS5_Cont1,./dev/rTESTDB_TS5_Cont1
```

Example 5

The database TESTDB has 2 partitions on nodes 10 and 20. The instance is SERVINST and the database path is /home/servinst on both nodes. The name of the database is being changed to SERVDB and the database path is being changed to /databases on both nodes. In addition, the log directory is being changed on node 20 from /testdb_logdir to /servdb_logdir.

Since changes are being made to both nodes, a configuration file must be created for each node and **db2relocatedb** must be run on each node with the corresponding configuration file.

On node 10, the following configuration file will be used:

```
DB_NAME=TESTDB,SERVDB
DB_PATH=/home/servinst,./databases
INSTANCE=servinst
NODE_NUM=10
```

On node 20, the following configuration file will be used:

```
DB_NAME=TESTDB,SERVDB
DB_PATH=/home/servinst,./databases
INSTANCE=servinst
NODE_NUM=20
LOG_DIR=/testdb_logdir,./servdb_logdir
```

Usage Notes
If the instance that a database belongs to is changing, the following must be done before running this command to ensure that changes to the instance and database support files will be made:

- If a database is being moved to another instance, create the new instance.
- Copy the files/devices belonging to the databases being copied onto the system where the new instance resides. The path names must be changed as necessary.
- Change the permission of the files/devices that were copied so that they are owned by the instance owner.

If the instance is changing, the tool must be run by the new instance owner.

In a EEE environment, this tool must be run against every node that requires changes. A separate configuration file must be supplied for each node, that includes the NODENUM value of the node being changed. For example, if the name of a database is being changed, every node will be affected and the **db2relocatedb** command must be run with a separate configuration file on each node. If containers belonging to a single node are being moved, the **db2relocatedb** command only needs to be run once on that node.

See Also

For more information, see the **db2inidb - Initialize a Mirrored Database** command in the *Command Reference*.

db2move

The db2move tool now has two addtional options, **—aw** and **—sn**. Full
documentation for this tool follows:

Database Movement Tool

This tool facilitates the movement of large numbers of tables between DB2 databases located on workstations. The tool queries the system catalog tables for a particular database and compiles a list of all user tables. It then exports these tables in PC/IXF format. The PC/IXF files can be imported or loaded to another local DB2 database on the same system, or can be transferred to another workstation platform and imported or loaded to a DB2 database on that platform.

Note: Tables with structured type columns are not moved when this tool is used.

Authorization

This tool calls the DB2 export, import, and load APIs, depending on the action requested by the user. Therefore, the requesting user ID must have the correct authorization required by those APIs, or the request will fail.

Command Syntax

```
>>--db2move--dbname--action--+-----------------------+--><
                             | -tc table-creators    |
                             | -tn table-names       |
                             | -sn schema names      |
                             | -io import-option     |
                             | -lo load-option       |
                             | -l  lobpaths          |
                             | -u  userid            |
                             | -p  password          |
                             | -aw                   |
```
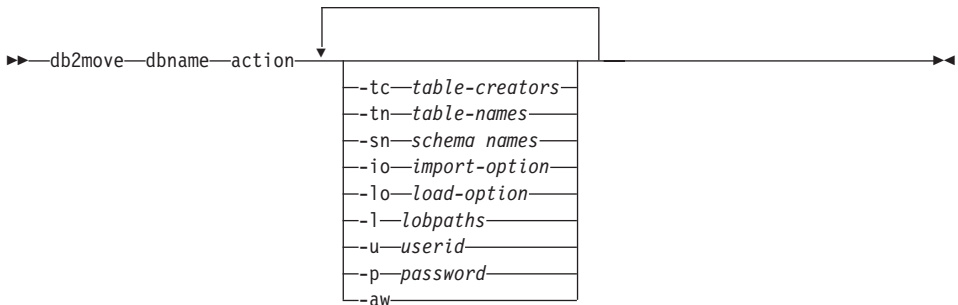
Command Parameters

dbname

Name of the database.

action Must be one of: EXPORT, IMPORT, or LOAD.

-tc table-creators. The default is all creators.

This is an EXPORT action only. If specified, only those tables created by the creators listed with this option are exported. If not specified, the default is to use all creators. When specifying multiple creators, each must be separated by commas; no blanks are allowed between creator IDs. The maximum number of creators that can be specified is 10. This option can be used with the "-tn" and "-sn" options to select the tables for export.

An asterisk (*) can be used as a wildcard character that can be placed anywhere in the string.

-tn   table-names. The default is all user tables.

This is an EXPORT action only. If specified, only those tables whose names match exactly those in the specified string are exported. If not specified, the default is to use all user tables. When specifying multiple table names, each must be separated by commas; no blanks are allowed between table names. The maximum number of table names that can be specified is 10. This option can be used with the "-tc" and "-sn" options to select the tables for export. **db2move** will only export those tables whose names are matched with specified table names and whose creators are matched with specified table creators.

An asterisk (*) can be used as a wildcard character that can be placed anywhere in the string.

-sn   schema names. The default is ALL SCHEMAS.

This is an EXPORT action only. If specified, only those tables whose schemas match exactly those in the specified string are exported. If not specified, the default is to use all schemas. When specifying multiple schema names, each must be separated by commas; no blanks are allowed between schema names. The maximum number of schema names that can be specified is 10. This option can be used with the "-tc" and "-tn" options to select the tables for export. **db2move** will only export those tables whose names are matched with specified table names, whose schemas are matched with specific table schemas, and whose creators are matched with specified table creators.

An asterisk (*) can be used as a wildcard character that can be placed anywhere in the string.

Note: Schema names less than 8 characters in length are padded to be 8 characters long. For example, if you want to include the schemas "AUSER" and "BUSER"and use the wildcard character, you must specify `-sn *USER*`.

-io   import-option. The default is REPLACE_CREATE.

Valid options are INSERT, INSERT_UPDATE, REPLACE, CREATE, and REPLACE_CREATE.

-lo   load-option. The default is INSERT.

Valid options are INSERT and REPLACE.

-l   lobpaths. The default is the current directory.

This option specifies the absolute path names where LOB files are created (as part of EXPORT) or searched for (as part of IMPORT or LOAD). When specifying multiple LOB paths, each must be separated by commas; no blanks are allowed between LOB paths. If the first path runs out of space (during EXPORT), or the files are not found in the path (during IMPORT or LOAD), the second path will be used, and so on.

If the action is EXPORT, and LOB paths are specified, all files in the LOB path directories are deleted, the directories are removed, and new directories are created. If not specified, the current directory is used for the LOB path.

-u   userid. The default is the logged on user ID.

Both user ID and password are optional. However, if one is specified, the other must be specified. If the command is run on a client connecting to a remote server, user ID and password should be specified.

-p   password. The default is the logged on password.

Both user ID and password are optional. However, if one is specified, the other must be specified. If the command is run on a client connecting to a remote server, user ID and password should be specified.

-aw   allow warnings.

Used for the EXPORT action only. If this option is specified, then any tables that receive warnings during export will be included in the db2move.lst file. If the option is omitted, then any tables that cause warnings during export are not included in the db2move.lst file. A table's .ixf file and .msg file are generated regardless of whether or not this option is used.

Examples

- db2move sample export

  This will export all tables in the SAMPLE database; default values are used for all options.

- db2move sample export -tc userid1,us*rid2 -tn tbname1,*tbname2

  This will export all tables created by "userid1" or user IDs LIKE "us%rid2", and with the name "tbname1" or table names LIKE "%tbname2".

- db2move sample import -l D:\LOBPATH1,C:\LOBPATH2

  This example is applicable to OS/2 or the Windows operating system only. The command will import all tables in the SAMPLE database; LOB paths "D:\LOBPATH1" and "C:\LOBPATH2" are to be searched for LOB files.

- db2move sample load -l /home/userid/lobpath,/tmp

This example is applicable to UNIX-based systems only. The command will load all tables in the SAMPLE database; both the /home/userid/lobpath subdirectory and the tmp subdirectory are to be searched for LOB files.

- `db2move sample import -io replace -u userid -p password`

This will import all tables in the SAMPLE database in REPLACE mode; the specified user ID and password will be used.

Usage Notes

This tool exports, imports, or loads user-created tables. If a database is to be duplicated from one operating system to another operating system, **db2move** facilitates the movement of the tables. It is also necessary to move all other objects associated with the tables, such as aliases, views, triggers, user-defined functions, and so on. **db2look** (DB2 Statistics and DDL Extraction Tool; see the *Command Reference*) can facilitate the movement of some of these objects by extracting the data definition language (DDL) statements from the database.

When export, import, or load APIs are called by **db2move**, the FileTypeMod parameter is set to lobsinfile. That is, LOB data is kept in separate files from PC/IXF files. There are 26 000 file names available for LOB files.

The LOAD action must be run locally on the machine where the database and the data file reside. When the load API is called by **db2move**, the CopyTargetList parameter is set to NULL; that is, no copying is done. If *logretain* is on, the load operation cannot be rolled forward later. The table space where the loaded tables reside is placed in backup pending state and is not accessible. A full database backup, or a table space backup, is required to take the table space out of backup pending state.

When issued on a Version 5.2 client against a Version 6 database, this tool does not support table or column names that are greater than 18 characters in length.

Files Required/Generated When Using EXPORT:

- Input: None.
- Output:

EXPORT.out	The summarized result of the EXPORT action.
db2move.lst	The list of original table names, their corresponding PC/IXF file names (tabnnn.ixf), and message file names (tabnnn.msg). This list, the exported PC/IXF files, and LOB files (tabnnnc.yyy) are used as input to the **db2move** IMPORT or LOAD action.
tabnnn.ixf	The exported PC/IXF file of a specific table.
tabnnn.msg	The export message file of the corresponding table.

| **tabnnnc.yyy** | The exported LOB files of a specific table.

"nnn" is the table number, "c" is a letter of the alphabet, "yyy" is a number ranging from 001 to 999.

These files are created only if the table being exported contains LOB data. If created, these LOB files are placed in the lobpath directories. There are a total of 26 000 possible names for the LOB files.

| **system.msg** | The message file containing system messages for creating or deleting file or directory commands. This is only used if the action is EXPORT and a LOB path is specified.

Files Required/Generated When Using IMPORT:

• Input:

| **db2move.lst** | An output file from the EXPORT action.

| **tabnnn.ixf** | An output file from the EXPORT action.

| **tabnnnc.yyy** | An output file from the EXPORT action.

• Output:

| **IMPORT.out** | The summarized result of the IMPORT action.

| **tabnnn.msg** | The import message file of the corresponding table.

Files Required/Generated When Using LOAD:

- Input:

  db2move.lst   An output file from the EXPORT action.

  tabnnn.ixf   An output file from the EXPORT action.

  tabnnnc.yyy   An output file from the EXPORT action.

- Output:

  LOAD.out   The summarized result of the LOAD action.

  tabnnn.msg   The LOAD message file of the corresponding table.

Additional Option in the GET ROUTINE Command

This command now supports the HIDE BODY parameter, which specifies that the body of the routine must be replaced by an empty body when the routine text is extracted from the catalogs.

This does not affect the compiled code; it only affects the text.

GET ROUTINE

Command Syntax

```
►►──GET ROUTINE──INTO──file_name──FROM──────────────┬──PROCEDURE──routine_name─────────────►
                                    └─SPECIFIC─┘

►──────────────────────────────────────────────────────────────────────►◄
   └─HIDE BODY─┘
```

CREATE DATABASE

DB2 now supports new collation sequence keywords, IDENTITY_16BIT and SQL_CS_IDENTITY_16BIT, for Unicode databases. When IDENTITY_16BIT is specified for the CLP CREATE DATABASE command or SQLEDBDESC.SQLDBCSS is set to SQL_CS_IDENTITY_16BIT in the sqlecrea() — Create Database API, all data in the Unicode database will be collated using the CESU-8 order. CESU-8 is *Compatibility Encoding Scheme for UTF-16: 8-Bit*, and as of this writing, its specification is contained in the Draft Unicode Technical Report #26 available at the Unicode Technical Consortium web site(www.unicode.org). CESU-8 is binary identical to UTF-8 except for the Unicode supplementary characters, that is, those characters that are defined outside the 16-bit Basic Multilingual Plane (BMP or Plane 0). In UTF-8 encoding, a supplementary character is represented by one 4-byte sequence, but the same character in CESU-8 requires two 3-byte sequences.

In a Unicode database, CHAR, VARCHAR, LONG VARCHAR, and CLOB data are stored in UTF-8, and GRAPHIC, VARGRAPHIC, LONG VARGRAPHIC, and DBCLOB data are stored in UCS-2. For IDENTITY or SQL_CS_NONE collation, non-supplementary characters in UTF-8 and UCS-2 have identical binary collation, but supplementary characters in UTF-8 collate differently from the same characters in UCS-2. IDENTITY_16BIT or SQL_CS_IDENTITY_16BIT ensures all characters, supplementary and non-supplementary, in a DB2 Unicode databases have the same binary collation.

Data Recovery and High Availability Guide and Reference

Data Recovery and High Availability Guide and Reference Available Online

The new Data Recovery and High Availability Guide and Reference is now available online in both HTML and PDF format at http://www.ibm.com/software/data/db2/udb/winos2unix/support. This information was previously contained in the Administration Guide. The information in these notes is in addition to the updated reference. All updated documentation is also available on CD. This CD can be ordered through DB2 service using the PTF number U478862. Information on contacting DB2 Service is available at http://www.ibm.com/cgi-bin/db2www/data/db2/udb/winos2unix/support/help.d2w/report.

New Archive Logging Behavior

Prior to FixPak 4, DB2 only checked for archive completion when a new log file was needed. Now DB2 checks for archive completion whenever the first active log changes. As a result, information is recorded to disk earlier and more often.

The benefit of this change is that if the system crashes, the information stored on disk (related to which log files are successfully archived) is more accurate and DB2 does not have to reissue the archive request for log files that are already archived.

There is no change to what DB2 does after detecting the successful archive of a particular log file.

DB2 now detects the completion of log archives earlier and will rename them earlier. Inactive truncated log files are deleted. As a result, the number of log files remaining in the active log path can be less than the LOGPRIMARY database configuration value. In this case, DB2 will create new log files when needed.

Before this change, restarting the database reduced the number of logs to equal the value of LOGPRIMARY. Now, when you restart a database, DB2 first examines the database log directory. If the number of empty logs is fewer than the number of primary logs, DB2 will allocate new logs to make up the difference. If more empty logs are available than there are primary logs in the database directory, DB2 will allow the database to be restarted with all the available empty logs in the database directory.

After database shutdown, any secondary log files in existence will remain in the active log path at restart time. To clear out the active log path, the DB2 ARCHIVE LOG command may be used.

How to Use Suspended I/O for Database Recovery

The information below about the db2inidb utility supersedes the information in the Version 7.2 What's New book.

db2inidb is a tool shipped with DB2 that can perform crash recovery or put a database in rollforward pending state.

Suspended I/O supports continuous system availability by providing a full implementation for online split mirror handling, that is, splitting a mirror without shutting down the database. If you cannot afford to do offline or online backups on a large database, you can do backups or system copies from a mirror image by using suspended I/O and a split mirror image.

Suspended I/O prevents disk writes while the split mirror image of a database is being taken. All database operations besides online backup and restore should function normally while a database is suspended. However, some operations may wait for I/O writes to resume if dirty pages must be flushed from the buffer pool or log buffers to the logs. These operations should resume normally once the database I/O is resumed. It is important that the database I/O be resumed from the same connection that it was originally suspended and that no other operations be performed from this connection until the database I/O resumes. Otherwise, subsequent connection attempts may hang if they require flushing dirty pages from the buffer pool to disk.

Subsequent connections will complete once database I/O resumes. If your connection attempts are hanging, and it has become impossible to resume the I/O from the connection that you used to suspend it, then you will have to run the RESTART command with the WRITE RESUME option. When used in this circumstance, the RESTART command will resume I/O writes without performing crash recovery. The RESTART command with the WRITE RESUME option will only perform crash recovery when you use it after a database crash.

In a partitioned database environment, you don't have to suspend I/O writes on all partitions simultaneously. You can suspend a subset of one or more partitions in order to create split mirrors to perform offline backups. If the catalog node is included in the subset, it must be the last partition to be suspended.

Mirroring a database primarily involves copying the entire contents of the database directory, and the local database directory. The local database directory, `sqldbdir`, is located at the same level of the file structure as the main database directory. In addition, if the log directory and table space containers are not in the database directory, then they must also be copied. Since the split mirrored database is dependent on these directory paths, the paths that these directories are copied to must be identical to those of the primary system. This means that the instance must also be the same. As a result of this dependency, it is not possible to create a mirror database on the same system as the primary database unless the new "relocate" option of the db2inidb tool is used.

The purpose of the "relocate" option is to relocate a database on a given system using a specified configuration file. This can involve changing the internal database directory, container directory, log directory, instance name and database names. Assuming the database directory, container directories and log directory were successfully mirrored to different directory paths on the same system as the primary database, the db2inidb tool can be used along with the "relocate" option to update the mirrored database's internal paths. A usage scenario with this option can be found below.

Depending on how the storage devices are being mirrored, the uses of db2inidb will vary. The following uses assume that the entire database is mirrored consistently through the storage system.

In a multinode environment, the db2inidb tool must be run on every partition before the split mirror can be used from any of the partitions. The db2inidb tool can be run on all partitions simultaneously by using the **db2_all** command.

1.

   Making a Clone Database:

   The objective here is to have a clone of the primary database to be used on another system. The following procedure describes how a clone database may be made:

   a. Suspend I/O writes on the primary database by entering the following command:

   ```
   db2 set write suspend for database
   ```

   b. Use operating system and disk subsystem level commands to split the mirror from the primary database. Ensure that you split both the data and the logs.

   c. Resume I/O writes on the primary database by entering the following command:

   ```
   db2 set write resume for database
   ```

After running the command, the primary database should be back to a normal state.

d. Mount the split mirror of the primary database on another system.

e. Start the database instance on the other system, by entering the following command:

```
db2start
```

f. Start the DB2 crash recovery by entering the following command:

```
db2inidb database_name AS  SNAPSHOT
```

> **Note:** This command will remove the suspend write state and roll back the changes made by transactions that were occurring at the time of the split.

You can also use this process to perform an offline backup, but if restored on the primary database, this backup cannot be used to roll forward, because the log chain will not match.

2.

Using the Split Mirror as a Standby Database:

As the mirrored (standby) database is continually rolling forward through the logs, new logs that are being created by the primary database are constantly fetched from the primary system. The following procedure describes how the split mirror can be used as a standby database:

a. Suspend I/O writes on the primary database:

```
db2 set write suspend for database
```

b. Use operating system and disk subsystem level commands to split the mirror from the primary database. Ensure that you only split the data and not the logs.

c. Resume the I/O writes on the primary database so that it goes back to normal processing.

```
db2 set write resume for database
```

d. Mount the split mirror of the database to another system.

e. Start the primary database instance by using the **db2start** command.

f. Place the mirror in roll forward pending:

```
db2inidb database_name AS STANDBY
```

> **Note:** This command will remove the suspend write state and place the mirrored database in rollforward pending state.

g. Copy logs by setting up a user exit program to retrieve log files from the primary system to ensure that the latest logs will be available for this mirrored database.

h. Roll forward the database to the end of the logs.

i. Go back to step g and repeat this process until the primary database is down.

j. Roll forward the database to the end of the logs, using the AND STOP option to bring the database back online. It will now be ready to use.

3.

Using the Split Mirror as a Backup Image:

The following procedure describes how to use the mirrored database as a backup image to restore over the primary database:

a. Stop the primary database instance with the **db2stop** command.

b. Use operating system and disk subsystem commands to copy the mirrored data back on top of the primary database. Do not copy back the log files. The logs on the primary database must be used for rollforward operations.

c. Start the primary database instance with the **db2start** command.

d. Run the following command to place the mirrored database in a rollforward pending state and to remove the suspend write state:

   db2inidb *database_name* AS MIRROR

e. Roll forward the database to the end of the logs, using the AND STOP option to bring the database back online. It will now be ready to use.

4.

Splitting a Mirror onto the Same System as the Primary Database:

The following procedure describes how to use the "relocate" option of the db2inidb tool to mirror a database onto the same system as the primary database. The example assumes that the database will be used under a new instance.

a. Create a new instance on the current system.

b. Suspend I/O writes on the primary database:

   db2 set write suspend for database

c. Use the operating system and disk subsystem level commands to split the mirror from the primary database.

> **Note:** The database directory, local database directory, container directories, and log directory must be copied to the new instance. If the container directories or the log directory exist under the database directory, then only the database directory and local database directory need to be copied.

d. Resume I/O writes on the primary database so that it goes back to normal processing:

   db2 set write resume for database

e. Create a configuration file with the following information:

DB_NAME=*name,optional_new_name*
DB_PATH=*primary_db_dir_path,mirrored_db_dir_path*
INSTANCE=*primary_instance,mirror_instance*
LOG_DIR=*primary_db_log_dir,mirrored_db_log_dir*
CONT_PATH=*primary_db_container_#1_path*,
        mirrored_db_container_#1_path ...
CONT_PATH=*primary_db_container_#n_path*,
        mirrored_db_container_#n_path
NODENUM=*node_#*

> **Note:** The LOG_DIR and the CONT_PATH fields are required only if
> the log directory and container directories exist outside of the
> database directory. All of the other fields are required, except
> for NODENUM, which will default to zero if not specified.

f. Start the database from the newly created instance:

   db2start

g. Relocate the mirrored database, remove the suspended state, and
   place the mirror in the rollforward pending state:

   db2inidb *database_name* as STANDBY relocate using config_file

h. Copy logs by setting up a user exit program to retrieve log files from
   the primary database to ensure that the latest logs will be available
   for this mirrored database.

i. Roll forward the database to the end of the logs.

j. Go back to step h and repeat this process until the primary database is
   down.

k. Roll forward the database to the end of the logs, using the AND
   STOP option to bring the database back online. It will now be ready
   to use.

New Backup and Restore Behavior When LOGRETAIN=CAPTURE

If a database is configured with LOGRETAIN set to CAPTURE, the following
operations cannot be performed:

- Online database backup
- Online or offline table space-level backup
- Online or offline table space-level restore

Following a database restore operation using an offline backup image taken
while LOGRETAIN is set to CAPTURE, the database is *not* put in rollforward
pending state. A database restore operation using an *online* database backup
image taken while LOGRETAIN is set to CAPTURE (Version 7.2 prior to
FixPak 4) is supported.

Incremental Backup and Recovery – Additional Information

During the second phase of processing, the database history is queried to build a chain of backup images required to perform the requested restore operation. If, for some reason, this is not possible, and DB2 is unable to build a complete chain of required images, the restore operation terminates, and an error message is returned. In this case, an automatic incremental restore will not be possible, and you will have issue the RESTORE DATABASE command with the INCREMENTAL ABORT option. This will cleanup any remaining resources so that you can proceed with a manual incremental restore.

During the third phase of processing, DB2 will restore each of the remaining backup images in the generated chain. If an error occurs during this phase, you will have to issue the RESTORE DATABASE command with the INCREMENTAL ABORT option to cleanup any remaining resources. You will then have to determine if the error can be resolved before you re-issue the RESTORE command or attempt the manual incremental restore again.

NEWLOGPATH2 Now Called DB2_NEWLOGPATH2

References to the NEWLOGPATH2 registry variable have been changed to DB2_NEWLOGPATH2.

Choosing a Backup Method for DB2 Data Links Manager on AIX or Solaris Operating Environment

Before setting the *PASSWORDACCESS* option in the Tivoli Storage Manager system options file, you must ensure that /usr/lib contains a symbolic link to the libApiDS.a library file.

Tivoli Storage Manager — LAN Free Data Transfer

DB2 Universal Database now allows users to use Tivoli's LAN Free Data Transfer technology for backups and restores to a TSM server. If you are using one of the following versions of DB2 Universal Database in conjunction with Tivoli's ADSM 3.1.x clients, you may experience problems when backing up or restoring to a TSM server:

- DB2 for AIX (32–bit)
- DB2 for Solaris Operating Environment (32–bit)
- DB2 for HP-UX (32–bit).

If you experience these problems, then carry out the following steps to correct them:

1. Issue a **db2stop** command.
2. Locate the sqllib/adsm directory on the DB2 UDB server.

3. Create a backup copy of libtadsm.a. Making a copy of it called libtadsm.a.bak is sufficient.
4. Copy libadsm.a to libtasdm.a.
5. Issue a **db2start** command.
6. Re-issue the failed backup or restore command.

Data Movement Utilities Guide and Reference

Extended Identity Values Now Fully Supported by Export Utility

The export utility now fully supports extended identity values. You will need both your client and server to be running with FixPak 7 or later in order to exploit this function.

Change to LOB File Handling by Export, Import, and Load

DB2 UDB now makes use of LOB location specifiers (LLSs) when importing, exporting, and loading large object (LOB) information. This allows multiple LOBs to be stored in a single file.

An LLS is a string indicating where LOB data can be found within a file. The format of the LLS is *filename.ext.nnn.mmm/*, where *filename.ext* is the name of the file that contains the LOB, *nnn* is the offset of the LOB within the file (measured in bytes), and mmm is the length of the LOB (in bytes). For example, an LLS of db2exp.001.123.456/ indicates that the lob is located in file db2exp.001, begins at an offset of 123 bytes into the file, and is 456 bytes long. If the indicated size in the LLS is 0, the LOB is considered to have a length of 0. If the length is -1, the LOB is considered to be NULL and the file name and offset do not matter.

When exporting data using the **lobsinfile** modifier, the LOBs will not always be placed into separate files. There may be multiple LOBs in each LOB file, and multiple LOB files per LOB path. The data file will now contain LLS records instead of just file names.

The import and load functions have also been changed to handle the changes to the export function. When loading or importing data with the **modified by lobsinfile** option specified, LLSs will be expected for each of the corresponding LOB columns. If something other than an LLS is encountered for a LOB column, the database will treat it as a LOB file, and will load the entire file as the LOB.

IXF Considerations

There are three new IXF data types. These three types correspond to character large objects (CLOBs), binary large objects (BLOBs), and double-byte character large objects (DBCLOBs) when represented by LLSs. The values of these data types are 964, 960, and 968 respectively.

IXF files now require each LOB column to have its own D record. This is
created automatically by the export tool, but must be created manually if you
are using a third party utility to create the IXF files. Additionally, an LLS is
required for each LOB in the table, and not just the non-null LOBs. If a LOB
column is null, you must write an LLS representing a null LOB.

Code Page Support for Import, Export and Load Utilities

The import, export and load utilities can now be used to transfer data from
the new Chinese code page GB 18030 (code page identifier 5488) and the new
Japanese code page ShiftJIS X0213 (code page identifier 1394) to DB2 UDB
Unicode databases. In addition, the export utility can be used to transfer data
from DB2 UDB Unicode databases to GB 18030 or ShiftJIS X0213 code page
data.

For example, the following command will load the Shift_JISX0213 data file
u/jp/user/x0213/data.del residing on a remotely connected client into
MYTABLE:

```
db2 load client from /u/jp/user/x0213/data.del
    of del modified by codepage=1394 insert into mytable
```

where MYTABLE is located on a DB2 UDB Unicode database.

Chapter 2. Import

Using Import with Buffered Inserts

The note at the end of this section should read:

Note: In all environments except EEE, the buffered inserts feature is disabled
during import operations in which the INSERT_UPDATE parameter is
specified.

Chapter 3. Load

Pending States After a Load Operation

The first two sentences in the last paragraph in this section have been
changed to the following:

The fourth possible state associated with the load process (check pending
state) pertains to referential and check constraints, DATALINKS constraints,
AST constraints, or generated column constraints. For example, if an existing
table is a parent table containing a primary key referenced by a foreign key in
a dependent table, replacing data in the parent table places both tables (not
the table space) in check pending state.

Load Restrictions and Limitations

The following restrictions apply to generated columns and the load utility:

- It is not possible to load a table having a generated column in a unique index unless the generated column is an "include column" of the index or the generatedoverride file type modifier is used. If this modifier is used, it is expected that all values for the column will be supplied in the input data file.
- It is not possible to load a table having a generated column in the partitioning key unless the generatedoverride file type modifier is used. If this modifier is used, it is expected that all values for the column will be supplied in the input data file.

totalfreespace File Type Modifier

The totalfreespace file type modifier (LOAD) has been modified to accept a value between 0 and 2 147 483 647.

Chapter 4. AutoLoader

AutoLoader Restrictions and Limitations

The following have been added to the restrictions and limitations for the AutoLoader utility:

1. AutoLoader must be executed on one of the server nodes.
2. If multiple instances exist, the AutoLoader can only be used against databases that are local to the instance specified by the DB2INSTANCE environment variable.

Using AutoLoader

The following has been added to the "Before Using AutoLoader" section:

Prior to invoking the AutoLoader utility ensure that rsh and/or rexec are functioning properly. Rexec is used to spawn remote processes if the password is specified in the AutoLoader configuration file. Otherwise, rsh is used.

rexecd Required to Run AutoLoader When Authentication Set to YES

In the AutoLoader Options section the following note will be added to the AUTHENTICATION and PASSWORD parameters description:

In a Linux environment, if you are running the AutoLoader with the authentication option set to YES, rexecd must be enabled on all machines. If rexecd is not enabled the following error message will be generated:

```
openbreeze.torolab.ibm.com: Connection refused
SQL6554N  An error occurred when attempting to remotely execute a process.
```

The following error messages will be generated in the db2diag.log file:

```
2000-10-11-13.04.16.832852   Instance:svtdbm   Node:000
PID:19612(db2atld)   Appid:
oper_system_services   sqloRemoteExec   Probe:31
```

AutoLoader May Hang During a Fork on AIX Systems Prior to 4.3.3

The AutoLoader is a multithreaded program and one of the threads forks off another process. Forking off a child process causes an image of the parent's memory to be created in the child.

On AIX systems prior to AIX 4.3.3, it is possible that locks used by libc.a to manage multiple threads allocating memory from the heap within the same process will be held by a non-forking thread. Since the non-forking thread will not exist in the child process, this lock will never be released in the child, causing the parent to sometimes hang.

AIX 4.3.3 contains a fix for a libc problem that could cause the AutoLoader to hang during a fork.

Appendix C. Export/Import/Load Utility File Formats

The following update has been added to this Appendix:

The export, import, and load utilities are not supported when they are used with a Unicode client connected to a non-Unicode database. Unicode client files are only supported when the Unicode client is connected to a Unicode database.

Replication Guide and Reference

Replication and Non-IBM Servers

You must use DataJoiner Version 2 or later to replicate data to or from non-IBM servers such as Informix, Microsoft SQL Server, Oracle, Sybase, and Sybase SQL Anywhere. You cannot use the relational connect function for this type of replication because DB2 Relational Connect Version 7 does not have update capability. Also, you must use DJRA (DataJoiner Replication Administration) to administer such heterogeneous replication on all platforms (AS/400, OS/2, OS/390, UNIX, and Windows) for all existing versions of DB2 and DataJoiner.

Replication on Windows 2000

DB2 DataPropagator Version 7 is compatible with the Windows 2000 operating system.

Known Error When Saving SQL Files

If you use the Control Center in DB2 Connect Personal Edition, you cannot save SQL files. If you try to save SQL files, you get an error message that the Database Administration Server (DAS) is not active, when in fact DAS is not available because it is not shipped with DB2 Connect PE.

Apply Program and Control Center Aliases

For the Apply program to work properly, the aliases used by the Apply program must match the aliases used by the Control Center.

DB2 Maintenance

It is recommended that you install the latest DB2 maintenance for the various DB2 products that you use in your replication environment.

Data Difference Utility on the Web

You can download the Data Difference utility (DDU) from the Web at ftp://ftp.software.ibm.com/ps/products/datapropagator/fixes/. The DDU is a sample utility that you can use to compare two versions of the same file and produce an output file that shows the differences. See the README file that accompanies the sample utility for details.

Chapter 3. Data Replication Scenario

Replication Scenarios

See the Library page of the DataPropagator Web site (http://www.ibm.com/software/data/dpropr/) for a new heterogeneous data replication scenario. Follow the steps in that scenario to copy changes from a replication-source table in an Oracle database on AIX to a target table in a database on DB2 for Windows NT. That scenario uses the DB2 DataJoiner Replication Administration (DJRA) tool, Capture triggers, the Apply program, and DB2 DataJoiner.

On page 44 of the book, the instructions in Step 6 for creating a password file should read as follows:

Step 6: Create a password file

Because the Apply program needs to connect to the source server, you must create a password file for user authentication. Make sure that the user ID that will run the Apply program can read the password file.

To create a password file:

1. From a Windows NT command prompt window, change to the `C:\scripts` directory.
2. Create a new file in this directory called `DEPTQUAL.PWD`. You can create this file using any text editor, such as Notepad. The naming convention for the password file is *applyqual*.pwd; where *applyqual* is a case-sensitive string that *must* match the case and value of the Apply qualifier used when you created the subscription set. For this scenario, the Apply qualifier is DEPTQUAL.

   Note: The filenaming convention from Version 5 of DB2 DataPropagator is also supported.
3. The contents of the password file has the following format:
   ```
   SERVER=server USER=userid PWD=password
   ```

   Where:

   server
   > The name of the source, target, or control server, exactly as it appears in the subscription set table. For this scenario, these names are SAMPLE and COPYDB.

   userid
   > The user ID that you plan to use to administer that particular database. This value is case-sensitive for Windows NT and UNIX operating systems.

password

> The password that is associated with that user ID. This value is case-sensitive for Windows NT and UNIX operating systems.

> Do not put blank lines or comment lines in this file. Add only the server-name, user ID, and password information.

4. The contents of the password file should look similar to:

```
SERVER=SAMPLE USER=subina PWD=subpw
SERVER=COPYDB USER=subina PWD=subpw
```

For more information about DB2 authentication and security, refer to the *IBM DB2 Administration Guide*.

Chapter 5. Planning for Replication

Table and Column Names

Replication does not support blanks in table and column names.

DATALINK Replication

DATALINK replication is available for the Solaris Operating Environment as part of Version 7.1 FixPak 1. It requires an FTP daemon that runs in the source and target DATALINK file system and supports the MDTM (modtime) command, which displays the last modification time of a given file. If you are using Version 2.6 of the Solaris Operating Environment, or any other version that does not include FTP support for MDTM, you need additional software such as WU-FTPD.

You cannot replicate DATALINK columns between DB2 databases on AS/400 and DB2 databases on other platforms.

On the AS/400 platform, there is no support for the replication of the "comment" attribute of DATALINK values.

If you are running AIX 4.2, before you run the default user exit program (ASNDLCOPY) you must install the PTF for APAR IY03101 (AIX 4210-06 RECOMMENDED MAINTENANCE FOR AIX 4.2.1). This PTF contains a Y2K fix for the "modtime/MDTM" command in the FTP daemon. To verify the fix, check the last modification time returned from the "modtime <file>" command, where <file> is a file that was modified after January 1, 2000.

If the target table is an external CCD table, DB2 DataPropagator calls the ASNDLCOPY routine to replicate DATALINK files. For the latest information about how to use the ASNDLCOPY and ASNDLCOPYD programs, see the prologue section of each program's source code. The following restrictions apply:

- Internal CCD tables can contain DATALINK indicators, but not DATALINK values.
- Condensed external CCD tables can contain DATALINK values.
- Noncondensed CCD target tables cannot contain any DATALINK columns.
- When the source and target servers are the same, the subscription set must not contain any members with DATALINK columns.

LOB Restrictions

Condensed internal CCD tables cannot contain references to LOB columns or LOB indicators.

Planning for Replication

On page 65, "Connectivity" should include the following fact:

```
If the Apply program cannot connect to the control server,
the Apply program terminates.
```

When using data blocking for AS/400, you must ensure that the total amount of data to be replicated during the interval does not exceed "4 million rows", not "4 MB" as stated on page 69 of the book.

Chapter 6. Setting up Your Replication Environment

Update-anywhere Prerequisite

If you want to set up update-anywhere replication with conflict detection and with more than 150 subscription set members in a subscription set, you must run the following DDL to create the ASN.IBMSNAP_COMPENSATE table on the control server:

```
CREATE TABLE ASN.IBMSNAP_COMPENSATE (
        APPLY_QUAL char(18) NOT NULL,
        MEMBER SMALLINT,
        INTENTSEQ CHAR(10) FOR BIT DATA,
        OPERATION CHAR(1));
```

Setting Up Your Replication Environment

Page 95, "Customizing CD table, index, and tablespace names" states that the DPREPL.DFT file is in either the \sqllib\bin directory or the \sqllib\java directory. This is incorrect, DPREPL.DFT is in the \sqllib\cc directory.

On page 128, the retention limit description should state that the retention limit is used to prune rows only when Capture warm starts or when you use the Capture prune command. If you started Capture with the auto-pruning option, it will not use the retention limit to prune rows.

Chapter 8. Problem Determination

The Replication Analyzer runs on Windows 32-bit systems and AIX. To run the Analyzer on AIX, ensure that the `sqllib/bin` directory appears before `/usr/local/bin` in your PATH environment variable to avoid conflicts with `/usr/local/bin/analyze`.

The Replication Analyzer has two additional optional keywords: CT and AT.

CT=*n* Show only those entries from the Capture trace table that are newer than *n* days old. This keyword is optional. If you do not specify this keyword, the default is 7 days.

AT=*n* Show only those entries from the Apply trail table that are newer than *n* days old. This keyword is optional. If you do not specify this keyword, the default is 7 days.

Example:

```
analyze mydb1 mydb2 f=mydirectory ct=4 at=2 deepcheck q=applyqual1
```

For the Replication Analyzer, the following keyword information is updated:

deepcheck
> Specifies that the Analyzer perform a more complete analysis, including the following information: CD and UOW table pruning information, DB2 for OS/390 tablespace-partitioning and compression detail, analysis of target indexes with respect to subscription keys, subscription timelines, and subscription-set SQL-statement errors. The analysis includes all servers. This keyword is optional.

lightcheck
> Specifies that the following information be excluded from the report: all column detail from the ASN.IBMSNAP_SUBS_COLS table, subscription errors or anomalies or omissions, and incorrect or inefficient indexes. This reduction in information saves resources and produces a smaller HTML output file. This keyword is optional and is mutually exclusive with the deepcheck keyword.

Analyzer tools are available in PTFs for replication on AS/400 platforms. These tools collect information about your replication environment and produce an HTML file that can be sent to your IBM Service Representative to aid in problem determination. To get the AS/400 tools, download the appropriate PTF (for example, for product 5769DP2, you must download PTF SF61798 or its latest replacement).

Add the following problem and solution to the "Troubleshooting" section:

Problem: The Apply program loops without replicating changes; the Apply trail table shows STATUS=2.

The subscription set includes multiple source tables. To improve the handling of hotspots for one source table in the set, an internal CCD table is defined for that source table, but in a different subscription set. Updates are made to the source table but the Apply process that populates the internal CCD table runs asynchronously (for example, the Apply program might not be started or an event not triggered, and so on). The Apply program that replicates updates from the source table to the target table loops because it is waiting for the internal CCD table to be updated.

To stop the looping, start the Apply program (or trigger the event that causes replication) for the internal CCD table. The Apply program will populate the internal CCD table and allow the looping Apply program to process changes from all source tables.

A similar situation could occur for a subscription set that contains source tables with internal CCD tables that are populated by multiple Apply programs.

Chapter 9. Capture and Apply for AS/400

On page 178, "A note on work management" should read as follows:

```
You can alter the default definitions or provide your own definitions.
If you create your own subsystem description, you must name the
subsystem QZSNDPR and create it in a library other than QDPR.
See "OS/400 Work Management V4R3", SC41-5306 for more information
about changing these definitions.
```

Add the following to page 178, "Verifying and customizing your installation of DB2 DataPropagator for AS/400":

If you have problems with lock contention due to high volume of transactions, you can increase the default wait timeout value from 30 to 120. You can change the job every time the Capture job starts or you can use the following procedure to change the default wait timeout value for all jobs running in your subsystem:

1. Issue the following command to create a new class object by duplicating QGPL/QBATCH:

```
CRTDUPOBJ OBJ(QBATCH) FROMLIB(QGPL) OBJTYPE(*CLS)
          TOLIB(QDPR) NEWOBJ(QZSNDPR
```

2. Change the wait timeout value for the newly created class (for example, to 300 seconds):

```
CHGCLS CLS(QDPR/QZSNDPR) DFTWAIT(300)
```

3. Update the routing entry in subsystem description QDPR/QZSNDPR to use the newly created class:

```
CHGRTGE SBSD(QDPR/QZSNDPR) SEQNBR(9999) CLS(QDPR/QZSNDPR
```

On page 194, "Using the delete journal receiver exit routine" should include this sentence: If you remove the registration for the delete journal receiver exit routine, make sure that all the journals used for source tables have DLTRCV(*NO).

On page 195, the ADDEXITPGM command parameters should read:

```
ADDEXITPGM EXITPNT(QIBM_QJO_DLT_JRNRCV)
           FORMAT(DRCV0100)
           PGM(QDPR/QZSNDREP)
           PGMNBR(*LOW)
           CRTEXITPNT(*NO)
           PGMDTA(65535 10 QSYS)
```

Chapter 10. Capture and Apply for OS/390

In Chapter 10, the following paragraphs are updated:

Prerequisites for DB2 DataPropagator for OS/390

You must have DB2 for OS/390 Version 5, DB2 for OS/390 Version 6, or DB2 for OS/390 Version 7 to run DB2 DataPropagator for OS/390 Version 7 (V7).

UNICODE and ASCII Encoding Schemes on OS/390

DB2 DataPropagator for OS/390 V7 supports UNICODE and ASCII encoding schemes. To exploit the new encoding schemes, you must have DB2 for OS/390 V7 and you must manually create or convert your DB2 DataPropagator source, target, and control tables as described in the following sections. However, your existing replication environment will work with DB2 DataPropagator for OS/390 V7 even if you do not modify any encoding schemes.

Choosing an Encoding Scheme

If your source, CD, and target tables use the same encoding scheme, you can minimize the need for data conversions in your replication environment. When you choose encoding schemes for the tables, follow the single CCSID rule: Character data in a table space can be encoded in ASCII, UNICODE, or EBCDIC. All tables within a table space must use the same encoding scheme. The encoding scheme of all the tables in an SQL statement must be the same. Also, all tables that you use in views and joins must use the same encoding scheme.

If you do not follow the single CCSID rule, DB2 will detect the violation and return SQLCODE -873 during bind or execution. Which tables should be

ASCII or UNICODE depends on your client/server configuration. Specifically, follow these rules when you choose encoding schemes for the tables:

- Source or target tables on DB2 for OS/390 can be EBCDIC, ASCII, or UNICODE. They can be copied from or to tables that have the same or different encoding scheme in any supported DBMS (DB2 family, or non-DB2 with DataJoiner).
- On a DB2 for OS/390 source server, all CD, UOW, register, and prune control tables on the same server must use the same encoding scheme. To ensure this consistency, always specify the encoding scheme explicitly.
- All the control tables (ASN.IBMSNAP_SUBS_xxxx) on the same control server must use the same encoding scheme.
- Other control tables can use any encoding scheme; however, it is recommended that the ASN.IBMSNAP_CRITSEC table remain EBCDIC.

Setting Encoding Schemes
To specify the proper encoding scheme for tables, modify the SQL that is used to generate the tables:

- Create new source and target tables with the proper encoding scheme, or change the encoding schemes of the existing target and source tables. It is recommended that you stop the Capture and Apply programs before you change the encoding scheme of existing tables, and afterwards that you cold start the Capture program and restart the Apply program. To change the encoding scheme of existing tables:
  1. Use the Reorg utility to copy the existing table.
  2. Drop the existing table.
  3. Re-create the table specifying the new encoding scheme.
  4. Use the Load utility to load the old data into the new table.

  See the *DB2 Universal Database for OS/390 Utility Guide and Reference* for more information on the Load and Reorg utilities.

- Create new control tables with the proper encoding scheme or modify the encoding scheme for existing ones.

  DPCNTL.MVS is shipped with DB2 for OS/390 in sqllib\samples\repl and it contains several CREATE TABLE statements that create the control tables. For those tables that need to be ASCII or UNICODE (for example, ASN.IBMSNAP_REGISTER and ASN.IBMSNAP_PRUNCNTL), add the CCSID ASCII or CCSID UNICODE keyword, as shown in the following example.

```
CREATE TABLE ASN.IBMSNAP_PRUNCNTL (
  TARGET_SERVER    CHAR( 18)              NOT NULL,
  TARGET_OWNER     CHAR( 18)              NOT NULL,
  TARGET_TABLE     CHAR( 18)              NOT NULL,
  SYNCHTIME        TIMESTAMP,
  SYNCHPOINT       CHAR( 10)              FOR BIT DATA,
  SOURCE_OWNER     CHAR( 18)              NOT NULL,
```

```
        SOURCE_TABLE        CHAR( 18)               NOT NULL,
        SOURCE_VIEW_QUAL    SMALLINT                NOT NULL,
        APPLY_QUAL          CHAR( 18)               NOT NULL,
        SET_NAME            CHAR( 18)               NOT NULL,
        CNTL_SERVER         CHAR( 18)               NOT NULL,
        TARGET_STRUCTURE    SMALLINT                NOT NULL,
        CNTL_ALIAS          CHAR(  8)
            ) CCSID UNICODE
                DATA CAPTURE CHANGES
                IN TSSNAP02;
```

To modify existing control tables and CD tables, use the Reorg and Load utilities.

- When you create new replication sources or subscription sets, modify the SQL file generated by the administration tool to specify the proper encoding scheme. The SQL has several CREATE TABLE statements that are used to create the CD and target tables for the replication source and subscription set, respectively. Add the keyword CCSID ASCII or CCSID UNICODE where appropriate. For example:

```
CREATE TABLE   user1.cdtable1 (
  employee_name   varchar,
  employee_age    decimal
    ) CCSID UNICODE;
```

The *DB2 UDB for OS/390 SQL Reference* contains more information about CCSID.

Chapter 11. Capture and Apply for UNIX platforms

Setting Environment Variables for Capture and Apply on UNIX and Windows

If you created the source database with a code page other than the default code page value, set the DB2CODEPAGE environment variable to that code page. See the *DB2 Administration Guide* for information about deriving code page values before you set DB2CODEPAGE. Capture must be run in the same code page as the database for which it is capturing data. DB2 derives the Capture code page from the active environment where Capture is running. If DB2CODEPAGE is not set, DB2 derives the code page value from the operating system. The value derived from the operating system is correct for Capture if you used the default code page when creating the database.

Chapter 14. Table Structures

On page 339, append the following sentence to the STATUS column description for the value ″2″:

If you use internal CCD tables and you repeatedly get a value of "2" in the status column of the Apply trail table, go to "Chapter 8: Problem Determination" and refer to "Problem: The Apply program loops without replicating changes, the Apply trail table shows STATUS=2".

Chapter 15. Capture and Apply Messages

Message ASN0017E should read:

ASN0017E

The Capture program encountered a severe internal error and could not issue the correct error message. The routine name is "routine". The return code is "return_code".

Message ASN1027S should be added:

ASN1027S

There are too many large object (LOB) columns specified. The error code is "<error_code>".

Explanation: Too many large object (BLOB, CLOB, or DBCLOB) columns are specified for a subscription set member. The maximum number of columns allowed is 10.

User response: Remove the excess large object columns from the subscription set member.

Message ASN1048E should read as follows:

ASN1048E

The execution of an Apply cycle failed. See the Apply trail table for full details: "<text>"

Explanation: An Apply cycle failed. In the message, "<text>" identifies the "<target_server>", "<target_owner, target_table, stmt_number>", and "<cntl_server>".

User response: Check the APPERRM fields in the audit trail table to determine why the Apply cycle failed.

Appendix A. Starting the Capture and Apply Programs from Within an Application

On page 399 of the book, a few errors appear in the comments of the Sample routine that starts the Capture and Apply programs; however the code in the sample is correct. The latter part of the sample pertains to the Apply parameters, despite the fact that the comments indicate that it pertains to the Capture parameters.

You can get samples of the Apply and Capture API, and their respective makefiles, in the following directories:

```
For NT - sqllib\samples\repl
For UNIX - sqllib/samples/repl
```

System Monitor Guide and Reference

db2ConvMonStream

In the Usage Notes, the structure for the snapshot variable datastream type SQLM_ELM_SUBSECTION should be sqlm_subsection.

Maximum Database Heap Allocated (db_heap_top)

The Maximum Database Heap Allocated data element is not collected by the DB2 Version 7 database manager.

Troubleshooting Guide

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

For a db2start command to be successful in a Windows 95, Windows 98, or Windows Millennium Edition (ME) environment, you must either:

- Log on using the Windows logon window or the Microsoft Networking logon window
- Issue the **db2logon** command (see note 1 on page 180 for information about the **db2logon** command).

In addition, the user ID that is specified either during the logon or for the **db2logon** command must meet DB2's requirements (see note 2 on page 181).

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

During the installation of DB2 Universal Database Version 7 on Windows 95, Windows 98, and Windows ME, the installation software, by default, adds a shortcut to the Startup folder that runs the **db2start** command when the system is booted (see note 1 on page 180 for more information). If the user of the system has neither logged on nor issued the **db2logon** command, the **db2start** command will terminate.

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

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

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

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

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

```
db2set -g db2path
```

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

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

3. To start DB2 when the system is booted, you should run the batch file from a shortcut in the Startup folder. You have two options:

- Modify the shortcut that is created by the DB2 installation program to run the batch file instead of **db2start.exe**. In the preceding example, the shortcut would now run the db2start.bat batch file. The shortcut that is created by DB2 installation program is called **DB2 - DB2.lnk**, and is located in c:\WINDOWS\Start Menu\Programs\Start\DB2 - DB2.lnk on most systems.

- Add your own shortcut to run the batch file, and delete the shortcut that is added by the DB2 installation program. Use the following command to delete the DB2 shortcut:

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

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

> **Note:** As an alternative to starting DB2 during the boot of the system, DB2 can be started prior to the running of any application that uses DB2. See note 5 on page 182 for details.

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

Notes:

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

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

The user ID that is specified for the command must meet the DB2 naming requirements (see note 2 on page 181 for more information). If the command is issued without a user ID and password, a window opens to

prompt the user for the user ID and password. If the only parameter provided is a user ID, the user is not prompted for a password; under certain conditions a password is required, as described below.

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

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

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

    The table is created with a high-level qualifier as db2local.tab1.

  You should use a user ID that is equal to the schema name of your tables and other objects.
- When the system acts as client to a server, and the user issues a CONNECT statement without a user ID and password (for example, CONNECT TO TEST) and authentication is set to server, the user ID and password from the **db2logon** command are used to validate the user at the remote server. If the user connects with an explicit user ID and password (for example, CONNECT TO TEST USER *userID* USING *password*), the values that are specified for the CONNECT statement are used.

2. In Version 7, the user ID that is either used to log on or specified for the **db2logon** command must conform to the following DB2 requirements:
   - It cannot be any of the following: USERS, ADMINS, GUESTS, PUBLIC, LOCAL, or any SQL reserved word that is listed in the *SQL Reference*.
   - It cannot begin with: SQL, SYS or IBM
   - Characters can include:
     – A through Z (Windows 95, Windows 98, and Windows ME support case-sensitive user IDs)
     – 0 through 9
     – @, #, or $

3. You can prevent the creation of the **db2start** shortcut in the Startup folder during a customized interactive installation, or if you are performing a response file installation and specify the DB2.AUTOSTART=NO option. If you use these options, there is no db2start shortcut in the Startup folder, and you must add your own shortcut to run the db2start.bat file.

4. On **Windows 98** and **Windows ME** an option is available that you can use to specify a user ID that is always logged on when Windows 98 or Windows ME is started. In this situation, the Windows logon window will not appear. If you use this option, a user is logged on and the **db2start** command will succeed if the user ID meets DB2 requirements (see note 2 on page 181 for details). If you do not use this option, the user will always be presented with a logon window. If the user cancels out of this window without logging on, the **db2start** command will fail unless the **db2logon** command was previously issued, or invoked from the batch file, as described above.

5. If you do not start DB2 during a system boot, DB2 can be started by an application. You can run the db2start.bat file as part of the initialization of applications that use DB2. Using this method, DB2 will only be started when the application that will use it is started. When the user exits the application, a **db2stop** command can be issued to stop DB2. Your business applications can start DB2 in this way, if DB2 is not started during the system boot.

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

   If you are calling either the **db2sync** command or the synchronization APIs from your application, you would start DB2 during the initialization of your application. If your users will be using the DB2 Synchronizer shortcut in the DB2 for Windows folder to start synchronization, the DB2 Synchronization shortcut must be modified to run a db2sync.bat file. The batch file should contain the following commands to ensure that DB2 is running before synchronization begins:

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

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

If you decide to start DB2 when the application starts, ensure that the
installation of DB2 does not add a shortcut to the Startup folder to start
DB2. See note 3 on page 182 for details.

Chapter 1. Good Troubleshooting Practices

Problem Analysis and Environment Collection Tool

There is a utility that will help you identify some of the information
associated with your problem and will collect other relevant information to
assist DB2 Customer Support to understand your environment and your
problem. Much of what is collected using this utility is discussed in the rest of
this chapter. The utility is **db2support**.

Details about the syntax and command line options is found in the *Command
Reference*.

The purpose of the utility is to collect environmental data about your client or
server machine that is running DB2; and then to collect and package a large
portion of the output as browsable XML, HTML, or a compressed file archive.
The utility also has an option that allows for the collection of some data from
you about the nature of your problem using an interactive question and
answer process. This process will help you clarify the problem and also
provide information to DB2 Customer Support when you finally contact them
regarding your problem.

Note: A thin or runtime client is not able to use this utility. The utility
requires that the client have the DB2 engine libraries installed.

Collection Outputs
The utility produces a compressed collection (single file archive) of important
database system information. Included in this archive is an HTML report of
the most essential information, which you can use to view the information.

By default, db2support will not collect table data, schema (DDL), or logs in
order to protect the security and sensitivity of customer data. With some
options, the user may elect to include aspects of their schema and data (such
as including archived logs). Options that expose database schema or data
should be used carefully. When db2support is invoked, a message indicating
how sensitive data is dealt with will be displayed.

The following are the files to be collected and compressed into a single
archive:

Collected under all conditions

1. db2diag.log
2. All trap files
3. Lock list files (with -d)
4. Dump files
5. User exit (with -d)
6. Buffer pool and table space (SPCS) control files (with -d)
7. Various system related files
8. Output from various system commands
9. db config (with -d)
10. dbm config files
11. Log File Header file (with -d)
12. Recovery History File
13. db2cli.ini

Optionally collected

1. Active log files
2. Contents of db2dump directory (i.e. what was not collected above)
3. Core files (-a for all core file, -r for only the most recent core file)
4. Extended system information (-s)

The following files make up the content of the HTML report:

Collected under all conditions

1. PMR number, if one exists. (if -n was specified)
2. Operating system and level. (e.g. AIX 4.2.1)
3. DB2 release information.
4. Engine library header information.
5. Detecting 32- or 64-bit
6. DB2 install path information.
7. For EEE report contents of db2nodes.cfg
8. How many CPUs, disks, and how much memory.
9. List of databases on this instance.
10. Registry information, environment, including path & libpath.
11. Disk freespace for current filesystem and inodes for Unix.
12. JDK level.
13. dbm config.
14. Listing of the database recovery history file.

15. 'ls -lR' (or windows equivilant) of the sqllib directory.
16. LIST NODE DIRECTORY
17. LIST ADMIN NODE DIRECTORY
18. LIST DCS DIRECTORY
19. LIST DCS APPLICATIONS EXTENDED
20. List of all installed software.

Collected if '-s' is specified
1. Detailed disk information (partition layout, type, LVM information, etc.)
2. Detailed network information
3. Kernel statistics
4. Firmware versions
5. Other platform specific commands

Collected if DB2 has been started
1. Client connection state
2. db/dbm config (db cfg require -d option)
3. CLI config
4. Memory pool info (size and consumed). Complete data if -d option used.
5. LIST ACTIVE DATABASES
6. LIST DATALINKS MANAGERS
7. LIST DCS APPLICATIONS

Collected if -c has been specified and a connection to the database can be made
1. Number of user tables
2. Approximate size of DB data
3. Database snapshot
4. Application snapshot
5. Buffer pool information
6. LIST APPLICATIONS
7. LIST COMMAND OPTIONS
8. LIST DATABASE DIRECTORY
9. LIST INDOUBT TRANSACTIONS
10. LIST NODEGROUPS
11. LIST NODES
12. LIST ODBC DATA SOURCES
13. LIST PACKAGES/TABLES

14. LIST TABLESPACE CONTAINERS

15. LIST TABLESPACES

16. LIST DRDA IN DOUBT TRANSACTIONS

If '-q' is specified, collect the following

The interactive question and answer mode is started. With the exception of an optional "describe your problem" question and a small number of requests for customer information, all of the questions will have multiple choice answers from which to select. All of the questions, including follow up questions, and the answers will be collected. In some cases, the utility will ask you to carry out a task and place the results of that task in an additional directory. A small decision tree is used during the interactive mode to determine the questions to ask. These interactive questions assist in determining the category of the problem and based on the category a few other relevant questions may be asked and additional data collected. At the end of the questions, any data that would have been collected in the automatic mode, will also be collected. The answers to all questions are stored in preparation to be sent to service along with any data collected in automatic mode.

Viewing detailed_system_info.html
If you are running db2support on a non-English installation and are experiencing difficulties properly viewing detailed_system_info.html, you may need to use Internet Explorer version 5 or later with DOS encoding. To change the encoding, select View —> Encoding —> Central European (DOS). If you do not already have the required encoding support, then Internet Explorer prompts you to download the required files from the Microsoft Updates web site. This information does not apply to double-byte languages (Simplified Chinese, Traditional Chinese, Japanese and Korean).

Viewing DB2 Support Tool Syntax One Page at a Time
To view the syntax for the DB2 Support Tool one page at a time, run the following command:

```
db2support | more
```

Chapter 2. Troubleshooting the DB2 Universal Database Server

Under the "Locking and Deadlocks" section, under the "Applications Slow or Appear to Hang" subsection, change the description under "Lock waits or deadlocks are not caused by next key locking" to :

Next key locking guarantees Repeatable Read (RR) isolation level by automatically locking the next key for all INSERT and DELETE statements and the next higher key value above the result set for SELECT statements. For UPDATE statements that alter key parts of an index, the original index key is deleted and the new key value is inserted. Next key locking is done on both

the key insertion and key deletion. It is required to guarantee ANSI and SQL92 standard RR, and is the DB2 default.

Examine snapshot information for the application. If the problem appears to be with next key locking, you can set the DB2_RR_TO_RS option on if none of your applications rely on Repeatable Read (RR) behavior and it is acceptable for scans to skip over uncommitted deletes.

When DB2_RR_TO_RS is on, RR behavior cannot be guaranteed for scans on user tables because next key locking is not done during index key insertion and deletion. Catalog tables are not affected by this option.

The other change in behavior is that with DB2_RR_TO_RS on, scans will skip over rows that have been deleted but not committed, even though the row may have qualified for the scan.

For example, consider the scenario where transaction A deletes the row with column1=10 and transaction B does a scan where column1>8 and column1<12.

With DB2_RR_TO_RS off, transaction B will wait for transaction A to commit or rollback. If it rolls back, the row with column1=10 will be included in the result set of transaction B's query.

With DB2_RR_TO_RS on, transaction B will not wait for transaction A to commit or rollback. It will immediately receive query results that do not include the deleted row.

Do not use this option if you require ANSI and SQL92 standard RR or if you do not want scans to skip uncommitted deletes.

Chapter 8. Troubleshooting DB2 Data Links Manager

In Version 7 FixPak 2, an SQL1179W warning message is generated by the server when precompiling a source file or binding a bind file without specifying a value for the FEDERATED option. The same message is generated when the source file or bind file includes static SQL references to a nickname. There are two exceptions:

- For clients that are at an earlier FixPak than Version 7 FixPak 2 or for downlevel clients, the sqlaprep() API does not report this SQL1179W warning in the message file. The Command Line Processor PRECOMPILE command also does not output the warning in this case.
- For clients that are at an earlier FixPak than Version 7 FixPak 2 or for downlevel clients, the sqlabndx API does report this SQL1179W warning in the message file. However, the message file also incorrectly includes an

SQL0092N message indicating that no package was created. This is not
correct as the package is indeed created. The Command Line Processor
BIND command returns the same erroneous warning.

Chapter 15. Logged Information

Gathering Stack Traceback Information on UNIX-Based Systems

The *Troubleshooting Guide* incorrectly states that to activate stack traceback on
every node of a multi-node system, you need to use the **db2_all** command.
Only the **db2_call_stack** command is needed. Use of **db2_all** and
db2_call_stack together will cause an error.

Chapter 5. Configuration

LOCKLIST

The following information should be added to Table 2.

```
Parameter    Previous Upper Limit    Current Upper Limit
LOCKLIST     60000                   524288
```

shmsys:shminfo_shmmax

DB2 users on the 64-bit Solaris operating system should increase the value of "shmsys:shminfo_shmmax" in /etc/system, as necessary, to be able to allocate a large database shared memory set. The *DB2 for UNIX Quick Beginnings* book recommends setting that parameter to "90% of the physical RAM in the machine, in bytes". This recommendation is also valid for 64-bit implementations.

However, there is a problem with the following recommendation in the *DB2 for UNIX Quick Beginnings* book: For 32-bit systems with more than 4 GB of RAM (up to 64 GB in total is possible on the Solaris operating system), if a user sets the shmmax value to a number larger than 4 GB, and is using a 32-bit kernel, the kernel only looks at the lower 32 bits of the number, sometimes resulting in a very small value for shmmax.

Chapter 6. Restrictions

There is currently no LDAP support on 64-bit operating systems.

32-bit and 64-bit databases cannot be created on the same path. For example, if a 32-bit database exists on <somepath>, then:

```
db2 create db <somedb> on <somepath>
```

if issued from a 64-bit instance, fails with "SQL10004C An I/O error occurred while accessing the database directory."

XML Extender Administration and Programming

Release Notes for the IBM DB2 XML Extender can be found on the DB2 XML Web site:
http://www.ibm.com/software/data/db2/extenders/xmlext/library.html

MQSeries

This section describes how DB2 and MQSeries can be used to construct applications that combine messaging and database access. The focus in this section will be a set of functions, similar to User-Defined Functions (UDFs), that may be optionally enabled in DB2 Universal Database, Version 7.2. Using these basic functions, it is possible to support a wide range of applications, from simple event notification to data warehousing.

For more information about data warehousing applications, refer to the newly refreshed Data Warehouse Center Administration Guide, which you can obtain from http://www.ibm.com/software/data/db2/udb/winos2unix/support.

Installation and Configuration for the DB2 MQSeries Functions

This section describes how to configure a DB2 environment to use the DB2 MQSeries Functions. Upon successful completion of the following procedure you will be able to use the DB2 MQSeries Functions from within SQL. A description of these functions can be found in the SQL Reference section of the Release Notes.

The basic procedure for configuring and enabling the DB2 MQSeries Functions is:

1. Install MQSeries on each physical machine.
2. Install MQSeries AMI on physical machine.
3. Enable and configure the DB2 MQSeries Functions.

In addition, to make use of the publish/subscribe capabilities provided by the DB2 MQSeries Functions, you must also install either MQSeries Integrator or the MQSeries Publish/Subscribe Functions on each physical machine. Information on MQSeries Integrator can be found at http://www.ibm.com/software/ts/mqseries/integrator. Information on the MQSeries Publish/Subscribe feature can be found at http://www.ibm.com/software/ts/mqseries/txppacs under category 3.

Install MQSeries

The first step is to ensure that a minimum of MQSeries Version 5.1 with the latest FixPak is installed on your DB2 server. If this version of MQSeries is already installed then skip to the next step, "Install MQSeries AMI." DB2 Version 7.2 includes a copy of the MQSeries server to be used with DB2. Platform specific instructions for installing MQSeries or for upgrading an

existing MQSeries installation can be found in a platform specific Quick Beginnings book at http://www.ibm.com/software/ts/mqseries/library/manuals. Be sure to set up a default queue manager as you go through the installation process.

Install MQSeries AMI

The next step is to install the MQSeries Application Messaging Interface (AMI). This is an extension to the MQSeries programming interfaces that provides a clean separation of administrative and programming tasks. The DB2 MQSeries Functions require the installation of this interface. If the MQSeries AMI is already installed on your DB2 server then skip to the next step, "Enable DB2 MQSeries Functions." If the MQSeries AMI is not installed then you can do so from either the installation package provided with DB2 7.2 or by downloading a copy of the AMI from the MQSeries Support Pacs web site at http://www.ibm.com/software/ts/mqseries/txppacs. The AMI may be found under "Category 3 – Product Extensions." For convenience, we have provided a copy of the MQSeries AMI with DB2. This file is located in the sqllib/cfg directory. The name of the file is operating system dependent:

AIX Version 4.3 and greater	ma0f_ax.tar.Z
HP-UX	ma0f_hp.tar.Z
Solaris Operating Environment	ma0f_sol7.tar.Z or mq0f_sol26.tar.Z
Windows 32-bit	ma0f_nt.zip

Follow the normal AMI installation process as outlined in the AMI readme file contained in the compressed installation image.

Enable DB2 MQSeries Functions

During this step, you will configure and enable a database for the DB2 MQSeries Functions. The **enable_MQFunctions** utility is a flexible command that first checks that the proper MQSeries environment has been set up and then installs and creates a default configuration for the DB2 MQSeries functions, enables the specified database with these functions, and confirms that the configuration works.

1. For Windows NT or Windows 2000, go to step 5.
2. Setting Groups on UNIX: If you are enabling these functions on UNIX, you must first add the DB2 instance owner (often db2inst1) and the userid associated with fenced UDFs (often db2fenc1) to the MQSeries group mqm. This is needed for the DB2 functions to access MQSeries.
3. Set DB2 Environment Variables on UNIX: Add the AMT_DATA_PATH environment variable to the list understood by DB2. You can edit the file $INSTHOME/sqllib/profile.env, add AMT_DATA_PATH to DB2ENVLIST. The **db2set** command can also be used.

4. On UNIX, restart the database instance: For the environment variable changes to take effect, the database instance must be restarted.

5. Change directory to `$INSTHOME/sqllib/cfg` for UNIX or `%DB2PATH%/cfg` on Windows.

6. Run the command **enable_MQFunctions** to configure and enable a database for the DB2 MQSeries Functions. In a DB2 UDB EEE environment, only carry out this step on the catalog node. Refer to "enable_MQFunctions" on page 207 for a complete description of this command. Some common examples are given below. After successful completion, the specified database will have been enabled and the configuration tested.

7. To test these functions using the Command Line Processor, issue the following commands after you have connected to the enabled database:

```
values DB2MQ.MQSEND('a test')
values DB2MQ.MQRECEIVE()
```

The first statement will send the message "a test" to the DB2MQ_DEFAULT_Q queue and the second will receive it back.

Note: As a result of running **enable_MQFunctions**, a default MQSeries environment will be established. The MQSeries queue manager DB2MQ_DEFAULT_MQM and the default queue DB2MQ_DEFAULT_Q will be created. The files `amt.xml`, `amthost.xml`, and `amt.dtd` will be created if they do not already exist in the directory pointed to by AMT_DATA_PATH. If an `amthost.xml` file does exist, and does not contain a definition for connectionDB2MQ, then a line will be added to the file with the appropriate information. A copy of the original file will be saved as `DB2MQSAVE.amthost.xml`.

MQSeries Messaging Styles

The DB2 MQSeries functions support three messaging models: datagrams, publish/subscribe (p/s), and request/reply (r/r).

Messages sent as datagrams are sent to a single destination with no reply expected. In the p/s model, one or more publishers send a message to a publication service which distributes the message to one or more subscribers. Request/reply is similar to datagram, but the sender expects to receive a response.

Message Structure

MQSeries does not, itself, mandate or support any particular structuring of the messages it transports.

Other products, such as MQSeries Integrator (MQSI) do offer support for messages formed as C or Cobol or as XML strings. Structured messages in MQSI are defined by a message repository. XML messages typically have a self-describing message structure and may also be managed through the repository. Messages may also be unstructured, requiring user code to parse or construct the message content. Such messages are often semi-structured, that is, they use either byte positions or fixed delimiters to separate the fields within a message. Support for such semi-structured messages is provided by the MQSeries Assist Wizard. Support for XML messages is provided through some new features to the DB2 XML Extender.

MQSeries Functional Overview

A set of MQSeries functions are provided with DB2 UDB Version 7.2 to allow SQL statements to include messaging operations. This means that this support is available to applications written in any supported language, for example, C, Java, SQL using any of the database interfaces. All examples shown below are in SQL. This SQL may be used from other programming languages in all the standard ways. All of the MQSeries messaging styles described above are supported. For more information about the MQSeries functions, see the SQL Reference section of the Release Notes.

In a basic configuration, an MQSeries server is located on the database server machine along with DB2. The MQSeries functions are installed into DB2 and provide access to the MQSeries server. DB2 clients may be located on any machine accessible to the DB2 server. Multiple clients can concurrently access the MQSeries functions through the database. Through the provided functions, DB2 clients may perform messaging operations within SQL statements. These messaging operations allow DB2 applications to communicate among themselves or with other MQSeries applications.

The **enable_MQFunctions** command is used to enable a DB2 database for the MQSeries functions. It will automatically establish a simple default configuration that client applications may utilize with no further administrative action. For a description, see "enable_MQFunctions" on page 207 and "disable_MQFunctions" on page 210. The default configuration allows application programmers a quick way to get started and a simpler interface for development. Additional functionality may be configured incrementally as needed.

Example 1: To send a simple message using the default configuration, the SQL statement would be:
```
VALUES DB2MQ.MQSEND('simple message')
```

This will send the message simple message to the MQSeries queue manager and queue specified by the default configuration.

The Application Messaging Interface (AMI) of MQSeries provides a clean separation between messaging actions and the definitions that dictate how those actions should be carried out. These definitions are kept in an external repository file and managed using the AMI Administration tool. This makes AMI applications simple to develop and maintain. The MQSeries functions provided with DB2 are based on the AMI MQSeries interface. AMI supports the use of an external configuration file, called the AMI Repository, to store configuration information. The default configuration includes an MQSeries AMI Repository configured for use with DB2.

Two key concepts in MQSeries AMI, service points and policies, are carried forward into the DB2 MQSeries functions. A service point is a logical end-point from which a message may be sent or received. In the AMI repository, each service point is defined with an MQSeries queue name and queue manager. Policies define the quality of service options that should be used for a given messaging operation. Key qualities of service include message priority and persistence. Default service points and policy definitions are provided and may be used by developers to further simplify their applications. Example 1 can be re-written as follows to explicitly specify the default service point and policy name:

Example 2:
```
VALUES DB2MQ.MQSEND('DB2.DEFAULT.SERVICE', 'DB2.DEFAULT.POLICY',
                                      'simple message')
```

Queues may be serviced by one or more applications at the server upon which the queues and applications reside. In many configurations multiple queues will be defined to support different applications and purposes. For this reason, it is often important to define different service points when making MQSeries requests. This is demonstrated in the following example:

Example 3:
```
VALUES DB2MQ.MQSEND('ODS_Input', 'simple message')
```

Note: In this example, the policy is not specified and thus the default policy will be used.

Limitations

MQSeries provides the ability for message operations and database operations to be combined in a single unit of work as an atomic transaction. This feature is not initially supported by the MQSeries Functions on Unix and Windows.

When using the sending or receiving functions, the maximum length of a message of type VARCHAR is 4000 characters. The maximum length when sending or receiving a message of type CLOB is 1 MB. These are also the maximum message sizes for publishing a message using MQPublish.

Different functions are sometimes required when working with CLOB messages and VARCHAR messages. Generally, the CLOB version of an MQ function uses the identical syntax as its counterpart. The only difference is that its name has the characters CLOB at the end. For example, the CLOB equivalent of MQREAD is MQREADCLOB. For a detailed list of these functions, see "CLOB data now supported in MQSeries functions" on page 353.

Error Codes

The return codes returned by the MQSeries Functions can be found in Appendix B of the MQSeries Application Messaging Interface Manual.

Usage Scenarios

The MQSeries Functions can be used in a wide variety of scenarios. This section will review some of the more common scenarios, including Basic Messaging, Application Connectivity and Data Publication.

Basic Messaging

The most basic form of messaging with the MQSeries DB2 Functions occurs when all database applications connect to the same DB2 server. Clients may be local to the database server or distributed in a network environment.

In a simple scenario, Client A invokes the MQSEND function to send a user-defined string to the default service location. The MQSeries functions are then executed within DB2 on the database server. At some later time, Client B invokes the MQRECEIVE function to remove the message at the head of the queue defined by the default service and return it to the client. Again, the MQSeries functions to perform this work are executed by DB2.

Database clients can use simple messaging in a number of ways. Some common uses for messaging are:

- Data collection -- Information is received in the form of messages from one or more possibly diverse sources of information. Information sources may be commercial applications such as SAP or applications developed in-house. Such data may be received from queues and stored in database tables for further processing or analysis.
- Workload distribution -- Work requests are posted to a queue shared by multiple instances of the same application. When an instance is ready to perform some work it receives a message from the top of the queue containing a work request to perform. Using this technique, multiple instances can share the workload represented by a single queue of pooled requests.
- Application signaling -- In a situation where several processes collaborate, messages are often used to coordinate their efforts. These messages may contain commands or requests for work to be performed. Typically, this

kind of signaling is one-way; that is, the party that initiates the message does not expect a reply. See "Request/Reply Communications" on page 203 for more information.

- Application notification -- Notification is similar to signaling in that data is sent from an initiator with no expectation of a response. Typically, however, notification contains data about business events that have taken place. "Publish/Subscribe" on page 204 is a more advanced form of notification.

The following scenario extends the simple scenario described above to incorporate remote messaging. That is, a message is sent between Machine A and Machine B. The sequence of steps is as follows:

1. The DB2 Client executes an MQSEND call, specifying a target service that has been defined to represent a remote queue on Machine B.

2. The MQSeries DB2 functions perform the actual MQSeries work to send the message. The MQSeries server on Machine A accepts the message and guarantees that it will deliver it to the destination defined by the service point definition and current MQSeries configuration of Machine A. The server determines that this is a queue on Machine B. It then attempts to deliver the message to the MQSeries server on Machine B, transparently retrying as needed.

3. The MQSeries server on Machine B accepts the message from the server on Machine A and places it in the destination queue on Machine B.

4. An MQSeries client on Machine B requests the message at the head of the queue.

Sending Messages

Using MQSEND, a DB2 user or developer chooses what data to send, where to send it, and when it will be sent. In the industry this is commonly called "Send and Forget," meaning that the sender just sends a message, relying on the guaranteed delivery protocols of MQSeries to ensure that the message reaches its destination. The following examples illustrate this.

Example 4: To send a user-defined string to the service point myPlace with the policy highPriority:

```
VALUES DB2MQ.MQSEND('myplace','highPriority','test')
```

Here, the policy highPriority refers to a policy defined in the AMI Repository that sets the MQSeries priority to the highest level and perhaps adjusts other qualities of service, such as persistence, as well.

The message content may be composed of any legal combination of SQL and user-specified data. This includes nested functions, operators, and casts. For instance, given a table EMPLOYEE, with VARCHAR columns LASTNAME,

FIRSTNAME, and DEPARTMENT, to send a message containing this
information for each employee in DEPARTMENT 5LGA you would do the
following:

Example 5:
```
SELECT DB2MQ.MQSEND(LASTNAME || ' ' || FIRSTNAME || ' ' || DEPARTMENT)
   FROM EMPLOYEE
   WHERE DEPARTMENT = '5LGA'
```

If this table also had an integer AGE column, it could be included as follows:

Example 6:
```
SELECT DB2MQ.MQSEND
      (LASTNAME || ' ' || FIRSTNAME || ' ' || DEPARTMENT|| ' ' || char(AGE))
   FROM EMPLOYEE
   WHERE DEPARTMENT = '5LGA'
```

If the table EMPLOYEE had a column RESUME of type CLOB instead of an
AGE column, then a message containing the information for each employee in
DEPARTMENT 5LGA could be sent out with the following:

Example 7:
```
SELECT DB2MQ.MQSEND
   (clob(LASTNAME) || ' ' || clob(FIRSTNAME) || ' ' ||
    clob(DEPARTMENT) || ' ' || RESUME))
FROM EMPLOYEE
WHERE DEPARTMENT = '5LGA'
```

Example 8:

Finally, the following example shows how message content may be derived
using any valid SQL expression. Given a second table DEPT containing
VARCHAR columns DEPT_NO and DEPT_NAME, messages can be sent that
contain employee LASTNAME and DEPT_NAME:

Example 8:
```
SELECT DB2MQ.MQSEND(e.LASTNAME || ' ' || d.DEPTNAME) FROM EMPLOYEE e, DEPT d
   WHERE e.DEPARTMENT = d.DEPTNAME
```

Retrieving Messages

The MQSeries DB2 Functions allow messages to be either received or read.
The difference between reading and receiving is that reading returns the
message at the head of a queue without removing it from the queue, while
receiving operations cause the message to be removed from the queue. A
message retrieved using a receive operation can only be retrieved once, while
a message retrieved using a read operation allows the same message to be
retrieved many times. The following examples demonstrate this:

Example 8:
```
VALUES DB2MQ.MQREAD()
```

This example returns a VARCHAR string containing the message at the head of queue defined by the default service using the default quality of service policy. It is important to note that if no messages are available to be read, a null value will be returned. The queue is not changed by this operation.

Example 9:
```
VALUES DB2MQ.MQRECEIVE('Employee_Changes')
```

The above example shows how a message can be removed from the head of the queue defined by the Employee_Changes service using the default policy.

One very powerful feature of DB2 is the ability to generate a table from a user-defined (or DB2–provided) function. You can exploit this table function feature to allow the contents of a queue to be materialized as a DB2 table. The following example demonstrates the simplest form of this:

Example 10:
```
SELECT t.* FROM table ( DB2MQ.MQREADALL()) t
```

This query returns a table consisting of all of the messages in the queue defined by the default service and the metadata about these messages. While the full definition of the table structure returned is defined in the Appendix, the first column reflects the contents of the message and the remaining columns contain the metadata. To return just the messages, the example could be rewritten:

Example 11:
```
SELECT t.MSG FROM table (DB2MQ.MQREADALL()) t
```

The table returned by a table function is no different from a table retrieved from the database directly. This means that you can use this table in a wide variety of ways. For instance, you can join the contents of the table with another table or count the number of messages in a queue:

Example 12:
```
SELECT t.MSG, e.LASTNAME
   FROM table (DB2MQ.MQREADALL() ) t, EMPLOYEE e
      WHERE t.MSG = e.LASTNAME
```

Example 13:
```
SELECT COUNT(*) FROM table (DB2MQ.MQREADALL()) t
```

You can also hide the fact that the source of the table is a queue by creating a view over a table function. For instance, the following example creates a view called NEW_EMP over the queue referred to by the service named NEW_EMPLOYEES:

Example 14:
```
CREATE VIEW NEW_EMP (msg) AS
   SELECT t.msg FROM table (DB2MQ.MQREADALL()) t
```

In this case, the view is defined with only a single column containing an entire message. If messages are simply structured, for instance containing two fields of fixed length, it is straightforward to use the DB2 built-in functions to parse the message into the two columns. For example, if you know that messages sent to a particular queue always contain an 18-character last name followed by an 18-character first name, then you can define a view containing each field as a separate column as follows:

Example 15:
```
CREATE VIEW NEW_EMP2 AS
   SELECT left(t.msg,18) AS LNAME, right(t.msg,18) AS FNAME
   FROM table(DB2MQ.MQREADALL()) t
```

A new feature of the DB2 Stored Procedure Builder, the MQSeries Assist Wizard, can be used to create new DB2 table functions and views that will map delimited message structures to columns.

Finally, it is often desirable to store the contents of one or more messages into the database. This may be done using the full power of SQL to manipulate and store message content. Perhaps the simplest example of this is:

Example 16:
```
INSERT INTO MESSAGES
   SELECT t.msg FROM table (DB2MQ.MQRECEIVEALL()) t
```

Given a table MESSAGES, with a single column of VARCHAR(2000), the statement above will insert the messages from the default service queue into the table. This technique can be embellished to cover a very wide variety of circumstances.

Application-to-Application Connectivity

Application integration is a common element in many solutions. Whether integrating a purchased application into an existing infrastructure or just integrating a newly developed application into an existing environment, we are often faced with the task of glueing a heterogeneous collection of subsystems together to form a working whole. MQSeries is commonly viewed as an essential tool for integrating applications. Accessible in most hardware,

software, and language environments, MQSeries provides the means to interconnect a very heterogeneous collection of applications.

This section will discuss some application integration scenarios and how they may be used with DB2. As the topic is quite broad, a comprehensive treatment of Application Integration is beyond the scope of this work. Therefore, the focus is on just two simple topics: Request/Reply communications, and MQSeries Integrator and Publish/Subscribe.

Request/Reply Communications
The Request/Reply (R/R) communications method is a very common technique for one application to request the services of another. One way to do this is for the requester to send a message to the service provider requesting some work to be performed. Once the work has been completed, the provider may decide to send results (or just a confirmation of completion) back to the requestor. But using the basic messaging techniques described above, there is nothing that connects the sender's request with the service provider's response. Unless the requester waits for a reply before continuing, some mechanism must be used to associate each reply with its request. Rather than force the developer to create such a mechanism, MQSeries provides a correlation identifier that allows the correlation of messages in an exchange.

While there are a number of ways in which this mechanism could be used, the simplest is for the requestor to mark a message with a known correlation identifier using, for instance, the following:

Example 17:
```
DB2MQ.MQSEND ('myRequester','myPolicy','SendStatus:cust1','Req1')
```

This statement adds a final parameter Req1 to the MQSEND statement from above to indicate the correlation identifier for the request.

To receive a reply to this specific request, use the corresponding MQRECEIVE statement to selectively retrieve the first message defined by the indicated service that matches this correlation identifier as follows:

Example 18:
```
DB2MQ.MQRECEIVE('myReceiver','myPolicy','Req1')
```

If the application servicing the request is busy and the requestor issues the above MQRECEIVE before the reply is sent, then no messages matching this correlation identifier will be found.

To receive both the service request and the correlation identifier a statement like the following is used:

Example 19:
```
SELECT msg, correlid FROM
          table (DB2MQ.MQRECEIVEALL('aServiceProvider','myPolicy',1)) t
```

This returns the message and correlation identifier of the first request from the service aServiceProvider.

Once the service has been performed, it sends the reply message to the queue described by aRequester. Meanwhile, the service requester could have been doing other work. In fact, there is no guarantee that the initial service request will be responded to within a set time. Application level timeouts such as this must be managed by the developer; the requester must poll to detect the presence of the reply.

The advantage of such time-independent asynchronous processing is that the requester and service provider execute completely independently of one another. This can be used both to accommodate environments in which applications are only intermittently connected and more batch-oriented environments in which multiple requests or replies are aggregated before processing. This kind of aggregation is often used in data warehouse environments to periodically update a data warehouse or operational data store.

Publish/Subscribe

Simple Data Publication: Another common scenario in application integration is for one application to notify other applications about events of interest. This is easily done by sending a message to a queue monitored by another application. The contents of the message can be a user-defined string or can be composed from database columns. Often a simple message is all that needs to be sent using the MQSEND function. When such messages need to be sent concurrently to multiple recipients, the Distribution List facility of the MQSeries AMI can be used.

A distribution list is defined using the AMI Administration tool. A distribution list comprises a list of individual services. A message sent to a distribution list is forwarded to every service defined within the list. This is especially useful when it is known that a few services will always be interested in every message. The following example shows sending of a message to the distribution list interestedParties:

Example 20:
```
DB2MQ.MQSEND('interestedParties','information of general interest');
```

When more control over the messages that particular services should receive is required, a Publish/Subscribe capability is needed. Publish/Subscribe

systems typically provide a scalable, secure environment in which many subscribers can register to receive messages from multiple publishers. To support this capability the MQPublish interface can be used, in conjunction with MQSeries Integrator or the MQSeries Publish/Subscribe facility.

MQPublish allows users to optionally specify a topic to be associated with a message. Topics allow a subscriber to more clearly specify the messages to be accepted. The sequence of steps is as follows:

1. An MQSeries administrator configures MQSeries Integrator publish/subscribe capabilities.

2. Interested applications subscribe to subscription points defined by the MQSI configuration, optionally specifying topics of interest to them. Each subscriber selects relevant topics, and can also utilize the content-based subscription techniques of MQSeries Integrator V2. It is important to note that queues, as represented by service names, define the subscriber.

3. A DB2 application publishes a message to the service point Weather. The messages indicates that the weather is Sleet with a topic of Austin, thus notifying interested subscribers that the weather in Austin is Sleet.

4. The mechanics of actually publishing the message are handled by the MQSeries functions provided by DB2. The message is sent to MQSeries Integrator using the service named Weather.

5. MQSI accepts the message from the Weather service, performs any processing defined by the MQSI configuration, and determines which subscriptions it satisfies. MQSI then forwards the message to the subscriber queues whose criteria it meets.

6. Applications that have subscribed to the Weather service, and registered an interest in Austin will receive the message Sleet in their receiving service.

To publish this data using all the defaults and a null topic, you would use the following statement:

Example 21:

```
SELECT DB2MQ.MQPUBLISH
                (LASTNAME || ' ' || FIRSTNAME || ' ' ||
                 DEPARTMENT || ' ' ||char(AGE))
   FROM EMPLOYEE
      WHERE DEPARTMENT = '5LGA'
```

Fully specifying all the parameters and simplifying the message to contain only the LASTNAME the statement would look like:

Example 22:

```
SELECT DB2MQ.MQPUBLISH('HR_INFO_PUB', 'SPECIAL_POLICY', LASTNAME,
   'ALL_EMP:5LGA', 'MANAGER')
   FROM EMPLOYEE
      WHERE DEPARTMENT = '5LGA'
```

This statement publishes messages to the HR_INFO_PUB publication service using the SPECIAL_POLICY service. The messages indicate that the sender is the MANAGER topic. The topic string demonstrates that multiple topics, concatenated using a ':' can be specified. In this example, the use of two topics allows subscribers to register for either ALL_EMP or just 5LGA to receive these messages.

To receive published messages, you must first register your interest in messages containing a given topic and indicate the name of the subscriber service that messages should be sent to. It is important to note that an AMI subscriber service defines a broker service and a receiver service. The broker service is how the subscriber communicates with the publish/subscribe broker and the receiver service is where messages matching the subscription request will be sent. The following statement registers an interest in the topic ALL_EMP.

Example 23:
```
DB2MQ.MQSUBSCRIBE('aSubscriber', 'ALL_EMP')
```

Once an application has subscribed, messages published with the topic ALL_EMP will be forwarded to the receiver service defined by the subscriber service. An application can have multiple concurrent subscriptions. To obtain the messages that meet your subscription, any of the standard message retrieval functions can be used. For instance if the subscriber service aSubscriber defines the receiver service to be aSubscriberReceiver then the following statement will non-destructively read the first message:

Example 24:
```
DB2MQ.MQREAD('aSubscriberReceiver')
```

To determine both the messages and the topics that they were published under, you would use one of the table functions. The following statement would receive the first five messages from aSubscriberReceiver and display both the message and the topic:

Example 25:
```
SELECT t.msg, t.topic
   FROM table (DB2MQ.MQRECEIVEALL('aSubscriberReceiver',5)) t
```

To read all of the messages with the topic ALL_EMP, you can leverage the power of SQL to issue:

Example 26:

```
SELECT t.msg FROM table (DB2MQ.MQREADALL('aSubscriberReceiver')) t
   WHERE t.topic = 'ALL_EMP'
```

Note: It is important to realize that if MQRECEIVEALL is used with a
constraint then the entire queue will be consumed, not just those
messages published with topic ALL_EMP. This is because the table
function is performed before the constraint is applied.

When you are no longer interested in subscribing to a particular topic you
must explicitly unsubscribe using a statement such as:

Example 27:

```
DB2MQ.MQUNSUBSCRIBE('aSubscriber', 'ALL_EMP')
```

Once this statement is issued the publish/subscribe broker will no longer
deliver messages matching this subscription.

Automated Publication: Another important technique in database messaging
is automated publication. Using the trigger facility within DB2, you can
automatically publish messages as part of a trigger invocation. While other
techniques exist for automated data publication, the trigger-based approach
allows administrators or developers great freedom in constructing the
message content and flexibility in defining the trigger actions. As with any
use of triggers, attention must be paid to the frequency and cost of execution.
The following examples demonstrate how triggers may be used with the
MQSeries DB2 Functions.

The example below shows how easy it is to publish a message each time a
new employee is hired. Any users or applications subscribing to the
HR_INFO_PUB service with a registered interest in NEW_EMP will receive a
message containing the date, name and department of each new employee.

Example 28:

```
CREATE TRIGGER new_employee AFTER INSERT ON employee REFERENCING NEW AS n
     FOR EACH ROW MODE DB2SQL
     VALUES DB2MQ.MQPUBLISH('HR_INFO_PUB&',  'NEW_EMP',
     current date || ' ' || LASTNAME || ' ' || DEPARTMENT)
```

enable_MQFunctions

enable_MQFunctions

Enables DB2 MQSeries functions for the specified database and validates that the DB2 MQSeries functions can be executed properly. The command will fail if MQSeries and MQSeries AMI have not been installed and configured.

Authorization
One of the following:

- *sysadm*
- *dbadm*
- IMPLICIT_SCHEMA on the database, if the implicit or explicit schema name of the function does not exist
- CREATEIN privilege on the schema, if the schema name, DB2MQ, exists

Command Syntax

```
►►──enable_MQFunctions──-n─database──-u─userid──-p─password──┬──────────┬──►
                                                             └─force────┘

►──┬──────────────┬───────────────────────────────────────────────────►◄
   └─noValidate───┘
```

Command Parameters

-n database
    Specifies the name of the database to be enabled.

-u userid
    Specifies the user ID to connect to the database.

-p password
    Specifies the password for the user ID.

-force Specifies that warnings encountered during a re-installation should be ignored.

-noValidate
    Specifies that validation of the DB2 MQSeries functions will not be performed.

Examples
In the following example, DB2MQ functions are being created. The user connects to the database SAMPLE. The default schema DB2MQ is being used.

```
enable_MQFunctions -n sample -u user1 -p password1
```

Usage Notes
The DB2 MQ functions run under the schema DB2MQ which is automatically created by this command.

Before Executing this command:

- Ensure that MQ and AMI are installed, and that the version of MQSeries is 5.1 or higher.
- Ensure that the environment variable $AMT_DATA_PATH is defined.
- Change the directory to the cfg subdirectory of the DB2PATH

On UNIX:

- Use db2set to add AMT_DATA_PATH to the DB2ENVLIST.
- Ensure that the user account associated with UDF execution is a member of the mqm group.
- Ensure that the user who will be calling this command is a member if the mqm group.

Note: AIX 4.2 is not supported by MQSeries 5.2.

disable_MQFunctions

disable_MQFunctions

Disables the use of DB2 MQSeries functions for the specified database.

Authorization
One of the following:
- *sysadm*
- *dbadm*
- IMPLICIT_SCHEMA on the database, if the implicit or explicit schema name of the function does not exist
- CREATEIN privilege on the schema, if the schema name, DB2MQ, exists

Command Syntax

```
►►──disable_MQFunctions──-n─database──-u─userid──-p─password───────────►◄
```

Command Parameters

-n database

Specifies the name of the database.

-u userid

Specifies the user ID used to connect to the database.

-p password

Specifies the password for the user ID.

Examples
In the following example, DB2MQ functions are disabled for the database SAMPLE.

```
disable_MQFunctions -n sample -u user1 -p password1
```

Part 4. Administrative Tools

Additional Setup Before Running Tools

Disabling the Floating Point Stack on Linux

On a Linux environment with glibc 2.2.x, you need to disable the floating point stack before running the DB2 Java GUI tools, such as the Control Center. To disable the floating point stack, set the LD_ASSUME_KERNEL environment variable to 2.2.5 as follows:

```
bash$ export LD_ASSUME_KERNEL=2.2.5
```

Specific Java Level Required in a Japanese Linux Environment

Linux users need a specific JDK level when running the DB2 Java GUI tools, such as the Control Center, on a Japanese environment. For example, Red Hat Linux 6.2J/7J/7.1/7.2 users should use IBMJava118-SDK-1.1.8-2.0.i386.rpm level.

Control Center

Choosing Redirected Restore Commits You to Restoring the Database

When restoring a database using the GUI tools, selecting the redirected restore option commits you to restoring the database. Once you select the option, the restore operation starts in the background and the database is placed in restore pending state. If you then cancel your action, the database will not be available until you complete another restore.

Ability to Administer DB2 Server for VSE and VM Servers

The DB2 Universal Database Version 7 Control Center has enhanced its support of DB2 Server for VSE and VM databases. All DB2 Server for VSE and VM database objects can be viewed by the Control Center. There is also support for the CREATE INDEX, REORGANIZE INDEX, and UPDATE STATISTICS statements, and for the REBIND command. REORGANIZE INDEX and REBIND require a stored procedure running on the DB2 Server for VSE and VM hosts. This stored procedure is supplied by the Control Center for VSE and VM feature of DB2 Server for VSE and VM.

The fully integrated Control Center allows the user to manage DB2, regardless of the platform on which the DB2 server runs. DB2 Server for VSE and VM objects are displayed on the Control Center main window, along with DB2 Universal Database objects. The corresponding actions and utilities to manage these objects are invoked by selecting the object. For example, a user can list the indexes of a particular database, select one of the indexes, and reorganize it. The user can also list the tables of a database and run update statistics, or define a table as a replication source.

For information about configuring the Control Center to perform administration tasks on DB2 Server for VSE and VM objects, refer to the *DB2 Connect User's Guide*, or the *Installation and Configuration Supplement*.

Java 1.2 Support for the Control Center

The Control Center supports bi-directional languages, such as Arabic and Hebrew, using bi-di support in Java 1.2. This support is provided for the Windows NT platform only.

Java 1.2 must be installed for the Control Center to recognize and use it:

1. JDK 1.2.2 is available on the DB2 UDB CD under the DB2\bidi\NT directory. `ibm-inst-n122p-win32-x86.exe` is the installer program, and

`ibm-jdk-n122p-win32-x86.exe` is the JDK distribution. Copy both files to a temporary directory on your hard drive, then run the installer program from there.

2. Install it under `<DB2PATH>\java\Java12`, where `<DB2PATH>` is the installation path of DB2.

3. Do not select JDK/JRE as the System VM when prompted by the JDK/JRE installation.

After Java 1.2 is installed successfully, starting the Control Center in the normal manner will use Java 1.2.

To stop the use of Java 1.2, you may either uninstall JDK/JRE from `<DB2PATH>\java\Java12`, or simply rename the `<DB2PATH>\java\Java12` sub-directory to something else.

Note: Do not confuse `<DB2PATH>\java\Java12` with `<DB2PATH>\Java12`. `<DB2PATH>\Java12` is part of the DB2 installation, and includes JDBC support for Java 1.2.

"Invalid shortcut" Error when Using the Online Help on the Windows Operating System

When using the Control Center online help, you may encounter an error like: "Invalid shortcut". If you have recently installed a new Web browser or a new version of a Web browser, ensure that HTML and HTM documents are associated with the correct browser. See the Windows Help topic "To change which program starts when you open a file".

Keyboard Shortcuts Not Working

In some languages, for the Control Center on UNIX based systems and on OS/2, some keyboard shortcuts (hotkeys) do not work. Please use the mouse to select options.

Java Control Center on OS/2

The Control Center must be installed on an HPFS-formatted drive.

"File access denied" Error when Attempting to View a Completed Job in the Journal on the Windows Operating System

On DB2 Universal Database for Windows NT, a "File access denied" error occurs when attempting to open the Journal to view the details of a job created in the Script Center. The job status shows complete. This behavior occurs when a job created in the Script Center contains the START command. To avoid this behavior, use START/WAIT instead of START in both the batch file and in the job itself.

Multisite Update Test Connect

Multisite Update Test Connect functionality in the Version 7 Control Center is limited by the version of the target instance. The target instance must be at least Version 7 for the "remote" test connect functionality to run. To run Multisite Update Test Connect functionality in Version 6, you must bring up the Control Center locally on the target instance and run it from there.

Control Center for DB2 for OS/390

The DB2 UDB Control Center for OS/390 allows you to manage the use of your licensed IBM DB2 utilities. Utility functions that are elements of separately orderable features of DB2 UDB for OS/390 must be licensed and installed in your environment before being managed by the DB2 Control Center.

The "CC390" database, defined with the Control Center when you configure a DB2 for OS/390 subsystem, is used for internal support of the Control Center. Do not modify this database.

Although DB2 for OS/390 Version 7.1 is not mentioned specifically in the Control Center table of contents, or the Information Center Task information, the documentation does support the DB2 for OS/390 Version 7.1 functions. Many of the DB2 for OS/390 Version 6-specific functions also relate to DB2 for OS/390 Version 7.1, and some functions that are DB2 for OS/390 Version 7.1-specific in the table of contents have no version designation. If you have configured a DB2 for OS/390 Version 7.1 subsystem on your Control Center, you have access to all the documentation for that version.

To access and use the Generate DDL function from the Control Center for DB2 for OS/390, you must have the Generate DDL function installed:

- For Version 5, install DB2Admin 2.0 with DB2 for OS/390 Version 5.
- For Version 6, install the small programming enhancement that will be available as a PTF for the DB2 Admin feature of DB2 for OS/390 Version 6.

- For Version 7.1, the Generate DDL function is part of the separately priced DB2 Admin feature of DB2 for OS/390 Version 7.1.

You can access Stored Procedure Builder from the Control Center, but you must have already installed it by the time you start the DB2 UDB Control Center. It is part of the DB2 Application Development Client.

To catalog a DB2 for OS/390 subsystem directly on the workstation, select to use the Client Configuration Assistant tool.

1. On the Source page, specify the **Manually configure a connection to a database** radio button.
2. On the Protocol page, complete the appropriate communications information.
3. On the Database page, specify the subsystem name in the **Database name** field.
4. On the Node Options page, select the **Configure node options (Optional)** check box.
5. Select MVS/ESA, OS/390 from the list in the **Operating system** field.
6. Click **Finish** to complete the configuration.

To catalog a DB2 for OS/390 subsystem via a gateway machine, follow steps 1-6 above on the gateway machine, and then:

1. On the client machine, start the Control Center.
2. Right click on the Systems folder and select **Add**.
3. In the Add System dialog, type the gateway machine name in the **System name** field.
4. Type DB2DAS00 in the **Remote instance** field.
5. For the TCP/IP protocol, in the Protocol parameters, specify the gateway machine's host name in the **Host name** field.
6. Type 523 in the **Service name** field.
7. Click **OK** to add the system. You should now see the gateway machine added under the **Systems** folder.
8. Expand the gateway machine name.
9. Right click on the Instances folder and select **Add**.
10. In the Add Instance dialog, click **Refresh** to list the instances available on the gateway machine. If the gateway machine is a Windows NT system, the DB2 for OS/390 subsystem was probably cataloged under the instance DB2.
11. Select the instance. The protocol parameters are filled in automatically for this instance.
12. Click **OK** to add the instance.

13. Open the **Instances** folder to see the instance you just added.
14. Expand the instance.
15. Right click on the **Databases** folder and select **Add**.
16. Click **Refresh** to display the local databases on the gateway machine. If you are adding a DB2 subsystem in the Add Database dialog, type the subsystem name in the Database name field. Option: Type a local alias name for the subsystem (or the database).
17. Click **OK**.

You have now successfully added the subsystem in the Control Center. When you open the database, you should see the DB2 for OS/390 subsystem displayed.

The first paragraph in the section "Control Center 390" states:

The DB2 UDB Control Center for OS/390 allows you to manage the use of your licensed IBM DB2 utilities. Utility functions that are elements of separately orderable features of DB2 UDB for OS/390 must be licensed and installed in your environment before being managed by the DB2 Control Center.

This section should now read:

The DB2 Control Center for OS/390 allows you to manage the use of your licensed IBM DB2 utilities. Utility functions that are elements of separately orderable products must be licensed and installed in your environment in order to be managed by DB2 Control Center.

Required Fix for Control Center for OS/390

You must apply APAR PQ36382 to the 390 Enablement feature of DB2 for OS/390 Version 5 and DB2 for OS/390 Version 6 to manage these subsystems using the DB2 UDB Control Center for Version 7. Without this fix, you cannot use the DB2 UDB Control Center for Version 7 to run utilities for those subsystems.

The APAR should be applied to the following FMIDs:

```
DB2 for OS/390 Version 5 390 Enablement: FMID JDB551D
DB2 for OS/390 Version 6 390 Enablement: FMID JDB661D
```

Change to the Create Spatial Layer Dialog

The "<<" and ">>" buttons have been removed from the Create Spatial Layer dialog.

Troubleshooting Information for the DB2 Control Center

In the "Control Center Installation and Configuration" chapter in your Quick Beginnings book, the section titled "Troubleshooting Information" tells you to unset your client browser's CLASSPATH from a command window if you are having problems running the Control Center as an applet. This section also tells you to start your browser from the same command window. However, the command for starting your browser is not provided. To launch Internet Explorer, type `start iexplore` and press **Enter**. To launch Netscape, type `start netscape` and press **Enter**. These commands assume that your browser is in your PATH. If it is not, add it to your PATH or switch to your browser's installation directory and reissue the **start** command.

Control Center Troubleshooting on UNIX Based Systems

If you are unable to start the Control Center on a UNIX based system, set the JAVA_HOME environment variable to point to your Java distribution:

- If java is installed under `/usr/jdk118`, set JAVA_HOME to `/usr/jdk118`.
- For the sh, ksh, or bash shell:
    ```
    export JAVA_HOME=/usr/jdk118.
    ```
- For the csh or tcsh shell:
    ```
    setenv JAVA_HOME /usr/jdk118
    ```

Possible Infopops Problem on OS/2

If you are running the Control Center on OS/2, using screen size 1024x768 with 256 colors, and with Workplace Shell Palette Awareness enabled, infopops that extend beyond the border of the current window may be displayed with black text on a black background. To fix this problem, either change the display setting to more than 256 colors, or disable Workplace Shell Palette Awareness.

Help for the jdk11_path Configuration Parameter

In the Control Center help, the description of the Java Development Kit 1.1 Installation Path (jdk11_path) configuration parameter is missing a line under the sub-heading **Applies To**. The complete list under **Applies To** is:

- Database server with local and remote clients
- Client
- Database server with local clients
- Partitioned database server with local and remote clients
- Satellite database server with local clients

Solaris System Error (SQL10012N) when Using the Script Center or the Journal

When selecting a Solaris system from the Script Center or the Journal, the following error may be encountered:

```
SQL10012N - An unexpected operating system error was received while
loading the specified library "/udbprod/db2as/sqllib/function/unfenced/
db2scdar!ScheduleInfoOpenScan". SQLSTATE=42724.
```

This is caused by a bug in the Solaris runtime linker. To correct this problem, apply the following patch:

```
105490-06 (107733 makes 105490 obsolete)
  for Solaris Operating Environment 2.6
```

Help for the DPREPL.DFT File

In the Control Center, in the help for the Replication page of the Tool Settings notebook, step 5d says:

```
Save the file into the working directory for the
Control Center (for example, SQLLIB\BIN) so that
the system can use it as the default file.
```

Step 5d should say:

```
Save the file into the working directory for the
Control Center (SQLLIB\CC) so that
the system can use it as the default file.
```

Launching More Than One Control Center Applet

You cannot launch more than one Control Center applet simultaneously on the same machine. This restriction applies to Control Center applets running in all supported browsers.

Online Help for the Control Center Running as an Applet

When the Control Center is running as an applet, the F1 key only works in windows and notebooks that have infopops.

You can press the F1 key to bring up infopops in the following components:
- DB2 Universal Database for OS/390
- The wizards

In the rest of the Control Center components, F1 does not bring up any help. To display help for the other components, please use the Help push button, or the Help pull-down menu.

Running the Control Center in Applet Mode (Windows 95)

An attempt to open the Script Center may fail if an invalid user ID and password are specified. Ensure that a valid user ID and password are entered when signing on to the Control Center.

Working with Large Query Results

It is easy for a user to produce a query that returns a large number of rows. It is not so easy for a user to predict how many rows might actually be returned. With a query that could potentially return thousands (or millions) of rows, there are two problems:

1. It can take a long time to retrieve the result.
2. A large amount of client memory can be required to hold the result.

To facilitate this process, DB2 breaks up large result sets into chunks. It will retrieve and display the results of a query one chunk at a time.

As a result:

1. Display time will be reduced as the first chunk of a query is available for viewing while the remaining chunks are being retrieved.
2. Memory requirements on the client will be reduced as only one chunk of a query result will be stored on the client at any given time.

To control the number of query result rows in memory::

1. Open the General page of the Tool Settings notebook.
2. In the **Maximum size** section, select:
   - **Sample Contents** to limit the number of result rows displayed in the Sample Contents window. Specify the chunk size of the result set (number of rows) in the entry field.
   - **Command Center** to limit the number of result rows displayed on the Query Results page of the Command Center. Specify the chunk size of the result set (number of rows) in the entry field.

When working with the results of a query in the Sample Contents window or on the Query Results page of the Command Center, the **Rows in memory** field indicates the number of rows being held in memory for the query. This number will never be greater than the **Maximum size** set. Click on **Next** to retrieve to the next chunk of the result set. When **Next** is inactive, you have reached the end of the result set.

Command Center

Command Center Interactive Page Now Recognizes Statement Terminator

The Command Center's Interactive page now recognizes the Statement termination character specified in the Tool Settings. If a Statement termination character is not specified, then the newline character is used by default.

Information Center

On Windows platforms, the README and PluginEx.java files can be found in `x:\sqllib\samples\java` where `x` is the drive on which DB2 is installed.

On UNIX, the README and PluginEx.java files can be found in `/u/db2inst1`/sqllib/samples/java where `/u/db2inst1` represents the directory in which DB2 is installed.

"Invalid shortcut" Error on the Windows Operating System

When using the Information Center, you may encounter an error like: "Invalid shortcut". If you have recently installed a new Web browser or a new version of a Web browser, ensure that HTML and HTM documents are associated with the correct browser. See the Windows Help topic "To change which program starts when you open a file".

Opening External Web Links in Netscape Navigator when Netscape is Already Open (UNIX Based Systems)

If Netscape Navigator is already open and displaying either a local DB2 HTML document or an external Web site, an attempt to open an external Web site from the Information Center will result in a Netscape error. The error will state that "Netscape is unable to find the file or directory named <external site>."

To work around this problem, close the open Netscape browser *before* opening the external Web site. Netscape will restart and bring up the external Web site.

Note that this error does not occur when attempting to open a local DB2 HTML document with Netscape already open.

Problems Starting the Information Center

On some systems, the Information Center can be slow to start if you invoke it using the Start Menu, First Steps, or the **db2ic** command. If you experience this problem, start the Control Center, then select Help —> Information Center.

Stored Procedure Builder

Support for Java Stored Procedures for z/OS or OS/390

In FixPak 7, the following enhancements were added to the Stored Procedure Builder for building interpreted Java stored procedures for DB2 for z/OS or OS/390, Version 7:

- Actual Cost support
- Enhanced error message handling
- Enablement of LINUX/390 servers
- Support for the @ sign in a stored procedure schema name

Compiled Java stored procedures are not supported in any version of z/OS or OS/390 and cannot be created using the Stored Procedure Builder. This is true for all versions of DB2.

Requirements:

- For prerequisites and setup tasks on DB2 for z/OS or OS/390, see APAR PQ52329.
- The Collection ID must match the one used when binding the JDBC driver on z/OS or OS/390.
- Modify the DB2SPB.ini file to include the following entries:

```
SPOPTION_WLM_JAVA_ENVIRONMENT = WLMENVJ
SPOPTION_JAVAPROC_BUILDER = SYSPROC.DSNTJSPP
SPOPTION_BIND_OPTIONS_JAVA = ACT(REP)
SPOPTION_COLLIDJ = DSNJDBC
```

To create a Java stored procedure for z/OS or OS/390 using the Stored Procedure Builder:

1. Open the Inserting Java Stored Procedure wizard:
   a. Under a z/OS or OS/390 database connection in the project tree, right-click the stored procedures folder.
   b. Click **Insert -> Java Stored Procedure Using Wizard**. The wizard opens.
2. Complete the wizard, specifying the z/OS or OS/390 options:
   - The Collection ID must match what was specified on the BIND PACKAGE(collid) when the JDBC drivers were bound on z/OS or OS/390.
   - The default for the Java Package is the procedure name, but you can modify this to any name.

3. Click **OK**. The stored procedure is created and listed in the project tree.
4. Right-click the stored procedure and click **Build**.

Support for SQL Stored Procedures for z/OS or OS/390

In FixPak 7, the Stored Procedure Builder has enhanced use of the ALTER procedures when building stored procedures for DB2 for z/OS or OS/390, Version 7 (APAR JR16764).

To create an SQL stored procedure for z/OS or OS/390 using the Stored Procedure Builder:

1. Open the Inserting SQL Stored Procedure wizard:
   a. Under a z/OS or OS/390 database connection in the project tree, right-click the stored procedures folder.
   b. Click **Insert -> SQL Stored Procedure Using Wizard**. The wizard opens.
2. Complete the wizard, specifying the z/OS or OS/390 options:
   - The Collection ID must match what was specified on the BIND PACKAGE(collid) when the JDBC drivers were bound on z/OS or OS/390.
   - The default for the SQL Package is the procedure name, but you can modify this to any name.
3. Click **OK**. The stored procedure is created and listed in the project tree.
4. Right-click the stored procedure and select **Build**.

Stored Procedure Builder Reference Update to z/OS or OS/390 Documentation

On the "Overview of SQL stored procedures" page of the Stored Procedure Builder online help, the reference to *IBM DB2 Universal Database SQL Procedures Guide and Reference Version 6* is outdated.

For more information about building SQL stored procedures on a z/OS or OS/390 server, you can refer to:
- *DB2 UDB for z/OS or OS/390 SQL Reference*
- *DB2 UDB for z/OS or OS/390 Application Programming and SQL Guide*

Support for Setting Result Set Properties

In FixPak 7, the Stored Procedure Builder has improved performance when running stored procedures that return result sets.

With the Stored Procedure Builder, you can run a stored procedure for testing purposes. Running stored procedures using the Stored Procedure Builder

allows you to test for a successful build to a database and for the presence of a result set. If your stored procedure returns a large result set, you might want to limit the number of rows and columns that display in the result pane.

To edit result set properties for stored procedures:

1. Click **File -> Environment Properties**.
2. Click the **Output** tab in the Environment Properties notebook.
3. To display all rows of a stored procedure result set in the result pane, select the **Display all rows** check box. To limit the number of rows displayed in the result pane, clear the **Display all rows** check box and type a number in the **Number of rows to display** field.
4. To display all of the data in each column of a stored procedure result set in the result pane, select the **Display all data in each column** check box. To limit the column width displayed in the result pane, clear the **Display all data in each column** check box and type a number (representing the number of characters) in the **Maximum column width** field. Data is truncated to fit the specified maximum column width when it is displayed in the result pane.
5. Click **OK** to apply your changes.

Dropping Procedures from a DB2 Database on Windows NT

In previous versions of the Stored Procedure Builder, the DROP PROCEDURE feature did not work properly when dropping procedures from a DB2 database running on Windows NT systems.

In FixPak 7, the Stored Procedure Builder properly drops procedures from a DB2 database running on Windows NT systems.

Wizards

Setting Extent Size in the Create Database Wizard

Using the Create Database Wizard, it is possible to set the Extent Size and Prefetch Size parameters for the User Table Space (but not those for the Catalog or Temporary Tables) of the new database. This feature will be enabled only if at least one container is specified for the User Table Space on the "User Tables" page of the Wizard.

MQSeries Assist Wizard

DB2 Version 7.2 provides a new MQSeries Assist wizard. This wizard creates a table function that reads from an MQSeries queue using the DB2 MQSeries Functions, which are also new in Version 7.2. The wizard can treat each MQSeries message as a delimited string or a fixed length column string depending on your specification. The created table function parses the string according to your specifications, and returns each MQSeries message as a row of the table function. The wizard also allows you to create a view on top of the table function and to preview an MQSeries message and the table function result. This wizard can be launched from Stored Procedure Builder or Data Warehouse Center.

Requirements for this wizard are:
- MQSeries version 5.2
- MQSeries Application Messaging Interface (AMI)
- DB2 MQSeries Functions

For details on these requirements, see "MQSeries" on page 193.

For samples and MQSeries Assist wizard tutorials, go to the tutorials section at http://www.ibm.com/software/data/db2/udb/ide

OLE DB Assist Wizard

This wizard helps you to create a table function that reads data from another database provider that supports the Microsoft OLE DB standard. You can optionally create a DB2 table with the data that is read by the OLE DB table function, and you can create a view for the OLE DB table function. This wizard can be launched from Stored Procedure Builder or Data Warehouse Center.

Requirements for this wizard are:
- An OLE DB provider (such as Oracle, Microsoft SQL Server)
- OLE DB support functions

For samples and OLE DB Assist wizard tutorials, go to the tutorials section at
http://www.ibm.com/software/data/db2/udb/ide

Part 5. Business Intelligence

Business Intelligence Tutorial

Revised Business Intelligence Tutorial

FixPak 2 includes a revised Business Intelligence Tutorial and Data Warehouse Center Sample database which correct various problems that exist in Version 7.1. In order to apply the revised Data Warehouse Center Sample database, you must do the following:

If you have not yet installed the sample databases, create new sample databases using the First Steps launch pad. Click **Start** and select **Programs —> IBM DB2 —> First Steps**.

If you have previously installed the sample databases, drop the sample databases DWCTBC, TBC_MD, and TBC. If you have added any data that you want to keep to the sample databases, back them up before dropping them. To drop the three sample databases:

1. To open the DB2 Command Window, click **Start** and select **Programs —> IBM DB2 —> Command Window**.
2. In the DB2 Command Window, type each of the following three commands, pressing Enter after typing each one:

   ```
   db2 drop database dwctbc
   db2 drop database tbc_md
   db2 drop database tbc
   ```
3. Close the DB2 Command Window.
4. Create new sample databases using the First Steps launch pad. Click **Start** and select **Programs —> IBM DB2 —> First Steps**.

DB2 Universal Database Quick Tour

The Quick Tour is not available on DB2 for Linux or Linux/390.

The Quick Tour is optimized to run with small system fonts. You may have to adjust your Web browser's font size to correctly view the Quick Tour on OS/2. Refer to your Web browser's help for information on adjusting font size. To view the Quick Tour correctly (SBCS only), it is recommended that you use an 8-point Helv font. For Japanese and Korean customers, it is recommended that you use an 8-point Mincho font. When you set font preferences, be sure to select the "Use my default fonts, overriding document-specified fonts" option in the Fonts page of the Preference window.

In some cases the Quick Tour may launch behind a secondary browser window. To correct this problem, close the Quick Tour, and follow the steps in "Error Messages when Attempting to Launch Netscape" on page 10.

When launching the Quick Tour, you may receive a JavaScript error similar to the following:

```
file:/C/Program Files/SQLLIB/doc/html/db2qt/index4e.htm, line 65:

Window is not defined.
```

This JavaScript error prevents the Quick Tour launch page, `index4e.htm`, from closing automatically after the Quick Tour is launched. You can close the Quick Tour launch page by closing the browser window in which `index4e.htm` is displayed.

In the "What's New" section, under the Data Management topic, it is stated that "on-demand log archive support" is supported in Version 7.1. This is not the case. It is also stated that:

```
The size of the log files has been increased from 4GB to 32GB.
```

This sentence should read:

```
The total active log space has been increased from 4GB to 32GB.
```

The section describing the DB2 Data Links Manager contains a sentence that reads:

```
Also, it now supports the use of the Veritas XBSA interface
for backup and restore using NetBackup.
```

This sentence should read:

Also, it now supports the XBSA interface for file archival
and restore. Storage managers that support the XBSA interface
include Legato NetWorker and Veritas NetBackup.

Data Warehouse Center Administration Guide

Update Available

The Data Warehouse Center Administration Guide was updated as part of
FixPak 4. The latest PDF is available for download online at
http://www.ibm.com/software/data/db2/udb/winos2unix/support. The
information in these notes is in addition to the updated reference. All updated
documentation is also available on CD. This CD can be ordered through DB2
service using the PTF number U478862. Information on contacting DB2
Service is available at http://www.ibm.com/cgi-
bin/db2www/data/db2/udb/winos2unix/support/help.d2w/report.

Warehouse Server Enhancements

The following improvements have been made to the warehouse server for
FixPak 5:

Updating configuration parameters
> The server will no longer update critical configuration parameters
> such as service names to an empty string.

Message DWC7906 updated
> Message DWC7906 now contains names of predecessor steps when
> reporting dependency problems.

The following improvements were made to the warehouse server for FixPak 4:

Error on agent shutdown (rc = 7170), secondary rc = 6106.
> This error occurred when the agent was shutdown before the server
> sent a shutdown request. This error was reported unnecessarily and
> will no longer be reported.

System Message and Comment written to log file
> When a user-defined program has finished running, the System
> Message and the Comment will be written to the warehouse log file.
> These messages are now visible from the Work In Progress display
> window.

Incremental commit now works correctly
> If an error occurs when a step is populating a target database and the
> value of Incremental commit is greater than 0, all of the results that
> were committed prior to the error will appear in the target database.
> Prior to FixPak 4, partial results were deleted.

Unable to run warehouse server after changing trace level error corrected
The warehouse server retrieves the name of the logging directory
from the system environment variable, VWS_LOGGING. If
VWS_LOGGING is missing, or points to an invalid directory name,
the TEMP system environment variable is used instead. If TEMP is
missing, or points to an invalid directory name, the logger trace files
are written to c:\. This corrects an error in versions prior to FixPak 4
that was caused by the retrieval of an invalid logging directory name
from the registry.

Additional support for commit commands in stored procedures
The warehouse server sends a commit command to the agent after
user-defined stored procedures have run.

Sample Contents enhanced
The warehouse server no longer has to wait for an agent shutdown
message, so Sample Contents runs more efficiently.

The size of trace log files can now be controlled
You can now control the size of a trace log file using the new system
environment variable, VWS_SERVER_LOG_MAX. If you set the value
of VWS_SERVER_LOG_MAX to greater than 0, the warehouse server
will stop enlarging the log file when it reaches the a size that is
approximately equal to the number of bytes indicated by the value of
VWS_SERVER_LOG_MAX. When the log file reaches the maximum
size, the newest trace log entries are retained and the oldest entries
are overwritten. When doing extensive tracing,
VWS_SERVER_LOG_MAX=150000000 (150M) is a reasonable size.

Using the OS/390 Agent to Run a Trillium Batch System JCL

The OS/390 agent now supports the Trillium Batch System user-defined
program that is created from the Data Warehouse Import Metadata notebook.
Previously, to run a Trillium Batch System JCL file, you had to use the
Windows, AIX, or Solaris Operating Environment agent to run the JCL
remotely. With this update, you can start the JCL with the OS/390 agent.

When you create the Trillium Batch System user-defined program step using
the Import Metadata notebook for the Trillium Batch System, you must select
Remote host as your connection for the OS/390 agent, even when the JCL is
on the same system as the agent. All parameters for the Remote host
connection must be entered.

After you create the Trillium Batch System user-defined program step, use the
Properties notebook of the Trillium Batch System step to change the agent site
to the OS/390 agent site that you want to use.

If the name of the JCL or the output error file contains any blanks or parentheses, you must enclose them in double quotation marks when you enter them into the **Script** or **JCL**, or **Output** error file fields.

Two New Sample Programs in the Data Warehouse Center

Two new sample programs are included with the Data Warehouse Center: EEE_Load and File_Wait. You can use the EEE_Load program to create steps to run the DB2 UDB EEE autoloader program within your data warehousing processes. Use the File_Wait program to create steps that will wait for a file, then run the next step in your process when the file becomes available. For more information about these programs, see the README.UDP file that is located in the ..\SQLLIB\TEMPLATES\SAMPLES directory on the system where the Data Warehouse Center server is installed.

Managing ETI·Extract® Conversion Programs with DB2 Warehouse Manager Updated

The Managing ETI·Extract® Conversion Programs with DB2 Warehouse Manager has been updated and can be downloaded online at http://www.ibm.com/software/data/db2/udb/winos2unix/support.

Importing and Exporting Metadata Using the Common Warehouse Metadata Interchange (CWMI)

Introduction

In addition to the existing support for tag language files, the Data Warehouse Center can now import and export metadata to and from XML files that conform to the Common Warehouse Metamodel (CWM) standard. Importing and exporting these CWM-compliant XML files is referred to as the *Common Warehouse Metadata Interchange (CWMI)*.

You can import and export metadata from the following Data Warehouse Center objects:
- Warehouse sources
- Warehouse targets
- Subject areas, including processes, sources, targets, steps, and cascade relationships.
- User-defined programs

The CWMI import and export utility does not currently support certain kinds of metadata, including: schedules, warehouse schemas, users, and groups.

The Data Warehouse Center creates a log file that contains the results of the import and export processes. Typically, the log file is created in the x:\program files\sqllib\logging directory (where x: is the drive where you installed DB2), or the directory that you specified as the VWS_LOGGING environment variable. The log file is plain text; you can view it with any text editor.

Importing Metadata

You can import metadata either from within Data Warehouse Center, or from the command line.

New objects that are created through the import process are assigned to the default Data Warehouse Center security group. For more information, see "Updating security after importing" in these Release Notes.

If you are importing metadata about a step, multiple files can be associated with the step. Metadata about the step is stored in an XML file, but sometimes a step has associated data stored as BLOBs. The BLOB metadata has the same file name as the XML file, but it is in separate files that have numbered extensions. All of the related step files must be in the same directory when you import.

Updating steps when they are in test or production mode

A step must be in development mode before the Data Warehouse Center can update the step's metadata. If the step is in test or production mode, demote the step to development mode before importing the metadata:
1. Log on to the Data Warehouse Center.
2. Right-click the step that you want to demote, and click **Mode**.
3. Click **Development**.

The step is now in development mode. Change the step back to either test or production mode after you import the metadata.

Importing data from the Data Warehouse Center

You can import metadata from within the Data Warehouse Center:
1. Log on to the Data Warehouse Center.
2. In the left pane, click **Warehouse**.
3. Click **Selected** —> **Import Metadata** —> **Interchange File...**
4. In the Import Metadata window, specify the file name that contains the metadata that you want to import. You can either type the file name or browse for the file.

- If you know the location, type the fully qualified path and file name that you want to import. Be sure to include the **.xml** file extension to specify that you want to import metadata in the XML format or the file will not be processed correctly.
- To browse for your files:
  a. Click the ellipsis (**...**) push button.
  b. In the File window, change **Files of type** to XML.
  c. Go to the correct directory and select the file that you want to import.

     Note: The file must have an **.xml** extension.
  d. Click **OK**.
5. In the **Import Metadata** window, click **OK** to finish. The **Progress** window is displayed while the Data Warehouse Center imports the file.

Using the command line to import metadata

You can also use the command line to import metadata. Here is the import command syntax:

CWMImport *XML_file dwcControlDB dwcUserId dwcPW [PREFIX = DWCtbschema]*

XML_file	The fully qualified path and file name (including the drive and directory) of the XML file that you want to import. This parameter is required.
dwcControlDB	The name of the warehouse control database into which you want to import your metadata. This parameter is required.
dwcUserId	The user ID that you use to connect to the warehouse control database. This parameter is required.
dwcPW	The user password that you use to connect to the warehouse control database. This parameter is required.
[PREFIX=DWCtbschema]	The database schema name for the Data Warehouse Center system tables. If no value for PREFIX= is specified, the default schema name is IWH. This parameter is optional.

Updating Your Metadata After Running the Import Utility
Updating security after importing

As a security measure, the Data Warehouse Center does not import or export passwords. You need to update the passwords on new objects as needed. For more details on import considerations, see the *Data Warehouse Center Administration Guide*, Chapter 12, "Exporting and importing Data Warehouse Center metadata."

When you import metadata, all of the objects are assigned to the *default security group*. You can change the groups who have access to the object:

1. Log on to the Data Warehouse Center.
2. Right-click on the folder that contains the object that you want to change.
3. Click **Properties**, and then click the **Security** tab.
4. Remove groups from the **Selected warehouse groups** list or add groups from **Available warehouse groups** list.
5. Click **OK**.

Exporting Metadata

You can export metadata either from within Data Warehouse Center, or from the command line.

Some steps have metadata that is stored as a BLOB. The BLOB metadata is exported to a separate file that has the same file name as the step's XML file, but with a numbered extension (.1, .2 and so on).

Exporting data from the Data Warehouse Center

You can export metadata from within the Data Warehouse Center:

1. Log on to the Data Warehouse Center.
2. In the left pane, click **Warehouse**.
3. Click **Selected** —> **Export Metadata**—> **Interchange file**.
4. In the Export Metadata window, specify the file name that will contain the exported metadata. You can either enter the file name or browse for the file:
   - If you know the fully qualified path and file name that you want to use, type it in the **File name** entry field. Be sure to include the .xml file extension to specify that you want to export metadata in the XML format.
   - To browse for your files:
     a. Click the ellipsis (...) push button.
     b. In the **File** window, change **Files of type** to XML.
     c. Go to the correct directory and select the file that you want to contain the exported metadata.

> **Note:** Any existing file that you select is overwritten with the exported metadata.

    d. Click **OK**.

5. When the Export Metadata window displays the correct filename, click the object from the **Available objects** list whose metadata you want to export.

6. Click the **>** sign to move the selected object from the **Available objects** list to the **Selected objects** list. Repeat until all of the objects that you want to export are listed in the **Selected objects** list.

7. Click **OK**.

The Data Warehouse Center creates an *input* file, which contains information about the Data Warehouse Center objects that you selected to export, and then exports the metadata about those objects. The progress window is displayed while the Data Warehouse Center is exporting the metadata.

Using the command line to export metadata

Before you can export metadata from the command line, you must first create an *input* file. The input file is a text file with an **.INP** extension, and it lists all of the objects by object type that you want to export. When you export from within the Data Warehouse Center, the input file is created automatically, but to export from the command line you must first create the input file. You can create the input file with any text editor. Type all of the object names as they appear in the Data Warehouse Center. Make sure you create the file in a read/write directory. When you run the export utility, the Data Warehouse Center writes the XML files to the same directory where the input file is.

Here's a sample input file:

```
<PROC>
Tutorial Fact Table Process
<IR>
Tutorial file source
Tutorial target
<UDP>
New Program group
```

In the <PROC> (processes) section, list all of the processes that you want to export. In the <IR> (information resources) section, list all the warehouse sources and targets that you want to export. The Data Warehouse Center automatically includes the tables and columns that are associated with these sources and targets. In the <UDP> (user defined programs) section, list all the program groups that you want to export.

To export metadata, enter the following command at a DOS command prompt:

```
CWMExport INPcontrol_file dwcControlDB dwcUserID dwcPW
[PREFIX=DWCtbschema]
```

INPcontrol_file	The fully qualified path and file name (including the drive and directory) of the .INP file that contains the objects that you want to export. This parameter is required.
dwcControlDB	The name of the warehouse control database that you want to export from. This parameter is required.
dwcUserID	The user ID that you use to connect to the warehouse control database. This parameter is required.
dwcPW	The password that you use to connect to the warehouse control database. This parameter is required.
[PREFIX=DWCtbschema]	The database schema name for the Data Warehouse Center system tables. If no value for PREFIX= is specified, the default value is IWH. This parameter is optional.

Tag Language Metadata Import/Export Utility

Key Definitions

The primary and foreign keys defined in tag language files are ignored if they are the same as those already defined in the control database. An error occurs if the keys are different from those already defined.

Step and Process Schedules

Step and process schedules are no longer deleted by the import utility. Schedules defined in a tag file are now added to the current list of schedules. This may cause duplicate schedules to appear. Duplicate schedules should be deleted by the user before steps are promoted to production mode.

SAP Step Information

Possible to Create Logically Inconsistent Table

If all of the following conditions are met, the resulting target table may not be logically consistent.

1. The BO has GetList and GetDetail export parameters and you have mapped all key fields.
2. On the Output Parameters page of the Properties notebook for the SAP step, you select a GetList export parameter whose SAP parameter name differs from the one used for the parameter mapping.

> **Note:** SAP parameter name refers to the part of the parameter appearing before the period in the fully qualified name. For example, for the parameter, DocList.DOCNUMBER, "DocList" is the SAP parameter name.

3. On the Output Parameters page of the Properties notebook for the SAP step, you select the GetDetail export parameter.

Example:

DocumentNumber is a key field. DocList.DOCNUMBER and DocNumberSelection.OPTION are GetList export parameters. DocData.USERNAME is a GetDetail export parameter.

You map DocumentNumber to DocList.DOCNUMBER. (Condition 1)

You select DocNumberSelection.OPTION as an output parameter. (Condition 2, since DocNumberSelection and DocList are different SAP parameter names.)

You select DocData.USERNAME as an output parameter. (Condition 3, since it is a GetDetail export parameter.)

These conditions result in a target table whose column sources are GetList and GetDetail parameters. The logical consistency of the relationship between the columns, however, is not assured.

SAP Connector Information

SAP Connector Installation Restrictions

The SAP Connector only supports English-language installations of the SAP R/3 system.

Performance of GetDetail BAPI

If GetDetail has a large number of input parameters, GetDetail BAPI performance is slow.

Web Connector Information

Supported WebSphere Site Analyzer Versions

Web Connector only supports WebSphere Site Analyzer Version 4.0. It does not support Version 4.1 at this time.

DB2 OLAP Starter Kit

The IBM DB2 OLAP Starter Kit 7.2 adds support for Oracle, MS-SQL, Sybase, and Informix relational database management systems (RDBMSs) on certain operating system platforms. Version 7.2 contains scripts and tools for all supported RDBMSs, including DB2. There are some restrictions; see "Known Problems and Limitations" on page 261 for more information.

The service level of DB2 OLAP Starter Kit for DB2 Universal Database Version 7.2 is the equivalent of patch 2 for Hyperion Essbase 6.1 plus patch 2 for Hyperion Integration Server 2.0.

OLAP Server Web Site

For the latest installation and usage tips for the DB2 OLAP Starter Kit, check the Library page of the DB2 OLAP Server Web site:

http://www.ibm.com/software/data/db2/db2olap/library.html

Supported Operating System Service Levels

The server components of the OLAP Starter Kit for Version 7.2 support the following operating systems and service levels:

- Windows NT 4.0 servers with SP 5 and Windows 2000
- AIX version 4.3.3 or higher
- Solaris Operating System version 2.6, 7, and 8 (Sun OS 5.6, 5.7, or 5.8)

The client components run on Windows 95, Windows 98, Windows NT 4.0 SP5, and Windows 2000.

Completing the DB2 OLAP Starter Kit Setup on UNIX

The DB2 OLAP Starter Kit install follows the basic procedures of the DB2 Universal Database install for UNIX. The product files are laid down by the installation program to a system directory: (for AIX: `/usr/lpp/db2_07_01`; for Solaris Operating Environment: `/opt/IBMdb2/V7.1`).

Then during the instance creation phase, two DB2 OLAP directories are created (`essbase` and `is`) within the instance user's home directory under `sqllib`. Only one instance of OLAP server can run on a machine at a time. To complete the set up, the user must manually set the `is/bin` directory so that it

is not a link to the is/bin directory in the system. It should link to a writable directory within the instance's home directory.

To complete the setup for the Solaris Operating Environment, logon using the instance ID, change to the sqllib/is directory, then enter the following:

```
rm bin
mkdir bin
cd bin
ln -s /opt/IBMdb2/V7.1/is/bin/ismesg.mdb ismesg.mdb
ln -s /opt/IBMdb2/V7.1/is/bin/olapicmd olapicmd
ln -s /opt/IBMdb2/V7.1/is/bin/olapisvr olapisvr
ln -s /opt/IBMdb2/V7.1/is/bin/essbase.mdb essbase.mdb
ln -s /opt/IBMdb2/V7.1/is/bin/libolapams.so libolapams.so
```

Additional Configuration for the Solaris Operating Environment

In the Solaris Operating Environment, you might encounter errors if the OLAP Starter Kit is not linked to the appropriate ODBC driver. To prevent these errors, run the following command, which creates a link in $ARBORPATH/bin to point to the OLAP driver sqllib/lib/libdb2.so:

```
ln -s $HOME/sqllib/lib/libdb2.so  libodbcinst.so
```

Additional Configuration for All Operating Systems

Starting in FixPak 3 of DB2 Universal database Version 7, the DB2 OLAP Starter Kit includes functions that require Java. After installing FixPak 3 or later, you might see the following error message on the OLAP Server console:

```
Can not find [directory] [/export/home/arbor7sk/sqllib/essbase/java/],
required to load JVM.
```

To correct this error, take the following steps:

1. Log on as the DB2 instance owner.
2. Find the directory in which you installed the DB2 OLAP Starter Kit. The default name for this directory is essbase.
3. In the essbase directory, create a subdirectory called java.
4. In the java subdirectory, create the following empty files:
   - essbase.jar
   - essdefs.dtd
   - jaxp.jar
   - parser.jar
   - udf.policy

Configuring ODBC for the OLAP Starter Kit

IBM DB2 OLAP Starer Kit 7.2 requires an ODBC.ini file for operation of Open Database Connectivity (ODBC) connections from OLAP Integration Server to the relational data source and to the OLAP Metadata Catalog.

- On Windows systems, this file is in the Registry under `HKEY_LOCAL_MACHINE/SOFTWARE/ODBC`. Use ODBC Data Source Administrator to store information about how to connect to a relational data source.
- On UNIX systems, the installation program creates a model `odbc.ini` file. To store information about how to connect to a relational data source, edit the file using your preferred editor.

The ODBC.ini file is available in ODBC software packages and is included with Microsoft Office software. Additional information about applications that install ODBC drivers or the ODBC Administrator is available at the following web site: `http://support.microsoft.com/`.

For Oracle users on AIX machines: To configure ODBC for Oracle, you must update the `ODBC.ini` file to point to the MERANT 3.6 drivers.

In Version 7.2, the OLAP Starter Kit manages ODBC connections to the relational data source and to the OLAP Metadata Catalog. To accommodate these ODBC connections, the OLAP Starter Kit uses ODBC drivers on Windows NT 4.0, Windows 2000, AIX, and Solaris systems.

- DB2 Universal Database Version 6 Database Client: DB2 Version 6 ODBC drivers on Windows NT 4.0 SP5 or Windows 2000, AIX 4.3.3, and Solaris Operating System 2.6, 7, or 8 (Sun OS 5.6, 5.7, or 5.8).
- DB2 Universal Database 7.1 Database Client: DB2 Version 7 ODBC drivers on Windows NT 4.0 SP5 or Windows 2000, AIX 4.3.3, and Solaris Operating System 2.6, 7, or 8 (Sun OS 5.6, 5.7, or 5.8).
- Oracle 8.04 and 8i SQL*Net 8.0 Database Client: MERANT 3.6 ODBC drivers on Windows NT 4.0 SP5 or Windows 2000, AIX 4.3.3, Solaris Operating System 2.6, 7 or 8 (Sun OS 5.6, 5.7, or 5.8).
- MS SQL Server 6.5.201 (no Database Client required): MS SQL Server 6.5 ODBC drivers on Windows NT 4.0 SP5 or Windows 2000.
- MS SQL Server 7.0 (no Database Client required): MS SQL Server 7.0 ODBC drivers on Windows NT 4.0 SP5 or Windows 2000.

Configuring Data Sources on UNIX Systems

On AIX and Solaris systems, you must manually set environment variables for ODBC and edit the `odbc.ini` file to configure the relational data source and OLAP Metadata Catalog. Make sure you edit the `odbc.ini` file if you add a new driver or data source or if you change the driver or data source.

If you will be using the DB2 OLAP Starter Kit on AIX or Solaris systems to access Merant ODBC sources and DB2 databases, change the value of the "Driver=" attribute in the DB2 source section of the .odbc.ini file as follows:

AIX: The Driver name is /usr/lpp/db2_07_01/lib/db2_36.o

Sample ODBC source entry for AIX:
```
[SAMPLE] Driver=/usr/lpp/db2_07_01/lib/db2_36.o
Description=DB2 ODBC Database
Database=SAMPLE
```

Solaris Operating Environment: The Driver name is /opt/IBMdb2/V7.1/lib/libdb2_36.so

Sample ODBC source entry for Solaris Operating Environment:
```
[SAMPLE] Driver=/opt/IBMdb2/V7.1/lib/libdb2_36.so
Description=DB2 ODBC Database
Database=SAMPLE
```

Configuring ODBC Environment Variables
On UNIX systems, you must set environment variables to enable access to ODBC core components. The is.sh and is.csh shell scripts that set the required variables are provided in the Starter Kit home directory. You must run one of these scripts before using ODBC to connect to data sources. You should include these scripts in the login script for the user name you use to run the OLAP Starter Kit.

Editing the odbc.ini File
To configure a data source in an odbc.ini file, you must add a name and description for the ODBC data source and provide the ODBC driver path, file name, and other driver settings in a separate section that you create for the data source name. The installation program installs a sample odbc.ini file in the ISHOME directory. The file contains generic ODBC connection and configuration information for supported ODBC drivers. Use the file as a starting point to map the ODBC drivers that you use to the relational data source and OLAP Metadata Catalog.

If you use a different file than the odbc.ini file, be sure to set the ODBCINI environment variable to the name of the file you use.

Adding a Data Source to an odbc.ini File
1. On the system running the OLAP Starter Kit servers, open the odbc.ini file by using a text editor such as **vi**.
2. Find the section starting with [ODBC Data Sources] and add a new line with the data source name and description, such as: mydata=data source

for analysis. To minimize confusion, the name of the data source should match the name of the database in the RDBMS.

3. Add a new section to the file by creating a new line with the name of the new data source enclosed in brackets, such as: [mydata].

4. On the lines following the data source name, add the full path and file name for the ODBC driver required for this data source and any other required ODBC driver information. Use the examples shown in the following sections as a guideline to map to the data source on your RDBMS. Make sure that the ODBC driver file actually exists in the location you specify for the `Driver=` setting.

5. When you have finished editing `odbc.ini`, save the file and exit the text editor.

Example of ODBC Settings for DB2

The following example shows how you might edit `odbc.ini` to connect to a relational data source, db2data, on DB2 Universal Database Version 6.1 on AIX, using an IBM DB2 native ODBC driver. In the **vi** editor, use the **$ODBCINI** command to edit the `odbc.ini` and insert the following statements:

```
[ODBC Data Sources]
db2data=DB2 Source Data on AIX

...
[db2data]
Driver=/home/db2inst1/sqllib/lib/db2.o
Description=DB2 Data Source - AIX, native
```

Example of ODBC Settings for Oracle

Here is an example of how you might edit `odbc.ini` to connect to a relational data source, oradata, on Oracle Version 8 (on Solaris Operating Environment), using a MERANT Version 3.6 ODBC driver. In this example, `LogonID` and `Password` are overridden with the actual values used in the OLAP Starter Kit user name and password.

```
[ODBC Data Sources]
oradata=Oracle8 Source Data on Solaris

...
[myoracle] Driver=
/export/home/users/dkendric/is200/odbclib/ARor815.so
Description=my oracle source
```

Configuring the OLAP Metadata Catalog on UNIX Systems

Configuring an OLAP Metadata Catalog on AIX and Solaris systems is similar to configuring a data source. For the OLAP Metadata Catalog database, add a data source name and section to the `odbc.ini` file, as described in "Editing the odbc.ini File" on page 254. No other changes are required.

You must create an OLAP Metadata Catalog database in a supported RDBMS before configuring it as an ODBC data source.

Here is an example how you might edit odbc.ini to connect to the OLAP Metadata Catalog, TBC_MD, on DB2 Version 6.1 (on Solaris Operating Environment), using a native ODBC driver:

```
[ODBC Data Sources]
ocd6a5a=db2 v6
...
[ocd6a5a]
Driver=/home/db2instl/sqllib/lib/db2.0
Description=db2
```

Configuring Data Sources on Windows Systems

To configure a relational data source on Windows NT or Windows 2000 systems, you must start ODBC Administrator and then create a connection to the data source that you will use for creating OLAP models and metaoutlines. Run the ODBC Administrator utility from the Windows Control Panel. The following example creates a DB2 data source; the dialog boxes for other RDBMSs will differ.

To configure a relational data source with ODBC Administrator, complete the following steps:

1. On the Windows desktop, open the Control Panel window.
2. In the Control Panel window, perform one of the following steps:
   a. On Windows NT, double-click the **ODBC** icon to open the ODBC Data Source Administrator dialog box.
   b. On Windows 2000, double-click the **Administrative Tools** icon, and then double-click the **Data Sources (ODBC)** icon to open the ODBC Data Source Administrator dialog box.
3. In the ODBC Data Source Administrator dialog box, click the **System DSN** tab.
4. Click **Add** to open the Create New Data Source dialog box.
5. In the driver list box of the Create New Data Source dialog box of ODBC Administrator, select an appropriate driver, such as IBM DB2 ODBC Driver, and click **Finish** to open the ODBC IBMDB2 Driver - Add dialog box.
6. In the ODBC IBM DB2 Driver - Add dialog box, in the **Database alias** drop-down list, select the name of the database for your relational source data (for example, TBC in the sample application).
7. In the **Description** text box, type an optional description that indicates how you use this driver and click **Add**. For example, type the following words to describe the My Business database:

Customers, products, markets

You might type the following words to describe the sample application database:

Sample relational data source

The descriptions help to identify the available data sources for your selection when you connect from OLAP Starter Kit Desktop.

8. Click OK to return to the ODBC Data Source Administrator dialog box. The data source name you entered and the driver you mapped to it are displayed in the **System Data Sources** list box on the **System DSN** tab.

To edit configuration information for a data source:

1. Select the data source name and click **Configure** to open the ODBC IBM DB2 - Add dialog box.
2. Correct any information you want to change.
3. Click **OK** twice to exit.

Configuring the OLAP Metadata Catalog on Windows Systems

To configure an OLAP Metadata Catalog on Windows NT or Windows 2000, start ODBC Administrator and then create a connection to the data source that contains the OLAP Metadata Catalog database.

The following example creates a DB2 data source; dialog boxes for other RDBMSs will differ. To create a data source for the OLAP Metadata Catalog, complete the following steps:

1. On the desktop, open the Control Panel window.
2. In the Control Panel window, perform one of the following steps:
   a. On Windows NT, double-click the **ODBC** icon to open the ODBC Data Source Administrator dialog box.
   b. On Windows 2000, double-click the **Administrative Tools** icon, and then double-click the **Data Sources (ODBC)** icon to open the ODBC Data Source Administrator dialog box.
3. In the ODBC Data Source Administrator dialog box, click the **System DSN** tab.
4. Click **Add** to open the Create New Data Source dialog box.
5. In the driver list box of the Create New Data Source dialog box of ODBC Administrator, select an appropriate driver, such as IBM DB2 ODBC Driver, and click **Finish** to open the ODBC IBMDB2 Driver - Add dialog box.
6. In the ODBC IBM DB2 Driver - Add dialog box, in the **Database alias** drop-down list, select the name of the database for your OLAP Metadata Catalog (for example, TBC_MD in the sample application). The name of the selected database is automatically displayed in the **Data Source Name** text box.
7. If you want to change the name of the data source, select the name displayed in the **Data Source Name** text box, type a new name to indicate how you use this driver, and click **Add**. For example, you might type the following name to indicate that you are using the driver to connect to the first OLAP Metadata Catalog:

```
OLAP Catalog first
```

You would type the following name to indicate that you are connecting to the sample application OLAP Metadata Catalog database:
```
TBC_MD
```

8. In the **Description** text box, enter a description that indicates how you use this driver. For example, you might type the following words to describe the OLAP Metadata Catalog:
```
My first models and metaoutlines
```

You might type the following words to describe the sample application OLAP Metadata Catalog database:
```
Sample models and metaoutlines
```

The descriptions help you to identify the catalog that you want to select when you connect to the OLAP Metadata Catalog from the OLAP Starter Kit Desktop.

9. Click OK to return to the ODBC Data Source Administrator dialog box. The data source name you entered and the driver you mapped to it are displayed in the **System Data Sources** list box on the **System DSN** tab.

To edit configuration information for a data source:

1. Select the data source name and click **Configure** to open the ODBC IBM DB2 - Add dialog box.
2. Correct any information you want to change.
3. Click **OK** twice to exit.

After You Configure a Data Source

After you configure the relational data source and OLAP Metadata Catalog, you can connect to them from the OLAP Starter Kit. You can then create, modify, and save OLAP models and metaoutlines.

The SQL Server ODBC driver may time out during a call to an SQL Server database. Try again when the database is not busy. Increasing the driver time-out period may avoid this problem. For more information, see the ODBC documentation for the driver you are using.

For more information on ODBC connection problems and solutions, see the *OLAP Integration Server System Administrator's Guide*.

Logging in from OLAP Starter Kit Desktop

To use the OLAP Starter Kit Desktop to create OLAP models and metaoutlines, you must connect the client software to two server components: DB2 OLAP Integration Server and DB2 OLAP Server. The login dialog prompts you for the necessary information for the Desktop to connect to these two servers. On the left side of the dialog, enter information about DB2 OLAP Integration Server. On the right side, enter information about DB2 OLAP Server.

To connect to DB2 OLAP Integration Server:

- Server: Enter the host name or IP address of your Integration Server. If you have installed the Integration Server on the same workstation as your desktop, then typical values are "localhost" or "127.0.0.1".
- OLAP Metadata Catalog: When you connect to OLAP Integration Server you must also specify a Metadata Catalog. OLAP Integration Server stores information about the OLAP models and metaoutlines you create in a relational database known as the Metadata Catalog. This relational database must be registered for ODBC. The catalog database contains a special set of relational tables that OLAP Integration Server recognizes. On the login dialog, you can specify an Integration Server and then expand the pull-down menu for the OLAP Metadata Catalog field to see a list of the ODBC data source names known to the OLAP Integration Server. Choose an ODBC database that contains the metadata catalog tables.
- User Name and Password: OLAP Integration Server will connect to the Metadata Catalog using the User name and password that you specify on this panel. This is a login account that exists on the server (not the client, unless the server and client are running on the same machine). The user name must be the user who created the OLAP Metadata Catalog. Otherwise, OLAP Integration Server will not find the relational tables in the catalog database because the table schema names are different.

The DB2 OLAP Server information is optional, so the input fields on the right side of the Login dialog may be left blank. However, some operations in the Desktop and the Administration Manager require that you connect to a DB2 OLAP Server. If you leave these fields blank, then the Desktop will display the Login dialog again if the Integration Server needs to connect to DB2 OLAP Server in order to complete an operation that you requested. It is recommended that you always fill in the DB2 OLAP Server fields on the Login dialog.

To connect to DB2 OLAP Server:

- Server: Enter the host name or IP address of your DB2 OLAP Server. If you are running the OLAP Starter Kit, then your OLAP Server and Integration

Server are the same. If the Integration Server and OLAP Server are installed on different hosts, then enter the host name or an IP address that is defined on OLAP Integration Server.

- User Name and Password: OLAP Integration Server will connect to DB2 OLAP Server using the user name and password that you specify on this panel. This user name and password must already be defined to the DB2 OLAP Server. OLAP Server manages its own user names and passwords separately from the host operating system.

Starter Kit Login Example

The following example assumes that you created the OLAP Sample, and you selected *db2admin* as your administrator user ID, and *password* as your administrator password during OLAP Starter Kit installation.

- For OLAP Integration Server: Server is *localhost*, OLAP Metadata Catalog is *TBC_MD*, User Name is *db2admin*, Password is *password*
- For DB2 OLAP Server: Server is *localhost*, User Name is *db2admin*

Manually Creating and Configuring the Sample Databases for OLAP Starter Kit

The sample databases are created automatically when you install OLAP Starter Kit. The following instructions explain how to setup the Catalog and Sample databases manually, if necessary.

1. In Windows, open the Command Center window by clicking **Start ⟶Programs⟶DB2 for Windows NT⟶ Command Window.**
2. Create the production catalog database:
   a. Type db2 create db OLAP_CAT
   b. Type db2 connect to OLAP_CAT
3. Create tables in the database:
   a. Navigate to \SQLLIB\IS\ocscript\ocdb2.sql
   b. Type db2 -tf ocdb2.sql
4. Create the sample source database:
   a. Type db2 connect reset
   b. Type db2 create db TBC
   c. Type db2 connect to TBC
5. Create tables in the database:
   a. Navigate to \SQLLIB\IS\samples\
   b. Copy tbcdb2.sql to \SQLLIB\samples\db2sampl\tbc
   c. Copy lddb2.sql to \SQLLIB\samples\db2sampl\tbc
   d. Navigate to \SQLLIB\samples\db2sampl\tbc
   e. Type db2 -tf tbcdb2.sql
   f. Type db2 - vf lddb2.sql to load sample source data into the tables.

6. Create the sample catalog database:
   a. Type `db2 connect reset`
   b. Type `db2 create db TBC_MD`
   c. Type `db2 connect to TBC_MD`
7. Create tables in the database:
   a. Navigate to \SQLLIB\IS\samples\tbc_md
   b. Copy ocdb2.sql to \SQLLIB\samples\db2sampl\tbcmd
   c. Copy lcdb2.sql to \SQLLIB\samples\db2sampl\tbcmd
   d. Navigate to \SQLLIB\samples\db2sampl\tbcmd
   e. Type `db2 -tf ocdb2.sql`
   f. Type `db2 -vf lcdb2.sql` to load sample metadata into the tables.
8. Configure ODBC for TBC_MD, TBC, AND OLAP_CAT:
   a. Open the NT control panel by clicking **Start→Settings→Control Panel**
   b. Select ODBC (or ODBC data sources) from the list.
   c. Select the System DSM tab.
   d. Click **Add**. The **Create New Data Source** window opens.
   e. Select IBM DB2 ODBC DRIVER from the list.
   f. Click **Finish.** The ODBC IBM D2 Driver - Add window opens.
   g. Type the name of the data source (OLAP_CAT) in the **Data source name** field.
   h. Type the alias name in the **Database alias** field, or click the down arrow and select OLAP_CAT from the list.
   i. Click OK.
   j. Repeat these steps for the TBC_MD and the TBC databases.

Migrating Applications to OLAP Starter Kit Version 7.2

The installation program does not reinstall the OLAP Starter Kit sample applications, databases, and data files. Your existing applications and databases are not affected in any way. However, it is always a good idea to back up your applications and databases before an installation.

Your applications are automatically migrated to Version 7.2 when you open them.

Known Problems and Limitations

This section lists known limitations for DB2 OLAP Starter Kit.

Informix RDBMS Compatibility with Merant Drivers for Windows Platforms

In order for the Merant drivers for Windows platforms to work with the Informix RDBMS, the following two entries must be added to the PATH statement:

- C:\Informix
- C:\Informix\bin

Both entries must be at the beginning of the PATH.

Possible Inconsistency Between Dimensions in OLAP Models and Associated Metaoutlines

Under certain conditions, you can create a dimension in a metaoutline that has no corresponding dimension in the OLAP model. This can occur in the following scenario:

1. Create a new OLAP model and save it.
2. Create a metaoutline based on the model but do not save the metaoutline.
3. Return to the OLAP model and delete a dimension on which one of the metaoutline dimensions is based.
4. Return to the metaoutline, save it, close it, and reopen it. The metaoutline will contain a dimension that does not have a corresponding dimension in the OLAP model.

The OLAP Starter Kit cannot distinguish between an inconsistent dimension created in this manner and a user-defined dimension in a metaoutline. Consequently, the inconsistent dimension will be displayed in the metaoutline, but the metaoutline regards it as a user-defined dimension since no corresponding dimension exists in the OLAP model.

On Windows 2000 Platforms, the Environment Variable Setting for TMP Causes Member and Data Loads to Fail

Because of a difference in the default system and user environment variable settings for TMP between Windows 2000 and Windows NT, member and data loads fail when the OLAP Starter Kit is running on Windows 2000 platforms. The resulting error message tells users that the temp file could not be created. You can work around this limitation on Windows 2000 by taking the following steps:

1. Create a directory named C:\TEMP
2. Set the environment variable TMP for both the system and the user to TMP=C:\TEMP

Installation of ODBC Does Not Replace Existing Merant Driver

The existing 3.6 Merant ODBC drivers will not be updated with this

installation. If you are upgrading from the OLAP Starter Kit Version 7.1, fixpack 2 or earlier, you should continue using the previously-installed ODBC drivers

Using Merant Informix ODBC Drives on UNIX Platforms

To use the Merant Informix ODBC drivers on UNIX platforms, you must do one of the following:

- Before starting the Starter Kit, set the LANG environment variable to "en_US". For example, for korn shell, type:

```
export LANG='en_US'
```

   Set this variable every time you start the OLAP Starter Kit.
- If your LANG environment variable is already set to a different value, make the following symbolic link after installation:

```
ln -s $ISHOME/locale/en_US $ISHOME/locale/$LANG
```

Mixing service levels of OLAP clients and servers

IBM recommends that you keep both client and server components of the DB2 OLAP Starter Kit at the same version and fixpack level. But in some situations, you might be able to mix different service levels of client and server components:

Using clients and servers at different service levels within a version

IBM does not support, and recommends against, using newer clients with older servers. However, you might be able to use older clients with newer servers, although IBM does not support it. You might experience some problems. For example:

- Messages from the server might be incorrect. You can work around this problem by upgrading the message.MDB file on the client to match the level on the server.
- New server features do not work. The client, server, or both may fail when you attempt to use a new feature.
- The client might not connect properly with the server.

Using multiple servers with a single client within a version

If you need to connect a client to several OLAP servers on different machines or operating systems, IBM recommends that you make them all the same version and service level. Your client should at least be at the same as the lowest level server. If you experience problems, you might need to use different client machines to match up with the appropriate host, or upgrade all clients and servers to the same service level.

Mixing clients and servers from different versions

IBM does not support using OLAP Starter Kit clients and servers from Version 7.1 with clients and servers from Version

Mixing IBM products (DB2 OLAP Starter Kit) with Hyperion
products (Hyperion Essbase and Hyperion Integration Server)
IBM does not support mixing OLAP clients and servers from
IBM with OLAP clients and servers from Hyperion Solutions.
There are some differences in feature that may cause
problems, even though mixing these components might work
in some situations.

OLAP Spreadsheet Add-in EQD Files Missing

In the DB2 OLAP Starter Kit, the Spreadsheet add-in has a component called
the Query Designer (EQD). The online help menu for EQD includes a button
called **Tutorial** that does not display anything. The material that should be
displayed in the EQD tutorials is a subset of chapter two of the *OLAP
Spreadsheet Add-in User's Guide for Excel*, and the *OLAP Spreadsheet Add-in
User's Guide for 1-2-3*. All the information in the EQD tutorial is available in
the HTML versions of these books in the Information Center, and in the PDF
versions.

Information Catalog Manager Administration Guide

Information Catalog Manager Initialization Utility

With the Initialize Information Catalog Manager (ICM) utility, you can now append an SQL statement to the end of the CREATE TABLE statement using the following command:

```
CREATEIC \DBTYPE dbtype \DGNAME dgname \USERID userid \PASSWORD password
\KA1 userid \TABOPT "directory:\tabopt.file"
```

You can specify the TABOPT keyword in the CREATEIC utility from the directory where DB2 is installed. The value following the TABOPT keyword is the tabopt.file file name with the full path. If the directory name contains blanks, enclose the name with quotation marks. The contents of the *tabopt.file* file must contain information to append to the CREATE TABLE statement. You can use any of the SQL statements below to write to this *tabopt.file* file. The ICM utility will read this file and then append it to the CREATE TABLE statement.

Table 9. SQL statements

IN MYTABLESPACE	Creates a table with its data in MYTABLESPACE
DATA CAPTURE CHANGES	Creates a table and logs SQL changes in an extended format
IN ACCOUNTING INDEX IN ACCOUNT_IDX	Creates a table with its data in ACCOUNTING and its index in ACCOUNT_IDX

The maximum size of the content file is 1000 single-byte characters.

This new capability is available only on Windows and UNIX systems.

Licensing issues

If you get the following message:

```
FLG0083E: You do not have a valid license for the IBM
Information Catalog Manager Initialization utility.
Please contact your local software reseller or IBM
marketing representative.
```

You must purchase the DB2 Warehouse Manager or the IBM DB2 OLAP Server and install the Information Catalog Manager component, which includes the Information Catalog Initialization utility.

Installation Issues

If you installed the DB2 Warehouse Manager or IBM DB2 OLAP Server and then installed another Information Catalog Manager Administrator component (using the DB2 Universal Database CD-ROM) on the same workstation, you might have overwritten the Information Catalog Initialization utility. In that case, from the \sqllib\bin directory, find the files `createic.bak` and `flgnmwcr.bak` and rename them to `createic.exe` and `flgnmwcr.exe` respectively.

If you install additional Information Catalog Manager components from DB2 Universal Database, the components must be on a separate workstation from where you installed the Data Warehouse Manager. For more information, see Chapter 3, Installing Information Catalog Manager components, in the DB2 Warehouse Manager Installation Guide.

Enhancement to Information Catalog Manager

Information Catalog Manager includes the following enhancements:

ICM now supports the import of ETI filter information for the source or target database, table or column. Upon Register to Warehouse, a new ICM object type ETI Conversion Data is used to store the filter information. These objects are then linked to the source or target database, table or column for which it was defined.

ICM has the ability to link a particular source or target database, table or column with multiple ETI Conversion Data objects as the result of registering different ETI Conversions to the same ICM catalog.

The same ability applies to Transformations in that a particular target column can now contain multiple Tranformations as the result of registering different ETI Conversions to the same ICM catalog. To do this, ICM made changes to the Transformation key when importing a ETI*Extract mapping.

To enable these features, use ETI*Extract 4.2.1 with MetaScheduler 4.1.0 to register with Data Warehouse Manager. More information on enabling these features is available in the Hints and Tips section of DB2 Warehouse Manager at http://www.ibm.com/software/data/db2/datawarehouse/support.html. Search on the keywords "ETI" or "Application Data".

Incompatibility between Information Catalog Manager and Sybase in the Windows Environment

The installation of Information Catalog Manager (ICM) Version 7 on the same Windows NT or Windows 2000 machine with Sybase Open Client results in an error, and the Sybase Utilities stops working. An error message similar to the following occurs:

```
Fail to initialize LIBTCL.DLL.  Please make sure the SYBASE environment
variable is set correctly.
```

Avoid this scenario by removing the environment parameter LC_ALL from the Windows Environment parameters. LC_ALL is a locale category parameter. Locale categories are manifest constants used by the localization routines to specify which portion of the locale information for a program to use. The *locale* refers to the locality (country/region) for which certain aspects of your program can be customized. Locale-dependent areas include, for example, the formatting of dates or the display format for monetary values. LC_ALL affects all locale-specific behavior (all categories).

If you remove the LC_ALL environment parameter so that ICM can coexist with Sybase on the Windows NT platform, the following facilities no longer work:

- Information Catalog User
- Information Catalog Administrator
- Information Catalog Manager

The removal of the LC_ALL parameter will not affect anything other than ICM.

Accessing DB2 Version 5 Information Catalogs with the DB2 Version 7 Information Catalog Manager

The DB2 Version 7 Information Catalog Manager subcomponents, as configured by the DB2 Version 7 install process, support access to information catalogs stored in DB2 Version 6 and DB2 Version 7 databases. You can modify the configuration of the subcomponents to access information catalogs that are stored in DB2 Version 5 databases. The DB2 Version 7 Information Catalog Manager subcomponents do not support access to data from DB2 Version 2 or any other previous versions.

To set up the Information Catalog Administrator, the Information Catalog User, and the Information Catalog Initialization Utility to access information catalogs that are stored in DB2 Version 5 databases:

1. Install DB2 Connect Enterprise Edition Version 6 on a workstation other than where the DB2 Version 7 Information Catalog Manager is installed.

DB2 Connect Enterprise Edition is included as part of DB2 Universal Database Enterprise Edition and DB2 Universal Database Enterprise - Extended Edition. If Version 6 of either of these DB2 products is installed, you do not need to install DB2 Connect separately.

Restriction: You cannot install multiple versions of DB2 on the same Windows NT or OS/2 workstation. You can install DB2 Connect on another Windows NT workstation or on an OS/2 or UNIX workstation.

2. Configure the Information Catalog Manager and DB2 Connect Version 6 for access to the DB2 Version 5 data. For more information, see the *DB2 Connect User's Guide*. The following steps are an overview of the steps that are required:

   a. On the DB2 Version 5 system, use the DB2 Command Line Processor to catalog the Version 5 database that the Information Catalog Manager is to access.

   b. On the DB2 Connect system, use the DB2 Command Line Processor to catalog:
      - The TCP/IP node for the DB2 Version 5 system
      - The database for the DB2 Version 5 system
      - The DCS entry for the DB2 Version 5 system

   c. On the workstation with the Information Catalog Manager, use the DB2 Command Line Processor to catalog:
      - The TCP/IP node for the DB2 Connect system
      - The database for the DB2 Connect system

   For information about cataloging databases, see the *DB2 Universal Database Installation and Configuration Supplement*.

3. At the warehouse with the Information Catalog Manager, bind the DB2 CLI package to each database that is to be accessed through DB2 Connect.

   The following DB2 commands give an example of binding to v5database, a hypothetical DB2 version 5 database. Use the DB2 Command Line Processor to issue the following commands. db2cli.lst and db2ajgrt are located in the \sqllib\bnd directory.

   ```
   db2 connect to v5database user userid using password
   db2 bind db2ajgrt.bnd
   db2 bind @db2cli.lst blocking all grant public
   ```

   where *userid* is the user ID for v5database and *password* is the password for the user ID.

An error occurs when db2cli.list is bound to the DB2 Version 5 database.
This error occurs because large objects (LOBs) are not supported in this
configuration. This error will not affect the warehouse agent's access to the
DB2 Version 5 database.

FixPak 14 for DB2 Universal Database Version 5, which is available in
June, 2000, is required for accessing DB2 Version 5 data through DB2
Connect. Refer to APAR number JR14507 in that FixPak.

Setting up an Information Catalog

Step 2 in the first section of Chapter 1, "Setting up an information catalog",
says:

```
When you install either the DB2 Warehouse Manager
or the DB2 OLAP Server, a default information catalog
is created on DB2 Universal Database for Windows NT.
```

The statement is incorrect. You must define a new information catalog. See the
"Creating the Information Catalog" section for more information.

Exchanging Metadata with Other Products

In Chapter 6, "Exchanging metadata with other products", in the section
"Identifying OLAP objects to publish", there is a statement in the second
paragraph that says:

```
When you publish DB2 OLAP Integration Server metadata, a linked relationship
is created between an information catalog "dimensions within a
multi-dimensional database" object type and a table object
in the OLAP Integration Server.
```

The statement should say:

```
When you publish DB2 OLAP Integration Server metadata, a linked relationship
is created between an information catalog "dimensions within a
multi-dimensional database object and a table object".
```

This statement also appears in Appendix C, "Metadata mappings", in the
section "Metadata mappings between the Information Catalog Manager and
OLAP Server".

Exchanging Metadata using the flgnxoln Command

In Chapter 6, "Exchanging Metadata", there is a section entitled "Identifying
OLAP objects to publish". At the end of this section there is an example of
using the flgnxoln command to publish OLAP server metadata to an
information catalog. The example incorrectly shows the directory for the

db2olap.ctl and db2olap.ff files as x:\Program Files\sqllib\logging. The directory name should be x:\Program Files\sqllib\exchange as described on page 87.

Exchanging Metadata using the MDISDGC Command

Chapter 6. Exchanging metadata with other products: "Converting MDIS-conforming metadata into a tag language file", page 97. You cannot issue the MDISDGC command from the MS-DOS command prompt. You must issue the MDISDGC command from a DB2 command window. The first sentence of the section, "Converting a tag language file into MDIS-conforming metadata," also says you must issue the DGMDISC command from the MS-DOS command prompt. You must issue the DGMDISC command from a DB2 command window.

Invoking Programs

Some examples in the Information Catalog Administration Guide show commands that contain the directory name Program Files. When you invoke a program that contains Program Files as part of its path name, you must enclose the program invocation in double quotation marks. For example, Appendix B, "Predefined Information Catalog Manager object types", contains an example in the section called "Initializing your information catalog with the predefined object types". If you use the example in this section, you will receive an error when you run it from the DOS prompt. The following example is correct:

```
"X:Program Files\SQLLIB\SAMPLES\SAMPDATA\DGWDEMO"
/T userid password dgname
```

Information Catalog Manager Programming Guide and Reference

Information Catalog Manager Reason Codes

In Appendix D: Information Catalog Manager reason codes, some text might be truncated at the far right column for the following reason codes: 31014, 32727, 32728, 32729, 32730, 32735, 32736, 32737, 33000, 37507, 37511, and 39206. If the text is truncated, please see the HTML version of the book to view the complete column.

Information Catalog Manager User's Guide

In Chapter 2, there is a section called "Registering a server node and remote information catalog." The section lists steps that you can complete from the DB2 Control Center before registering a remote information catalog using the Information Catalog Manager. The last paragraph of the section says that after completing a set of steps from the DB2 Control Center (add a system, add an instance, and add a database), you must shut down the Control Center before opening the Information Catalog Manager. That information is incorrect. It is not necessary to shut down the Control Center before opening the Information Catalog Manager.

The same correction also applies to the online help task "Registering a server node and remote information catalog", and the online help for the Register Server Node and Information Catalog window.

Information Catalog Manager: Online Messages

Corrections to FLG messages

Message FLG0260E

The second sentence of the message explanation should say:

```
The error caused a rollback of the information catalog,
which failed.  The information catalog is not in stable
condition, but no changes were made.
```

Message FLG0051E

The second bullet in the message explanation should say:

```
The information catalog contains too many objects or object types.
```

The administrator response should say:

```
Delete some objects or object types from the current
information catalog using the import function.
```

Message FLG0003E

The message explanation should say:

```
The information catalog must be registered before you can use it.
The information catalog might not have been registered correctly.
```

Message FLG0372E

The first sentence of the message explanation should say:

```
The ATTACHMENT-IND value was ignored for an object
because that object is an Attachment object.
```

Message FLG0615E

The second sentence of the message should say:

```
The Information Catalog Manager has encountered an unexpected
database error or cannot find the bind file
in the current directory or path.
```

Information Catalog Manager: Online Help

Information Catalog window: The online help for the Selected menu Open item incorrectly says "Opens the selected object". It should say "Opens the Define Search window".

Information Catalog Manager for the Web

When using an information catalog that is located on a DB2 UDB for OS/390 system, case insensitive search is not available. This is true for both a simple search and an advanced search. The online help does not explain that all searches on a DB2 UDB for OS/390 information catalog are case sensitive for a simple search. Moreover, all grouping category objects are expandable, even when there are no underlying objects.

DB2 Warehouse Manager Installation Guide

DB2 Warehouse Manager Installation Guide Update Available

The DB2 Warehouse Manager Installation Guide has been updated and the latest .pdf is available for download online at http://www.ibm.com/software/data/db2/udb/winos2unix/support. All updated documentation is also available on CD. This CD can be ordered through service using the PTF number U478862. The information in these notes is in addition to the updated reference.

Software requirements for warehouse transformers

The Java Developer's Kit (JDK) Version 1.1.8 or later must be installed on the database where you plan to use the warehouse transformers.

Connector for SAP R/3

When mapping columns from fields of an SAP R/3 business object to DB2 tables, some generated column names might be longer than 30 characters. In this case, the generated column name will reflect only the first 30 characters of the SAP field name. If the generated name is not what you want, you can change it using the Properties notebook for the table.

Installation Prerequisites

if a value is specified in a destination field on an SAP source page, then set the RFC_INI environment.. For example, Set RFC_INI=c:\rfcapl.ini. After you set this variable, you must reboot the machine.

Connector for the Web

If you have problems running the Connector for the Web, IBM Service might request that you send a trace for the Connector.

To enable tracing for the Connector for the Web, set the Warehouse Center agent trace to a level greater than 0. The trace file is named WSApid.log, where pid is the Windows process ID for the agent. The trace file is created in the \sqllib\logging directory.

Installation Prerequisites

Install the Java run-time environment (JRE) or Java virtual machine (JVM), version 1.2.2 or later, and make it your default. To make a version of the JRE your default, add the path for the 1.2.2 JRE to your system PATH variable (for example, C:\JDKs\IBM\java12\bin;). After you change your default JRE, you

must reboot the machine. If you do not have Java installed, you can install it
from the Data Warehouse Connectors installation CD.

Post-installation considerations for the iSeries agent

In Chapter 4, under the "Installing the AS/400 (iSeries) warehouse agent"
section, under the "Post-installation considerations" subsection, change the
first paragraph to:

> The warehouse agent performs all step functions in a single unit of
> work. Prior to V4R5, DB2 Universal Database for iSeries limited the
> number of rows that can be inserted in a single commit scope to 4
> million. This limitation has been increased to 500 million rows in V4R5.
> If you are using a V4R4 (or below) system and have queries that exceed
> this size, either subdivide the queries or use the warehouse-supplied FTP
> programs to move data.

Before using transformers with the iSeries warehouse agent

In Chapter 4, remove the section "Before using transformers with the iSeries
Agent" and all of its subsections.

Query Patroller Administration Guide

DB2 Query Patroller Client is a Separate Component

The DB2 Query Patroller client is a separate component that is not part of the DB2 Administration client. This means that it is not installed during the installation of the DB2 Administration Client, as indicated in the Query Patroller Installation Guide. Instead, the Query Patroller client must be installed separately.

The version and level of the Query Patroller client and the Query Patroller server must be the same.

Changing the Node Status

The following is an update to the Node Administration section of the Query Patroller Administration Guide.

Use the following procedure to change the node status:
1. On the Node Administration page, select a node.
2. Click on **View / Edit**.

   The Detailed Information for Node window opens.
3. Select the new status in the **Status Requested** field.

   Note: Status Requested is the only field in the Detailed Information for Node window that can be changed; all other fields display values that have been supplied by DB2 Query Patroller.
4. Click on **OK**.

The following list provides information for each node parameter:

Node ID
    Provides the ID for the node.

Node Status
    Contains the current node status:

    - **Active** indicates that the node is able to run jobs.
    - **Inactive** indicates that the node's DB2 Query Patroller component is shut down. The node is not available to DB2 Query Patroller. To reactivate the node, use the `iwm` administrative user account to issue the `dqpstart` command.

281

- **Quiescing** indicates that the node is in transition to the quiescent state. Running jobs will complete, but no new jobs will be scheduled on the node.
- **Quiesced** indicates that the node is quiescent. The node is available to DB2 Query Patroller but no new jobs are being scheduled to that node.

Status Requested

Indicates what the node status will be changed to:

- **Active** indicates that the node will be made active.
- **Inactive** indicates that the node will be made inactive. Running jobs will complete and no new jobs will be scheduled.
- **Force** indicates that the node will be made inactive immediately. Running jobs are terminated immediately and no new jobs will be scheduled.
- **Quiesced** indicates that the node will be made quiescent. Running jobs will complete.

Date/Time Last Status

Indicates the date and time node status was last changed.

Scheduled Jobs

Provides the number of jobs scheduled to run plus the number of jobs running on this node.

CPU Utilization

Provides the CPU utilization of the node as a percentage (0 - 100). If CPU utilization information is not being collected, the value is -1.

Disk Available

Indicates the bytes available in the file system where results are created. If disk utilization is not being monitored, the value is -1.

Node Manager PID

Indicates the process ID of the node manager process.

Migrating from Version 6 of DB2 Query Patroller Using dqpmigrate

The dqpmigrate command must be used if the Version 7 Query Patroller Server was installed over the Version 6 Query Patroller Server. For FixPak 2 or later, you do not have to run dqpmigrate manually as the installation of the FixPak runs this command for you. Without using this command, the existing users defined in v6 have no EXECUTE privileges on several new stored procedures added in Version 7.

Note: dqpmigrate.bnd is found in the sqllib/bnd directory and dqpmigrate.exe is found in the sqllib/bin directory.

To use **dqpmigrate** manually to grant the EXECUTE privileges, perform the
following after installing the FixPak:

1. Bind the /sqllib/bnd/dqpmigrate.bnd package file to the database where
   the Query Patroller server has been installed by entering the following
   command:

   ```
   db2 bind dqpmigrate.bnd
   ```

2. Execute **dqpmigrate** by entering the following:

   ```
   dqpmigrate dbalias userid passwd
   ```

Enabling Query Management

In the "Getting Started" chapter under "Enabling Query Management", the
text should read:

```
You must be the owner of the data base, or you must have SYSADM,
SYSCTRL, or SYSMAINT authority to set database configuration parameters.
```

Location of Table Space for Control Tables

In Chapter 1, System Overview, under DB2 Query Patroller Control Tables,
the following text is to be added at the end of the section's first paragraph:

The table space for the DB2 Query Patroller control tables must reside in a
single-node nodegroup, or DB2 Query Patroller will not function properly.

New Parameters for dqpstart Command

In Chapter 2, Getting Started, under Starting and Stopping DB2 Query
Patroller, the following text is to be added following the last paragraph:

New Parameters for the **dqpstart** command:

RESTART parameter:
> Allows the user to replace the host name and/or the node type of the
> specified node in the dqpnodes.cfg file. DB2 Query Patroller will be
> started on this node.

> **Note:** Before running the DQPSTART command with the RESTART
> parameter, ensure the following:
> 1. DB2 Query Patroller is already stopped on the host that is
>   going to be replaced.
> 2. DB2 Query Patroller is not already running on the new host.

The syntax is as follows:

```
dqpstart nodenum node_num restart hostname server | agent | none
```

ADDNODE parameter:
Allows the user to add a new node to the dqpnodes.cfg file. DB2
Query Patroller will be started on this node after the new node entry
is added to the dqpnodes.cfg file. The syntax is as follows:

```
dqpstart nodenum node_num addnode hostname server | agent | none
```

DROPNODE parameter:
Allows the user to drop a node from the dqnodes.cfg file. DB2 Query
Patroller will be stopped on this node before the node entry is
dropped from the dqpnodes.cfg file. The syntax is as follows:

```
dqpstop nodenum node_num dropnode
```

New Parameter for iwm_cmd Command

A new **-v** parameter has been added to the **iwm_cmd** command to allow the
user to recover the status of the jobs that were running on the node specified.
Only jobs on an inactive node are allowed to be recovered. This command
should be issued when there is a node failure and there are some jobs running
on that node or being cancelled at the time. Jobs that were in "Running" state
will be resubmitted and set back to "Queued" state. Jobs that were in
"Cancelling" state will be set to "Cancelled" state.

The partial syntax is as follows:

```
►►─iwm_cmd─────────────────────────────────-v─node_id_to_recover──────────►◄
           └─-u─user_id─┐
                        └─-p─password─┘
```

node_id_to_recover
Specifies the node on which the jobs are to be recovered.

New Registry Variable: DQP_RECOVERY_INTERVAL

There is a new registry variable called DQP_RECOVERY_INTERVAL which is
used to set the interval of time in minutes that the iwm_scheduler searches for
recovery files. The default is 60 minutes.

Starting Query Administrator

In the "Using QueryAdministrator to Administer DB2 Query Patroller"
chapter, instructions are provided for starting QueryAdministrator from the
Start menu on Windows. The first step provides the following text:

```
If you are using Windows, you can select DB2
Query Patroller –> QueryAdministrator
from the IBM DB2 program group.
```

The text should read:

```
DB2 Query Patroller -> QueryAdmin.
```

User Administration

In the "User Administration" section of the "Using QueryAdministrator to Administer DB2 Query Patroller" chapter, the definition for the Maximum Elapsed Time parameter indicates that if the value is set to 0 or -1, the query will always run to completion. This parameter cannot be set to a negative value. The text should indicate that if the value is set to 0, the query will always run to completion.

The Max Queries parameter specifies the maximum number of jobs that the DB2 Query Patroller will run simultaneously. Max Queries must be an integer within the range of 0 to 32767.

Data Source Administration

In Chapter 3, Using Query Administrator to Administer DB2 Query Patroller, there are some new and changed descriptions for the data source parameters.

Static Cost is the DB2 estimated cost of the query in timerons. This cost is stored in the job entry for each job. You can see it as the Estimated Cost when using Query Monitor to look at the job details of a job.

Zero Cost Query is the query with a static cost, or estimated cost, of zero. No query actually has an estimated cost of zero (even the very simplest ones have a cost of around 5). Rather, this occurs if a job is submitted with the do not do cost analysis option. You can only choose this option if you have set up the user profile to allow it. In most cases, you will not have your user profiles set up in this way. You should keep this option for superusers like other administrators, selected special users, or yourself. These users can then run whatever queries they want. The system treats queries from these users as zero cost so that the queries are treated as high priority.

Cost Time Zero, Cost Time Slope, Cost Time Interval, and Cost Time Min are no longer used.

The Cost Factor is the multiplier to convert the Static Cost in timerons is not the cost in the accounting table. The cost in the accounting table is equal to the Static Cost multiplied by the Cost Factor.

Creating a Job Queue

In the "Job Queue Administration" section of the "Using QueryAdministrator to Administer DB2 Query Patroller" chapter, the screen capture in the steps for "Creating a Job Queue" should be displayed after the second step. The Information about new Job Queue window opens once you click **New** on the Job Queue Administration page of the QueryAdministrator tool.

References to the Job Queues page or the Job Queues tab should read Job Queue Administration page and Job Queue Administration tab, respectively.

Job Accounting Table

In chapter 11, Monitoring the DB2 Query Patroller System, the section on Job Accounting describes the columns in the Job Accounting table. The table name is IWM.IWM003_JOB_ACCT.

Using the Command Line Interface

For a user with User authority on the DB2 Query Patroller system to submit a query and have a result table created, the user may require CREATETAB authority on the database. The user does not require CREATETAB authority on the database if the DQP_RES_TBLSPC profile variable is left unset, or if the DQP_RES_TBLSPC profile variable is set to the name of the default table space. The creation of the result tables will succeed in this case because users have the authority to create tables in the default table space.

Query Enabler Notes

- When using third-party query tools that use a keyset cursor, queries will not be intercepted. In order for Query Enabler to intercept these queries, you must modify the db2cli.ini file to include:

      [common]
      DisableKeySetCursor=1

- For AIX clients, please ensure that the environment variable LIBPATH is not set. Library libXext.a, shipped with the JDK, is not compatible with the library in the /usr/lib/X11 subdirectory. This will cause problems with the Query Enabler GUI.

DB2 Query Patroller Tracker may Return a Blank Column Page

FixPak 3 includes a fix for the DB2 Query Patroller Tracker. The Tracker will now correctly report queries which hit no columns. An example of such a query is "SELECT COUNT(*) FROM ...". Since this kind of query does not hit any column in the table, the Tracker will present a blank page for the column page. This blank column page is not a defect.

Additional Information for DB2 Query Patroller Tracker GUI Tool

The accounting table is used by the Tracker tool when used to display or analyze historical job data. In order to use Tracker, the administrator must first use Query Administrator to change the Accounting Status on the System Administrator panel to Write To Table. Then, whenever a job completes, extra information is saved in a job accounting table.

Next, the administrator must log on to the Query Patroller server as user iwm and run the iwm_tracker (Tracker backend) tool. This tool should be run periodically when the system load is low, or just before when the Tracker tool is used.

Finally, when these two tasks are completed, you can run the Tracker GUI tool to view or analyze the job data.

If the cost factor is one, which is the default, then the cost displayed for each job using the Tracker is the same value as the cost displayed using Query Monitor. In both cases, the time is in timerons.

However, you may want to use other units of value. Suppose you want to bill each user for their use of the system. If, for example, the charge is one dollar for 10 000 timerons of work, then you would enter a cost factor of 0.0001. This has the Tracker converting, storing, and displaying each job's cost in dollars.

The *Query Patroller Administration Guide* discusses dollars per megabyte which is incorrect and should be replaced by dollars per timeron.

Query Patroller and Replication Tools

Query Patroller Version 7 will intercept the queries of the replication tools (asnapply, asnccp, djra and analyze) and cause these tools to malfunction. A workaround is to disable dynamic query management when running these tools.

Improving Query Patroller Performance

The following text should appear at the end of Chapter 6, Performance Tuning:

Using the BIND Option, INSERT BUF to Improve DB2 Query Patroller Performance

By default, DB2 Query Patroller creates result tables to store the results of the queries it manages. To increase the performance of inserts to these result tables, include the INSERT BUF option when binding one of the DB2 Query Patroller bind files.

Bind the DB2 Query Patroller bind files to the database as follows:

From the DB2_RUNTIME\bnd directory on Windows, or the DB2_RUNTIME/bnd path on UNIX, enter the following commands:

```
db2 connect to database user iwm using password
db2 bind @db2qp.lst blocking all grant public
db2 bind iwmsx001.bnd insert buf
db2 bind @db2qp_sp.lst
db2 commit
```

where database is the database the replacement database that will be managed by DB2 Query Patroller, and password is the password for the administrative user account, iwm.

Lost EXECUTE Privilege for Query Patroller Users Created in Version 6

Because of some new stored procedures (IWM.DQPGROUP, IWM.DQPVALUR, IWM.DQPCALCT, and IWM.DQPINJOB) added in Query Patroller Version 7, existing users created in Query Patroller Version 6 do not hold the EXECUTE privilege on those packages. An application to automatically correct this problem has been added to FixPak 1.

When you try to use DQP Query Admin to modify DQP user information, please do not try to remove existing users from the user list.

Query Patroller Restrictions

Because of JVM (Java Virtual Machine) platform restrictions, the Query Enabler is not supported on HP-UX and NUMA-Q. In addition, the Query Patroller Tracker is not supported on NUMA-Q. If all of the Query Patroller client tools are required, we recommend the use of a different platform (such as Windows NT) to run these tools against the HP-UX or NUMA-Q server.

Appendix B. Troubleshooting DB2 Query Patroller Clients

In Appendix B, Troubleshooting DB2 Query Patroller Clients, section: Common Query Enabler Problems, problem #2, the text of the first bullet is replaced with:

```
Ensure that the path setting includes jre.
```

Part 6. Application Development

Administrative API Reference

db2ArchiveLog (new API)

db2ArchiveLog

Closes and truncates the active log file for a recoverable database. If user exit is enabled, issues an archive request.

Authorization
One of the following:

- *sysadm*
- *sysctrl*
- *sysmaint*
- *dbadm*

Required Connection
This API automatically establishes a connection to the specified database. If a connection to the specified database already exists, the API will return an error.

API Include File
db2ApiDf.h

C API Syntax

```
/* File: db2ApiDf.h */
/* API:  Archive Active Log */
SQL_API_RC SQL_API_FN
   db2ArchiveLog (
      db2Uint32 version,
      void *pDB2ArchiveLogStruct,
      struct sqlca * pSqlca);

typedef struct
{
   char               *piDatabaseAlias;
   char               *piUserName;
   char               *piPassword;
   db2Uint16          iAllNodeFlag;
   db2Uint16          iNumNodes;
   SQL_PDB_NODE_TYPE  *piNodeList;
   db2Uint32          iOptions;
} db2ArchiveLogStruct
```

Generic API Syntax

```
/* File: db2ApiDf.h */
/* API:  Archive Active Log */
SQL_API_RC SQL_API_FN
   db2gArchiveLog (
      db2Uint32 version,
      void *pDB2ArchiveLogStruct,
      struct sqlca * pSqlca);

typedef struct
{
   db2Uint32          iAliasLen;
   db2Uint32          iUserNameLen;
   db2Uint32          iPasswordLen;
   char               *piDatabaseAlias;
   char               *piUserName;
   char               *piPassword;
   db2Uint16          iAllNodeFlag;
   db2Uint16          iNumNodes;
   SQL_PDB_NODE_TYPE  *piNodeList;
   db2Uint32          iOptions;
} db2ArchiveLogStruct
```

API Parameters

version
> Input. Specifies the version and release level of the variable passed in as the second parameter, *pDB2ArchiveLogStruct*.

pDB2ArchiveLogStruct
> Input. A pointer to the *db2ArchiveLogStruct* structure.

pSqlca
> Output. A pointer to the *sqlca* structure.

iAliasLen
> Input. A 4-byte unsigned integer representing the length in bytes of the database alias.

iUserNameLen
> A 4-byte unsigned integer representing the length in bytes of the user name. Set to zero if no user name is used.

iPasswordLen
> Input. A 4-byte unsigned integer representing the length in bytes of the password. Set to zero if no password is used.

piDatabaseAlias
> Input. A string containing the database alias (as cataloged in the system database directory) of the database for which the active log is to be archived.

piUserName
> Input. A string containing the user name to be used when attempting a connection.

piPassword
> Input. A string containing the password to be used when attempting a connection.

iAllNodeFlag
> MPP only. Input. Flag indicating whether the operation should apply to all nodes listed in the db2nodes.cfg file. Valid values are:

> **DB2ARCHIVELOG_NODE_LIST**
>> Apply to nodes in a node list that is passed in *piNodeList*.

> **DB2ARCHIVELOG_ALL_NODES**
>> Apply to all nodes. *piNodeList* should be NULL. This is the default value.

> **DB2ARCHIVELOG_ALL_EXCEPT**
>> Apply to all nodes except those in the node list passed in *piNodeList*.

iNumNodes
> MPP only. Input. Specifies the number of nodes in the *piNodeList* array.

piNodeList
> MPP only. Input. A pointer to an array of node numbers against which to apply the archive log operation.

iOptions
> Input. Reserved for future use.

db2ConvMonStream

In the Usage Notes, the structure for the snapshot variable datastream type
SQLM_ELM_SUBSECTION should be sqlm_subsection.

db2DatabasePing (new API)

db2DatabasePing - Ping Database

Tests the network response time of the underlying connectivity between a client and a database server. This API can be used by an application when a host database server is accessed via DB2 Connect either directly or through a gateway.

Authorization

None

Required Connection

Database

API Include File

db2ApiDf.h

C API Syntax

```
/* File: db2ApiDf.h */
/* API: Ping Database */
/* ... */


SQL_API_RC SQL_API_FN
  db2DatabasePing (
      db2Uint32     versionNumber,
      void         *pParmStruct,
      struct sqlca *pSqlca);
/* ... */

typedef SQL_STRUCTURE db2DatabasePingStruct
{
  char         iDbAlias[SQL_ALIAS_SZ + 1];
  db2Uint16    iNumIterations;
  db2Uint32   *poElapsedTime;
}
```

Generic API Syntax

```
/* File: db2ApiDf.h */
/* API: Ping Database */
/* ... */
SQL_API_RC SQL_API_FN
  db2gDatabasePing (
      db2Uint32     versionNumber,
      void         *pParmStruct,
      struct sqlca *pSqlca);
/* ... */

typedef SQL_STRUCTURE db2gDatabasePingStruct
{
  db2Uint16    iDbAliasLength;
```

```
        char        iDbAlias[SQL_ALIAS_SZ];
        db2Uint16   iNumIterations;
        db2Uint32   *poElapsedTime;
    }
```

API Parameters

versionNumber

> Input. Version and release of the DB2 Universal Database or DB2
> Connect product that the application is using.
>
> **Note:** Constant db2Version710 or higher should be used for DB2
> Version 7.1 or higher.

pParmStruct

> Input. A pointer to the db2DatabasePingStruct Structure.

iDbAliasLength

> Input. Length of the database alias name.
>
> **Note:** This parameter is not currently used. It is reserved for future
> use.

iDbAlias

> Input. Database alias name.
>
> **Note:** This parameter is not currently used. It is reserved for future
> use.

iNumIterations

> Input. Number of test request iterations. The value must be between 1
> and 32767 inclusive.

poElapsedTime

> Output. A pointer to an array of 32-bit integers where the number of
> elements is equal to iNumIterations. Each element in the array will
> contain the elapsed time in microseconds for one test request iteration.
>
> **Note:** The application is responsible for allocating the memory for this
> array prior to calling this API.

pSqlca

> Output. A pointer to the *sqlca* structure. For more information about
> this structure, see the *Administrative API Reference*.

Usage Notes

A database connection must exist before invoking this API, otherwise an error
will result.

This function can also be invoked using the PING command. For a description of this command, see the *Command Reference*.

db2HistData

The following entries should be added to Table 11. Fields in the db2HistData Structure:

Field Name	Data Type	Description
oOperation	char	See Table 12.
oOptype	char	See Table 13.

The following table will be added following Table 11.

Table 12. Valid event values for oOperation in the db2HistData Structure

Value	Description	C Definition	COBOL/FORTRAN Definition
A	add tablespace	DB2HISTORY_OP_ADD_TABLESPACE	DB2HIST_OP_ADD_TABLESPACE
B	backup	DB2HISTORY_OP_BACKUP	DB2HIST_OP_BACKUP
C	load-copy	DB2HISTORY_OP_LOAD_COPY	DB2HIST_OP_LOAD_COPY
D	dropped table	DB2HISTORY_OP_DROPPED_TABLE	DB2HIST_OP_DROPPED_TABLE
F	roll forward	DB2HISTORY_OP_ROLLFWD	DB2HIST_OP_ROLLFWD
G	reorganize table	DB2HISTORY_OP_REORG	DB2HIST_OP_REORG
L	load	DB2HISTORY_OP_LOAD	DB2HIST_OP_LOAD
N	rename tablespace	DB2HISTORY_OP_REN_TABLESPACE	DB2HIST_OP_REN_TABLESPACE
O	drop tablespace	DB2HISTORY_OP_DROP_TABLESPACE	DB2HIST_OP_DROP_TABLESPACE
Q	quiesce	DB2HISTORY_OP_QUIESCE	DB2HIST_OP_QUIESCE
R	restore	DB2HISTORY_OP_RESTORE	DB2HIST_OP_RESTORE
S	run statistics	DB2HISTORY_OP_RUNSTATS	DB2HIST_OP_RUNSTATS
T	alter tablespace	DB2HISTORY_OP_ALT_TABLESPACE	DB2HIST_OP_ALT_TBS
U	unload	DB2HISTORY_OP_UNLOAD	DB2HIST_OP_UNLOAD

The following table will also be added.

Table 13. Valid oOptype values db2HistData Structure

oOperation	oOptype	Description	C/COBOL/FORTRAN Definition

B	F	Offline	DB2HISTORY_OPTYPE_OFFLINE
	N	Online	DB2HISTORY_OPTYPE_ONLINE
	I	Incremental offline	DB2HISTORY_OPTYPE_INCR_OFFLINE
	O	Incremental online	DB2HISTORY_OPTYPE_INCR_ONLINE
	D	Delta offline	DB2HISTORY_OPTYPE_DELTA_OFFLINE
	E	Delta online	DB2HISTORY_OPTYPE_DELTA_ONLIN
F	E	End of log	DB2HISTORY_OPTYPE_EOL
	P	Point in time	DB2HISTORY_OPTYPE_PIT
L	I	Insert	DB2HISTORY_OPTYPE_INSERT
	R	Replace	DB2HISTORY_OPTYPE_REPLACE
Q	S	Quiesce share	DB2HISTORY_OPTYPE_SHARE
	U	Quiesce update	DB2HISTORY_OPTYPE_UPDATE
	X	Quiesce exclusive	DB2HISTORY_OPTYPE_EXCL
	Z	Quiesce reset	DB2HISTORY_OPTYPE_RESET
R	F	Offline	DB2HISTORY_OPTYPE_OFFLINE
	N	Online	DB2HISTORY_OPTYPE_ONLINE
	I	Incremental offline	DB2HISTORY_OPTYPE_INCR_OFFLINE
	O	Incremental online	DB2HISTORY_OPTYPE_INCR_ONLINE
T	C	Add containers	DB2HISTORY_OPTYPE_ADD_CONT
	R	Rebalance	DB2HISTORY_OPTYPE_REB

db2HistoryOpenScan

The following value will be added to the **iCallerAction** parameter.

DB2HISTORY_LIST_CRT_TABLESPACE
Select only the CREATE TABLESPACE and DROP TABLESPACE
records that pass the other filters.

db2Runstats

When the db2Runstats API is collecting statistics on indexes only, then
previously collected distribution statistics are retained. Otherwise, the API will
drop previously collected distribution statistics.

db2GetSnapshot – Get Snapshot

The syntax for the db2GetSnapshot API should be as follows:

```
int db2GetSnapshot( unsigned char version;
db2GetSnapshotData *data,
struct sqlca *sqlca);

The parameters described in data are:
typedef struct db2GetSnapshotData{
   sqlma    *piSqlmaData;
   sqlm_collected  *poCollectedData
   void     *poBuffer;
   db2uint32  iVersion;
   db2int32  iBufferSize;
   db2uint8  iStoreResult;
db2uint16  iNodeNumber;
db2uint32  *poOutputFormat;

}db2GetSnapshotData;
```

db2XaGetInfo (new API)

db2XaGetInfo - Get Information for Resource Manager

Extracts information for a particular resource manager once an xa_open call has been made.

Authorization
None

Required Connection
Database

API Include File
sqlxa.h

C API Syntax

```
/* File: sqlxa.h */
/* API: Get Information for Resource Manager */
/* ... */
SQL_API_RC SQL_API_FN
db2XaGetInfo (
   db2Uint32 versionNumber,
   void * pParmStruct,
   struct sqlca * pSqlca);

typedef SQL_STRUCTURE db2XaGetInfoStruct
{
  db2int32 iRmid;
  struct sqlca oLastSqlca;
} db2XaGetInfoStruct;
```

API Parameters

versionNumber
>   Input. Specifies the version and release level of the structure passed in as the second parameter, *pParmStruct*.

pParmStruct
>   Input. A pointer to the *db2XaGetInfoStruct* structure.

pSqlca
>   Output. A pointer to the *sqlca* structure. For more information about this structure, see the *Administrative API Reference*.

iRmid Input. Specifies the resource manager for which information is required.

oLastSqlca
>   Output. Contains the *sqlca* for the last XA API call.
>
>   **Note:** Only the *sqlca* that resulted from the last failing XA API can be retrieved.

db2XaListIndTrans (new API that supercedes sqlxphqr)

db2XaListIndTrans - List Indoubt Transactions

Provides a list of all indoubt transactions for the currently connected database.

Scope
This API affects only the node on which it is issued.

Authorization
One of the following:

- *sysadm*
- *dbadm*

Required Connection
Database

API Include File
db2ApiDf.h

C API Syntax

```
/* File: db2ApiDf.h */
/* API: List Indoubt Transactions */
/* ... */
SQL_API_RC SQL_API_FN
db2XaListIndTrans (
    db2Uint32 versionNumber,
    void * pParmStruct,
    struct sqlca * pSqlca);

typedef SQL_STRUCTURE db2XaListIndTransStruct
{
  db2XaRecoverStruct * piIndoubtData;
  db2Uint32            iIndoubtDataLen;
  db2Uint32            oNumIndoubtsReturned;
  db2Uint32            oNumIndoubtsTotal;
  db2Uint32            oReqBufferLen;
} db2XaListIndTransStruct;

typedef SQL_STRUCTURE db2XaRecoverStruct
{
  sqluint32     timestamp;
  SQLXA_XID     xid;
  char          dbalias[SQLXA_DBNAME_SZ];
  char          applid[SQLXA_APPLID_SZ];
  char          sequence_no[SQLXA_SEQ_SZ];
  char          auth_id[SQL_USERID_SZ];
  char          log_full;
  char          connected;
  char          indoubt_status;
  char          originator;
  char          reserved[8];
} db2XaRecoverStruct;
```

API Parameters

versionNumber
> Input. Specifies the version and release level of the structure passed in as the second parameter, *pParmStruct*.

pParmStruct
> Input. A pointer to the *db2XaListIndTransStruct* structure.

pSqlca
> Output. A pointer to the *sqlca* structure. For more information about this structure, see the *Administrative API Reference*.

piIndoubtData
> Input. A pointer to the application supplied buffer where indoubt data will be returned. The indoubt data is in *db2XaRecoverStruct* format. The application can traverse the list of indoubt transactions by using the size of the *db2XaRecoverStruct* structure, starting at the address provided by this parameter.
>
> If the value is NULL, DB2 will calculate the size of the buffer required and return this value in *oReqBufferLen*. *oNumIndoubtsTotal* will contain the total number of indoubt transactions. The application may allocate the required buffer size and issue the API again.

oNumIndoubtsReturned
> Output. The number of indoubt transaction records returned in the buffer specified by *pIndoubtData*.

oNumIndoubtsTotal
> Output. The Total number of indoubt transaction records available at the time of API invocation. If the *piIndoubtData* buffer is too small to contain all the records, *oNumIndoubtsTotal* will be greater than the total for *oNumIndoubtsReturned*. The application may reissue the API in order to obtain all records.
>
> **Note:** This number may change between API invocations as a result of automatic or heuristic indoubt transaction resynchronisation, or as a result of other transactions entering the indoubt state.

oReqBufferLen
> Output. Required buffer length to hold all indoubt transaction records at the time of API invocation. The application can use this value to determine the required buffer size by calling the API with *pIndoubtData* set to NULL. This value can then be used to allocate the required buffer, and the API can be issued with *pIndoubtData* set to the address of the allocated buffer.

Note: The required buffer size may change between API invocations as a result of automatic or heuristic indoubt transaction resynchronisation, or as a result of other transactions entering the indoubt state. The application may allocate a larger buffer to account for this.

timestamp
Output. Specifies the time when the transaction entered the indoubt state.

xid Output. Specifies the XA identifier assigned by the transaction manager to uniquely identify a global transaction.

dbalias
Output. Specifies the alias of the database where the indoubt transaction is found.

applid Output. Specifies the application identifier assigned by the database manager for this transaction.

sequence_no
Output. Specifies the sequence number assigned by the database manager as an extension to the *applid*.

auth_id
Output. Specifies the authorization ID of the user who ran the transaction.

log_full
Output. Indicates whether or not this transaction caused a log full condition. Valid values are:

SQLXA_TRUE
This indoubt transaction caused a log full condition.

SQLXA_FALSE
This indoubt transaction did not cause a log full condition.

connected
Output. Indicates whether or not the application is connected. Valid values are:

SQLXA_TRUE
The transaction is undergoing normal syncpoint processing, and is waiting for the second phase of the two-phase commit.

SQLXA_FALSE
The transaction was left indoubt by an earlier failure, and is now waiting for resynchronisation from the transaction manager.

indoubt_status

> Output. Indicates the status of this indoubt transaction. Valid values are:

> **SQLXA_TS_PREP**
>> The transaction is prepared. The connected parameter can be used to determine whether the transaction is waiting for the second phase of normal commit processing or whether an error occurred and resynchronisation with the transaction manager is required.

> **SQLXA_TS_HCOM**
>> The transaction has been heuristically committed.

> **SQLXA_TS_HROL**
>> The transaction has been heuristically rolled back.

> **SQLXA_TS_MACK**
>> The transaction is missing commit acknowledgement from a node in a partitioned database.

> **SQLXA_TS_END**
>> The transaction has ended at this database. This transaction may be re-activated, committed, or rolled back at a later time. It is also possible that the transaction manager encountered an error and the transaction will not be completed. If this is the case, this transaction requires heuristic actions, because it may be holding locks and preventing other applications from accessing data.

Usage Notes

A typical application will perform the following steps after setting the current connection to the database or to the partitioned database coordinator node:

1. Call **db2XaListIndTrans** with *piIndoubtData* set to NULL. This will return values in *oReqBufferLen* and *oNumIndoubtsTotal*.

2. Use the returned value in *oReqBufferLen* to allocate a buffer. This buffer may not be large enough if there are additional indoubt transactions because the initial invocation of this API to obtain *oReqBufferLen*. The application may provide a buffer larger than *oReqBufferLen*.

3. Determine if all indoubt transaction records have been obtained. This can be done by comparing *oNumIndoubtsReturned* to *oNumIndoubtTotal*. If *oNumIndoubtsTotal* is greater than *oNumIndoubtsReturned*, the application can repeat the above steps.

See Also

"sqlxhfrg - Forget Transaction Status", "sqlxphcm - Commit an Indoubt Transaction", and "sqlxphrl - Roll Back an Indoubt Transaction" in the *Administrative API Reference*.

Forget Log Record

The following information will be added to Appendix F following the MPP Subordinator Prepare section.

This log record is written after a rollback of indoubt transactions or after a commit of two-phase commit. The log record is written to mark the end of the transaction and releases any log resources held. In order for the transaction to be forgotten, it must be in a heuristically completed state.

Table 10. Forget Log Record Structure

Description	Type	Offset (Bytes)
Log header	LogManagerLogRecordHeader	0(20)
time	sqluint64	20(8)
Total Length: 28 bytes		

sqlaintp – Get Error Message

The following usage note is to be added to the description of this API:

```
In a multi-threaded application, sqlaintp must be attached
to a valid context; otherwise, the message text for
SQLCODE -1445 cannot be obtained.
```

sqlbctcq – Close Tablespace Container Query

Load is not a valid Authorization level for this API.

sqleseti – Set Client Information

The data values provided with the API can also be accessed by SQL special register. The values in these registers are stored in the database code page. Data values provided with this API are converted to the database code page before being stored in the special registers. Any data value that exceeds the maximum supported size after conversion to the database code page will be truncated before being stored at the server. These truncated values will be returned by the special registers. The original data values will also be stored at the server and are not converted to the database code page. The unconverted values can be returned by calling the **sqleqryi** API.

sqlubkp – Backup Database

For the **BackupType** parameter the SQLUB_FULL value will be replaced by the SQLUB_DB. A backup of all tablespaces in the database will be taken.

To support the new incremental backup functionality the SQLUB_INCREMENTAL and SQLUB_DELTA parameters will also be added. An incremental backup image is a copy of all database data which has changed since the most recent successful, full backup. A delta backup image is a copy of all database data that has changed since the most recent successful backup of any type

sqlureot – Reorganize Table

The following sentence will be added to the Usage Notes:

REORGANIZE TABLE cannot use an index that is based on an index extension.

sqlurestore – Restore Database

For the **RestoreType** parameter the SQLUD_FULL value will be replaced by the SQLUD_DB. A restore of all table spaces in the database will be taken. This will be run offline.

To support the new incremental restore functionality the SQLUD_INCREMENTAL parameter will also be added.

An incremental backup image is a copy of all database data which has changed since the most recent successful full backup.

Documentation Error Regarding AIX Extended Shared Memory Support (EXTSHM)

In "Appendix E. Threaded Applications with Concurrent Access", Note 2 should now read:

2. By default, AIX does not permit 32-bit applications to attach to more than 11 shared memory segments per process, of which a maximum of 10 can be used for local DB2 connections.

To use EXTSHM with DB2, do the following:

In client sessions:
```
export EXTSHM=ON
```

When starting the DB2 server:
```
export EXTSHM=ON
db2set DB2ENVLIST=EXTSHM
db2start
```

On EEE, also add the following lines to sqllib/db2profile:

```
EXTSHM=ON
export EXTSHM
```

SQLFUPD

locklist

The name of the token has changed from SQLF_DBTN_LOCKLIST to SQLF_DBTN_LOCK_LIST. The locklist parameter has been changed from a SMALLINT to a 64-bit unsigned INTEGER. The following addition should be made to the table of Updatable Database Configuration Parameters.

Parameter Name	Token	Token Value	Data Type
locklist	SQLF_DBTN_LOCK_LIST	704	Uint64

The new maximum for this parameter is 524 288.

Additionally, in "Chapter 3. Data Structures", Table 53. Updatable Database Configuration Parameters incorrectly lists the token value for dbheap as 701. The correct value is 58.

SQLEDBDESC

Two values will be added to the list of valid values for SQLDBCSS (defined in sqlenv). They are:

SQL_CS_SYSTEM_NLSCHAR
Collating sequence from system using the NLS version of compare routines for character types.

SQL_CS_USER_NLSCHAR
Collating sequence from user using the NLS version of compare routines for character types.

Application Building Guide

The Application Building Guide was updated as part of FixPak 4. The latest PDF is available for download online at http://www.ibm.com/software/data/db2/udb/winos2unix/support. All updated documentation is also available on CD. This CD can be ordered through DB2 service using the PTF number U478862. Information on contacting DB2 Service is available at http://www.ibm.com/cgi-bin/db2www/data/db2/udb/winos2unix/support/help.d2w/report.

Linux on S/390 and zSeries Support

DB2 for Linux on S/390 and zSeries supports the following operating system environments:

- SuSE v7.0
- SuSE SLES 7
- TurboLinux v6.1

Note: To run DB2 Version 7 on SuSE SLES 7, you need to install the libstdc++ v6.1 compat RPM, which is on CD 1 of the SuSE Linux Enterprise Server Developer's Edition CD set, in the path "CD1/suse/a1/compat.rpm". This must be done as root. To install the RPM, mount the CD image to a directory. For example, to mount it to directory /mnt, install the RPM with this command:

```
rpm -Uh /mnt/CD1/suse/a1/compat.rpm
```

and run **ldconfig** afterwards.

Linux Rexx Support

DB2 for Linux for Intel x86 (32-bit) supports Object REXX Interpreter for Linux Version 2.1.

DB2 for Linux on S/390 supports Object REXX 2.2.0 for Linux/390.

Additional Notes for Distributing Compiled SQL Procedures

On UNIX systems, ensure that the instance owner (i.e., the user under which the DB2 engine executes) and the owner of the $DB2PATH/adm/.fenced file belong to the same primary group. Alternatively, each of these two users should belong to the other's primary group.

If a GET ROUTINE or a PUT ROUTINE operation (or their corresponding procedure) fails to execute successfully, it will always return an error (SQLSTATE 38000), along with diagnostic text providing information about the cause of the failure. For example, if the procedure name provided to GET ROUTINE does not identify an SQL procedure, diagnostic "100, 02000" text will be returned, where "100" and "02000" are the SQLCODE and SQLSTATE, respectively, that identify the cause of the problem. The SQLCODE and SQLSTATE in this example indicate that the row specified for the given procedure name was not found in the catalog tables.

Application Development Guide

Update Available

The Application Development Guide was updated as part of FixPak 4. The latest PDF is available for download online at http://www.ibm.com/software/data/db2/udb/winos2unix/support. The information in these notes is in addition to the updated reference. All updated documentation is also available on CD. This CD can be ordered through DB2 service using the PTF number U478862. Information on contacting DB2 Service is available at http://www.ibm.com/cgi-bin/db2www/data/db2/udb/winos2unix/support/help.d2w/report.

Precaution for registering C/C++ routines (UDFs, stored procedures, or methods) on Windows

When registering a C or C++ routine (UDF, stored procedure, or method) on Windows® operating systems, take the following precaution when identifying a routine body in the CREATE statement's EXTERNAL NAME clause. If you use an absolute path id to identify the routine body, you must append the .dll extension. For example:

```
CREATE PROCEDURE getSalary( IN inParm INT, OUT outParm INT )
  LANGUAGE c
  PARAMETER STYLE sql
  DYNAMIC RESULT SETS 1
  FENCED THREADSAFE
  RETURNS NULL ON NULL INPUT
  EXTERNAL NAME 'd:\mylib\myfunc.dll'
```

Correction to "Debugging Stored Procedures in Java"

In the section "Preparing to Debug", Chapter 21, you are directed to use the **db2dbugd** command. This is incorrect. Instead, use the following command:

```
idebug -qdaemon -quiport=portno
```

The default port number is 8000. idebug is a client-side daemon of the IBM Distributed Debugger, and ships with VisualAge for Java, and WebSphere Studio Application Developer.

New Requirements for executeQuery and executeUpdate

To comply with the J2EE 1.3 standard, the DB2 JDBC driver, as of FixPak 5, no longer allows the use of a non-query statement with executeQuery, nor a query statement with executeUpdate. Attempting to do so will result in one of the following exceptions:

- CLI0637E QUERY cannot be found,
- CLI0637E UPDATE cannot be found.

If the type of statement is unknown, use execute().

JDBC Driver Support for Additional Methods

CallableStatement.getBlob() and CallableStatement.getClob():

The JDBC driver now supports the methods CallableStatement.getBlob() and CallableStatement.getClob(). Since DB2 does not support LOB locators in stored procedure parameters, enough system memory must be available to hold the maximum possible size of your LOB data, the value specified in CREATE PROCEDURE. An out of memory exception will result if there is not sufficient memory.

This support is unavailable for uncataloged stored procedures.

Statement.setFetchSize(int rows) and ResultSet.setFetchSize(int rows):

The JDBC driver now supports Statement.setFetchSize(int rows) and ResultSet.setFetchSize(int rows). These methods can now be used to improve ResultSet performance.

JDBC and 64-bit systems

JDBC is not supported for instances or clients using 64-bit addressing. This limitation includes systems running 64-bit DB2 UDB Version 7 for AIX 4.3.3, for AIX 5, for Solaris operating systems, and for HP-UX. However, JDBC is supported on 32-bit instances regardless of whether the system is running the 64-bit or 32-bit version of DB2 UDB.

IBM OLE DB Provider for DB2 UDB

For information on using the IBM OLE DB Provider for DB2, refer to http://www.ibm.com/software/data/db2/udb/ad/v71/oledb.html.

CLI Guide and Reference

Binding Database Utilities Using the Run-Time Client

The Run-Time Client cannot be used to bind the database utilities (import, export, reorg, the command line processor) and DB2 CLI bind files to each database before they can be used with that database. You must use the DB2 Administration Client or the DB2 Application Development Client instead.

You must bind these database utilities and DB2 CLI bind files to each database before they can be used with that database. In a network environment, if you are using multiple clients that run on different operating systems, or are at different versions or service levels of DB2, you must bind the utilities once for each operating system and DB2-version combination.

Using Static SQL in CLI Applications

For more information on using static SQL in CLI applications, see the Web page at: http://www.ibm.com/software/data/db2/udb/staticcli/

Limitations of JDBC/ODBC/CLI Static Profiling

JDBC/ODBC/CLI static profiling currently targets straightforward applications. It is not meant for complex applications with many functional components and complex program logic during execution.

An SQL statement must have successfully executed for it to be captured in a profiling session. In a statement matching session, unmatched dynamic statements will continue to execute as dynamic JDBC/ODBC/CLI calls.

An SQL statement must be identical character-by-character to the one that was captured and bound to be a valid candidate for statement matching. Spaces are significant: for example, "COL = 1" is considered different than "COL=1". Use parameter markers in place of literals to improve match hits.

When executing an application with pre-bound static SQL statements, dynamic registers that control the dynamic statement behavior will have no effect on the statements that are converted to static.

If an application issues DDL statements for objects that are referenced in subsequent DML statements, you will find all of these statements in the capture file. The JDBC/ODBC/CLI Static Profiling Bind Tool will attempt to bind them. The bind attempt will be successful with DBMSs that support the

VALIDATE(RUN) bind option, but it fail with ones that do not. In this case, the application should not use Static Profiling.

The Database Administrator may edit the capture file to add, change, or remove SQL statements, based on application-specific requirements.

ADT Transforms

The following supercedes existing information in the book.

- There is a new descriptor type (smallint) SQL_DESC_USER_DEFINED_TYPE_CODE, with values:

    ```
    SQL_TYPE_BASE 0 (this is not a USER_DEFINED_TYPE)
    SQL_TYPE_DISTINCT 1
    SQL_TYPE_STRUCTURED 2
    This value can be queried with either SQLColAttribute
      or SQLGetDescField (IRD only).

    The following attributes are added to obtain the actual type names:
    SQL_DESC_REFERENCE_TYPE
    SQL_DESC_STRUCTURED_TYPE
    SQL_DESC_USER_TYPE
    The above values can be queried using SQLColAttribute
      or SQLGetDescField (IRD only).
    ```

- Add SQL_DESC_BASE_TYPE in case the application needs it. For example, the application may not recognize the structured type, but intends to fetch or insert it, and let other code deal with the details.

- Add a new connection attribute called SQL_ATTR_TRANSFORM_GROUP to allow an application to set the transform group (rather than use the SQL "SET CURRENT DEFAULT TRANSFORM GROUP" statement).

- Add a new statement/connection attribute called SQL_ATTR_RETURN_USER_DEFINED_TYPES that can be set or queried using SQLSetConnectAttr, which causes CLI to return the value SQL_DESC_USER_DEFINED_TYPE_CODE as a valid SQL type. This attribute is required before using any of the transforms.

    - By default, the attribute is off, and causes the base type information to be returned as the SQL type.

    - When enabled, SQL_DESC_USER_DEFINED_TYPE_CODE will be returned as the SQL_TYPE. The application is expected to check for SQL_DESC_USER_DEFINED_TYPE_CODE, and then to retrieve the appropriate type name. This will be available to SQLColAttribute, SQLDescribeCol, and SQLGetDescField.

- The SQLBindParameter does not give an error when you bind SQL_C_DEFAULT, because there is no code to allow SQLBindParameter to specify the type SQL_USER_DEFINED_TYPE. The standard default C types will be used, based on the base SQL type flowed to the server. For example:

```
                    sqlrc = SQLBindParameter (hstmt, 2, SQL_PARAM_INPUT, SQL_C_CHAR,
                                              SQL_VARCHAR, 30, 0, &c2, 30, NULL);
```

Chapter 1. Introduction to CLI

Differences Between DB2 CLI and Embedded SQL

Disregard the third item from the end of the list in the "Advantages of Using DB2 CLI" section. The correct information is as follows:

DB2 CLI provides the ability to retrieve multiple rows and result sets generated from a stored procedure residing on a DB2 Universal Database server, a DB2 for MVS/ESA server (Version 5 or later), or an OS/400 server (Version 5 or later). Support for multiple result sets retrieval on OS/400 requires that PTF (Program Temporary Fix) SI01761 be applied to the server. Contact your OS/400 system administrator to ensure that this PTF has been applied.

Chapter 3. Using Advanced Features

Writing Multi-Threaded Applications

The following should be added to the end of the "Multi-Threaded Mixed Applications" section:

Note: It is recommended that you do not use the default stack size, but instead increase the stack size to at least 256 000. DB2 requires a minimum stack size of 256 000 when calling a DB2 function. You must ensure therefore, that you allocate a total stack size that is large enough for both your application and the minimum requirements for a DB2 function call.

Writing a DB2 CLI Unicode Application

The following is a new section for this chapter.

There are two main areas of support for DB2 CLI Unicode Applications:
1. The addition of a set of functions that can accept Unicode string arguments in place of ANSI string arguments.
2. The addition of new C and SQL data types to describe Unicode data.

The following sections provide more information for both of these areas. To be considered a Unicode application, the application must set the SQL_ATTR_ANSI_APP connection attribute to SQL_AA_FALSE, before a connection is made. This will ensure that the CLI will use Unicode as the preferred method of communication between itself and the database.

Unicode Functions

ODBC API functions have suffixes to indicate the format of their string arguments: those that accept unicode end in W; those that accept ANSI have no suffix.

Note: ODBC adds equivalent functions with names that end in A, but these are not used by DB2 CLI.

The following is a list of those functions that are available in DB2 CLI, which have both ANSI and Unicode Versions.

SQLBrowseConnect	SQLForeignKeys	SQLPrimaryKeys
SQLColAttribute	SQLGetConnectAttr	SQLProcedureColumns
SQLColAttributes	SQLGetConnectOption	SQLProcedures
SQLColumnPrivileges	SQLGetCursorName	SQLSetConnectAttr
SQLColumns	SQLGetDescField	SQLSetConnectOption
SQLConnect	SQLGetDescRec	SQLSetCursorName
SQLDataSources	SQLGetDiagField	SQLSetDescField
SQLDescribeCol	SQLGetDiagRec	SQLSetStmtAttr
SQLDriverConnect	SQLGetInfo	SQLSpecialColumns
SQLGetStmtAttr	SQLStatistics	SQLError
SQLNativeSQL	SQLTablePrivileges	SQLExecDirect
SQLPrepare	SQLTables	

Unicode functions whose arguments are always the length of strings interpret these arguments as count-of-characters. Functions that return length information for server data also describe the display size and precision in terms of characters. When the length (transfer size of the data) could refer to string or nonstring data, the length is interpreted as a count of bytes. For example, SQLGetInfoW will still take the length as count-of-bytes, but SQLExecDirectW will use count-of-characters. CLI will return data from result sets in either Unicode or ANSI, depending on the application's binding. If an application binds to SQL_C_CHAR, the driver will convert SQL_WCHAR data to SQL_CHAR. An ODBC driver manager, if used, maps SQL_C_WCHAR to SQL_C_CHAR for ANSI drivers but does no mapping for Unicode drivers.

New datatypes and Valid Conversions

Additional ODBC and CLI defined data types have been added to accommodate Unicode databases. These types supplement the set of C and SQL types that already exist. The new C type, SQL_C_WCHAR, indicates that the C buffer contains UCS-2 data in native endian format. The new SQL types, SQL_WCHAR, SQL_WVARCHAR, and SQL_WLONGVARCHAR, indicate that a particular column or parameter marker contains Unicode data. For DB2 Unicode databases, graphic columns will be described using the new types.

Table 11. Supported Data Conversions

SQL Data Type	SQL_C_CHAR	SQL_C_WCHAR	SQL_C_LONG	SQL_C_SHORT	SQL_C_TINYINT	SQL_C_FLOAT	SQL_C_DOUBLE	SQL_C_TYPE_DATE	SQL_C_TYPE_TIME	SQL_C_TYPE_TIMESTAMP	SQL_C_BINARY	SQL_C_BIT	SQL_C_DBCHAR	SQL_C_CLOB_LOCATOR	SQL_C_BLOB_LOCATOR	SQL_C_DBCLOB_LOCATOR	SQL_C_BIGINT	SQL_C_NUMERIC
BLOB	X	X									D				X			
CHAR	D	X	X	X	X	X	X	X	X	X	X	X					X	X
CLOB	D	X											X	X				
DATE	X	X						D		X								
DBCLOB		X											X	D		X		
DECIMAL	D	X	X	X	X	X	X				X	X					X	X
DOUBLE	X	X	X	X	X	X	D				X						X	X
FLOAT	X	X	X	X	X	X	D				X						X	X
GRAPHIC (Non-Unicode)	X	X											D					
GRAPHIC (Unicode)	X	X	X	X	X	X	X	X	X	X	X	X	D				X	
INTEGER	X	X	D	X	X	X	X				X						X	X
LONG VARCHAR	D	X									X							
LONG VARGRAPHIC (Non-Unicode)	X	X									X		D					
LONG VARGRAPHIC (Unicode)	X	X									X		D					

Table 11. Supported Data Conversions (continued)

SQL Data Type	SQL_C_CHAR	SQL_C_WCHAR	SQL_C_LONG	SQL_C_SHORT	SQL_C_TINYINT	SQL_C_FLOAT	SQL_C_DOUBLE	SQL_C_TYPE_DATE	SQL_C_TYPE_TIME	SQL_C_TYPE_TIMESTAMP	SQL_C_BINARY	SQL_C_BIT	SQL_C_DBCHAR	SQL_C_CLOB_LOCATOR	SQL_C_BLOB_LOCATOR	SQL_C_DBCLOB_LOCATOR	SQL_C_BIGINT	SQL_C_NUMERIC
NUMERIC	D	X	X	X	X	X	X					X						X
REAL	X	X	X	X	X	D	X					X						X
SMALLINT	X	X	X	D	X	X	X					X					X	X
BIGINT	X	X	X	X	X	X	X				X	X					D	X
TIME	X	X							D	X								
TIMESTAMP	X	X						X	X	D								
VARCHAR	D	X	X	X	X	X	X	X	X	X	X	X					X	X
VARGRAPHIC (Non-Unicode)	X	X											D					
VARGRAPHIC (Unicode)	X	X	X	X	X	X	X	X	X	X	X	X	D				X	

Note:

D Conversion is supported. This is the default conversion for the SQL data type.

X All IBM DBMSs support the conversion.

blank No IBM DBMS supports the conversion.

- Data is not converted to LOB Locator types, rather locators represent a data value, refer to Using Large Objects for more information.

- SQL_C_NUMERIC is only available on 32-bit Windows operating systems.

Obsolete Keyword/Patch Value

Before Unicode applications were supported, applications that were written to work with single-byte character data could be made to work with double-byte graphic data by a series of `cli ini` file keywords, such as GRAPHIC=1,2 or 3, Patch2=7. These workarounds presented graphic data as character data, and also affected the reported length of the data.

These keywords are no longer required for Unicode applications, and should not be used due to the risk of potential side effects. If it is not known if a particular application is a Unicode application, we suggest you try without any of the keywords that affect the handling of graphic data.

Literals in Unicode Databases

In non-unicode databases, data in LONG VARGRAPHIC and LONG VARCHAR columns cannot be compared. Data in GRAPHIC/VARGRAPHIC and CHAR/VARCHAR columns can only be compared, or assigned to each other, using explicit `cast` functions since no implicit code page conversion is supported. This includes GRAPHIC/VARGRAPHIC and CHAR/VARCHAR literals where a GRAPHIC/VARGRAPHIC literal is differentiated from a CHAR/VARCHAR literal by a G prefix.

For Unicode databases, casting between GRAPHIC/VARGRAPHIC and CHAR/VARCHAR literals is not required. Also, a G prefix is not required in front of a GRAPHIC/VARGRAPHIC literal. Provided at least one of the arguments is a literal, implicit conversions occur. This allows literals with or without the G prefix to be used within statements that use either SQLPrepareW() or SQLExecDirect(). Literals for LONG VARGRAPHICs still must have a G prefix.

For more information, see "Casting Between Data Types" in "Chapter 3. Language Elements" of the *SQL Reference*.

New CLI Configuration Keywords

The following three keywords have been added to avoid any extra overhead when Unicode applications connect to a database.

1. **DisableUnicode**

   Keyword Description:
   > Disables the underlying support for Unicode.

   db2cli.ini Keyword Syntax:
   > DisableUnicode = **0** | **1**

   Default Setting:
   > 0 (false)

DB2 CLI/ODBC Settings Tab:
> This keyword cannot be set using the CLI/ODBC Settings notebook. The db2cli.ini file must be modified directly to make use of this keyword.

Usage Notes:

With Unicode support enabled, and when called by a Unicode application, CLI will attempt to connect to the database using the best client code page possible to ensure there is no unnecessary data loss due to code page conversion. This may increase the connection time as code pages are exchanged, or may cause code page conversions on the client that did not occur before this support was added.

Setting this keyword to True (1) will cause all Unicode data to be converted to the application's local code page first, before the data is sent to the server. This can cause data loss for any data that cannot be represented in the local code page.

2. **ConnectCodepage**

Keyword Description:
> Specifies a specific code page to use when connecting to the data source to avoid extra connection overhead.

db2cli.ini Keyword Syntax:
> ConnectCodepage = **0** | **1** | <any valid db2 code page>

Default Setting:
> 0

DB2 CLI/ODBC Settings Tab:
> This keyword cannot be set using the CLI/ODBC Settings notebook. The db2cli.ini file must be modified directly to make use of this keyword.

Usage Notes:

Non-Unicode applications always connect to the database using the application's local code page, or the DB2Codepage environment setting. By default, CLI will ensure that Unicode applications will connect to Unicode databases using UTF-8 and UCS-2 code pages. The default for connecting to non-unicode databases is to use the databases's code page if the database server is running DB2 for Windows, DB2 for Unix or DB2 for OS/2. This ensures that there is no unnecessary data loss due to code page conversion.

This keyword allows the user to specify the database's code page when connecting to a non-Unicode database in order to avoid any extra overhead on the connection.

Specifying a value of 1 causes SQLDriverConnect() to return the correct value in the output connection string, so the value can be used on future SQLDriverConnect() calls.

3. **UnicodeServer**

   Keyword Description:
   > Indicates that the data source is a unicode server. Equivalent to setting ConnectCodepage=1208.

   db2cli.ini Keyword Syntax:
   > UnicodeServer = **0** | **1**

   Default Setting:
   > 0

   DB2 CLI/ODBC Settings Tab:
   > This keyword cannot be set using the CLI/ODBC Settings notebook. The db2cli.ini file must be modified directly to make use of this keyword.

   Usage Notes:

   This keyword is equivalent to ConnectCodepage=1208, and is added only for convenience. Set this keyword to avoid extra connect overhead when connecting to DB2 for OS/390 Version 7 or higher. There is no need to set this keyword for DB2 for Windows, DB2 for Unix or DB2 for OS/2 databases, since there is no extra processing required.

Microsoft Transaction Server (MTS) as Transaction Monitor

The following corrects the default value for the DISABLEMULTITHREAD configuration keyword in the "Installation and Configuration" sub-section:

- DISABLEMULTITHREAD keyword (default 0)

Scrollable Cursors

The following information should be added to the "Scrollable Cursors" section:

Server-side Scrollable Cursor Support for OS/390

The UDB client for the Unix, Windows, and OS/2 platforms supports updatable server-side scrollable cursors when run against OS/390 Version 7 databases. To access an OS/390 scrollable cursor on a three-tier environment, the client and the gateway must be running DB2 UDB Version 7.1, FixPak 3 or later.

There are two application enablement interfaces that can access scrollable cursors: ODBC and JDBC. The JDBC interface can only access static scrollable cursors, while the ODBC interface can access static and keyset-driven server-side scrollable cursors.

Cursor Attributes: The table below lists the default attributes for OS/390 Version 7 cursors in ODBC.

Table 12. Default attributes for OS/390 cursors in ODBC

Cursor Type	Cursor Sensitivity	Cursor Updatable	Cursor Concurrency	Cursor Scrollable
forward-only[a]	unspecified	non-updatable	read-only concurrency	non-scrollable
static	insensitive	non-updatable	read-only concurrency	scrollable
keyset-driven	sensitive	updatable	values concurrency	scrollable
a Forward-only is the default behavior for a scrollable cursor without the FOR UPDATE clause. Specifying FOR UPDATE on a forward-only cursor creates an updatable, lock concurrency, non-scrollable cursor.				

Supported Fetch Orientations: All ODBC fetch orientations are supported via the SQLFetchScroll or SQLExtendedFetch interfaces.

Updating the Keyset-Driven Cursor: A keyset-driven cursor is an updatable cursor. The CLI driver appends the FOR UPDATE clause to the query, except when the query is issued as a SELECT ... FOR READ ONLY query, or if the FOR UPDATE clause already exists. The keyset-driven cursor implemented in DB2 for OS/390 is a values concurrency cursor. A values concurrency cursor results in optimistic locking, where locks are not held until an update or delete is attempted. When an update or delete is attempted, the database server compares the previous values the application retrieved to the current values in the underlying table. If the values match, then the update or delete succeeds. If the values do not match, then the operation fails. If failure occurs, the application should query the values again and re-issue the update or delete if it is still applicable.

An application can update a keyset-driven cursor in two ways:
- Issue an UPDATE WHERE CURRENT OF "<cursor name>" or DELETE WHERE CURRENT OF "<cursor name>" using SQLPrepare() with SQLExecute() or SQLExecDirect().
- Use SQLSetPos() or SQLBulkOperations() to update, delete, or add a row to the result set.

  Note: Rows added to a result set via SQLSetPos() or SQLBulkOperations() are inserted into the table on the server, but are not added to the server's result set. Therefore, these rows are not updatable nor are they sensitive to changes made by other transactions. The inserted rows will appear, however, to be part of the result set, since they are

cached on the client. Any triggers that apply to the inserted rows will appear to the application as if they have not been applied. To make the inserted rows updatable, sensitive, and to see the result of applicable triggers, the application must issue the query again to regenerate the result set.

Troubleshooting for Applications Created Before Scrollable Cursor Support: Since scrollable cursor support is new, some ODBC applications that were working with previous releases of UDB for OS/390 or UDB for Unix, Windows, and OS/2 may encounter behavioral or performance changes. This occurs because before scrollable cursors were supported, applications that requested a scrollable cursor would receive a forward-only cursor. To restore an application's previous behavior before scrollable cursor support, set the following configuration keywords in the db2cli.ini file:

Table 13. Configuration keyword values restoring application behavior before scrollable cursor support

Configuration Keyword Setting	Description
PATCH2=6	Returns a message that scrollable cursors (both keyset-driven and static) are not supported. CLI automatically downgrades any request for a scrollable cursor to a forward-only cursor.
DisableKeysetCursor=1	Disables both the server-side and client-side keyset-driven scrollable cursors. This can be used to force the CLI driver to give the application a static cursor when a keyset-driven cursor is requested.
UseServerKeysetCursor=0	Disables the server-side keyset-driven cursor for applications that are using the client-side keyset-driven cursor library to simulate a keyset-driven cursor. Only use this option when problems are encountered with the server-side keyset-driven cursor, since the client-side cursor incurs a large amount of overhead and will generally have poorer performance than a server-side cursor.

Using Compound SQL

The following note is missing from the book:

```
Any SQL statement that can be prepared dynamically, other than a query,
can be executed as a statement inside a compound statement.

Note: Inside Atomic Compound SQL, savepoint, release savepoint, and
rollback to savepoint SQL statements are also disallowed. Conversely,
Atomic Compound SQL is disallowed in savepoint.
```

Using Stored Procedures

Privileges for building and debugging SQL and Java stored procedures
The following privileges must be granted to users who want to build, debug,
and run SQL stored procedures:

- db2 grant CONNECT on database to userid
- db2 grant IMPLICIT_SCHEMA on database to userid
- db2 grant BINDADD on database to userid
- db2 grant SELECT on SYSIBM.SYSDUMMY1 to userid
- db2 grant SELECT on SYSCAT.PROCEDURES to userid
- db2 grant UPDATE on DB2DBG.ROUTINE_DEBUG to userid

The following privileges must be granted to users who want to build, debug,
and run Java stored procedures:

- db2 grant CONNECT on database to userid
- db2 grant IMPLICIT_SCHEMA on database to userid
- db2 grant BINDADD on database to userid(required only if you build Java
  stored procedures with static SQL using SQLJ)
- db2 grant SELECT on SYSIBM.SYSDUMMY1 to userid
- db2 grant SELECT on SYSCAT.PROCEDURES to userid
- db2 grant UPDATE on DB2DBG.ROUTINE_DEBUG to userid

To create the DB2DBG.ROUTINE_DEBUG table, issue the following
command:

db2 -tf sqllib/misc/db2debug.ddl

For more information about debugging Java stored procedures, see the
Application Development Guide.

Writing a Stored Procedure in CLI
Following is an undocumented limitation on CLI stored procedures:

    If you are making calls to multiple CLI stored procedures,
    the application must close the open cursors from one stored procedure
    before calling the next stored procedure. More specifically, the first
    set of open cursors must be closed before the next stored procedure
    tries to open a cursor.

CLI Stored Procedures and Autobinding
The following supplements information in the book:

The CLI/ODBC driver will normally autobind the CLI packages the first time
a CLI/ODBC application executes SQL against the database, provided the
user has the appropriate privilege or authorization. Autobinding of the CLI

packages cannot be performed from within a stored procedure, and therefore
will not take place if the very first thing an application does is call a CLI
stored procedure. Before running a CLI application that calls a CLI stored
procedure against a new DB2 database, you must bind the CLI packages once
with this command:

UNIX

```
db2 bind <BNDPATH>/@db2cli.lst blocking all
```

Windows and OS/2

```
db2bind "%DB2PATH%\bnd\@db2cli.lst" blocking
```

The recommended approach is to always bind these packages at the time the
database is created to avoid autobind at runtime. Autobind can fail if the user
does not have privilege, or if another application tries to autobind at the same
time.

Chapter 4. Configuring CLI/ODBC and Running Sample Applications

Configuration Keywords

CURRENTFUNCTIONPATH
Disregard the last paragraph in the CURRENTFUNCTIONPATH keyword.
The correct information is as follows:

This keyword is used as part of the process for resolving unqualified function
and stored procedure references that may have been defined in a schema
name other than the current user's schema. The order of the schema names
determines the order in which the function and procedure names will be
resolved. For more information on function and procedure resolution, refer to
the SQL Reference.

SKIPTRACE
The following describes this new configuration keyword:

Keyword Description:
Allows CLI applications to be excluded from the trace function.

db2cli.ini Keyword Syntax:
SKIPTRACE = **0** | **1**

Default Setting:
Do not skip the trace function.

DB2 CLI/ODBC Settings Tab:
This keyword cannot be set using the CLI/ODBC Settings notebook.
The db2cli.ini file must be modified directly to make use of this
keyword.

Usage Notes:
> This keyword can improve performance by allowing the trace
> function to bypass CLI applications. Therefore, if the DB2 trace facility
> db2trc is turned on and this keyword is set to 1, the trace will not
> contain information from the execution of the CLI application.

> Turning SKIPTRACE on is recommended for production environments
> on the UNIX platform where trace information is not required. Test
> environments may benefit, however, from having trace output, so this
> keyword can be turned off (or left at its default setting) when detailed
> execution information is desired.

> SKIPTRACE must be set in the [COMMON] section of the db2cli.ini
> configuration file.

Chapter 5. DB2 CLI Functions

SQLBindFileToParam - Bind LOB File Reference to LOB Parameter

The last parameter - IndicatorValue - in the SQLBindFileToParam() CLI
function is currently documented as "output (deferred)". It should be "input
(deferred)".

SQLColAttribute — Return a Column Attribute

The following updates are additions to the "Description" column for the
SQL_DESC_AUTO_UNIQUE_VALUE and SQL_DESC_UPDATABLE
arguments:

SQL_DESC_AUTO_UNIQUE_VALUE
> SQL_FALSE is returned in *NumericAttributePtr* for all DB2 SQL data
> types. Currently DB2 CLI is not able to determine if a column is an
> identity column, therefore SQL_FALSE is always returned. This
> limitation does not fully conform to the ODBC specifications. Future
> versions of DB2 CLI for Unix and Windows servers will provide
> auto-unique support.

SQL_DESC_UPDATABLE
> Indicates if the column data type is an updateable data type:
> * SQL_ATTR_READWRITE_UNKNOWN is returned in
>   *NumericAttributePtr* for all DB2 SQL data types. It is returned
>   because DB2 CLI is not currently able to determine if a column is
>   updateable. Future versions of DB2 CLI for Unix and Windows
>   servers will be able to determine if a column is updateable.

SQLGetData - Get Data From a Column

The following text replaces the current sentence that appears under the
Explanation column for SQLSTATE 22007 of the SQLSTATEs table for
SQLGetData:

Conversion from a string to a datetime format was indicated, but an invalid string representation or value was specified, or the value was an invalid date.

SQLGetInfo - Get General Information

The following corrects the information in the "Usage" section under "Information Returned by SQLGetInfo":

- The *InfoType* SQL_CURSOR_CLOSE_BEHAVIOR should be SQL_CLOSE_BEHAVIOR.
- The note for SQL_DATABASE_NAME (string) should be as follows:

  Note: This string is the same as that returned by the SELECT CURRENT SERVER statement on non-host systems. For host databases, such as DB2 for OS/390 or DB2 for OS/400, the string returned is the DCS database name that was provided when the CATALOG DCS DATABASE DIRECTORY command was issued at the DB2 Connect gateway.

SQLGetLength - Retrieve Length of A String Value

The following corrects the footnote in "Table 113. SQLGetLength Arguments" :

Note: a This is in characters for DBCLOB data.

SQLNextResult - Associate Next Result Set with Another Statement Handle

The following text should be added to Chapter 5, "DB2 CLI Functions":

Purpose
Specification: DB2 CLI 7.x

Syntax
```
SQLRETURN   SQLNextResult (SQLHSTMT StatementHandle1
                           SQLHSTMT StatementHandle2);
```

Function Arguments

Table 14. SQLNextResult Arguments

Data Type	Argument	Use	Description
SQLHSTMT	*StatementHandle*	input	Statement handle.
SQLHSTMT	*StatementHandle*	input	Statement handle.

Usage
A stored procedure returns multiple result sets by leaving one or more cursors open after exiting. The first result set is always accessed by using the

statement handle that called the stored procedure. If multiple result sets are returned, either SQLMoreResults() or SQLNextResult() can be used to describe and fetch the result set.

SQLMoreResults() is used to close the cursor for the first result set and allow the next result set to be processed, whereas SQLNextResult() moves the next result set to *StatementHandle2*, without closing the cursor on *StatementHandle1*. Both functions return SQL_NO_DATA_FOUND if there are no result sets to be fetched.

Using SQLNextResult() allows result sets to be processed in any order once they have been transferred to other statement handles. Mixed calls to SQLMoreResults() and SQLNextResult() are allowed until there are no more cursors (open result sets) on *StatementHandle1*.

When SQLNextResult() returns SQL_SUCCESS, the next result set is no longer associated with *StatementHandle1*. Instead, the next result set is associated with *StatementHandle2*, as if a call to SQLExecDirect() had just successfully executed a query on *StatementHandle2*. The cursor, therefore, can be described using SQLNumResultSets(), SQLDescribeCol(), or SQLColAttribute().

After SQLNextResult() has been called, the result set now associated with *StatementHandle2* is removed from the chain of remaining result sets and cannot be used again in either SQLNextResult() or SQLMoreResults(). This means that for 'n' result sets, SQLNextResult() can be called successfully at most 'n-1' times.

If SQLFreeStmt() is called with the SQL_CLOSE option, or SQLFreeHandle() is called with *HandleType* set to SQL_HANDLE_STMT, all pending result sets on this statement handle are discarded.

SQLNextResult() returns SQL_ERROR if *StatementHandle2* has an open cursor or *StatementHandle1* and *StatementHandle2* are not on the same connection. If any errors or warnings are returned, SQLError() must always be called on *StatementHandle1*.

Note: SQLMoreResults() also works with a parameterized query with an array of input parameter values specified with SQLParamOptions() and SQLBindParameter(). SQLNextResult(), however, does not support this.

Return Codes
- SQL_SUCCESS
- SQL_SUCCESS_WITH_INFO
- SQL_STILL_EXECUTING
- SQL_ERROR

- SQL_INVALID_HANDLE
- SQL_NO_DATA_FOUND

Diagnostics

Table 15. SQLNextResult SQLSTATEs

SQLSTATE	Description	Explanation
40003 **08**S01	Communication Link failure.	The communication link between the application and data source failed before the function completed.
58004	Unexpected system failure.	Unrecoverable system error.
HY001	Memory allocation failure.	DB2 CLI is unable to allocate the memory required to support execution or completion of the function.
HY010	Function sequence error.	The function was called while in a data-at-execute (SQLParamData(), SQLPutData()) operation. *StatementHandle2* has an open cursor associated with it. The function was called while within a BEGIN COMPOUND and END COMPOUND SQL operation.
HY013	Unexpected memory handling error.	DB2 CLI was unable to access the memory required to support execution or completion of the function.
HYT00	Time-out expired.	The time-out period expired before the data source returned the result set. Time-outs are only supported on non-multitasking systems such as Windows 3.1 and Macintosh System 7. The time-out period can be set using the SQL_ATTR_QUERY_TIMEOUT attribute for SQLSetConnectAttr().

Restrictions
Only SQLMoreResults() can be used for parameterized queries.

References
- "SQLMoreResults - Determine If There Are More Result Sets" on page 535
- "Returning Result Sets from Stored Procedures" on page 120

SQLSetEnvAttr - Set Environment Attribute

The following is an additional environment attribute that belongs in the "Environment Attributes" section under "Usage":

SQL_ATTR_KEEPCTX
> A 32-bit integer value that specifies whether the context should be kept when the environment handle is freed. This attribute should be set at the environment level. It can be used by mutli-threaded

applications to manage contexts associated with each thread's
connections, database resources, and data transmission. The possible
values are:

- **SQL_FALSE:** The application will release the context when a
  thread's environment handle is freed. This is the default value.
- SQL_TRUE: The context will remain valid when a thread's
  environment handle is freed, making the context available for other
  existing threads on the same connection. Setting
  SQL_ATTR_KEEPCTX to SQL_TRUE may resolve some problems
  associated with conflicting contexts in multi-threaded applications.

Note: This is an IBM extension.

SQLSetStmtAttr — Set Options Related to a Statement

The following replaces the existing information for the statement attribute
SQL_ATTR_QUERY_TIMEOUT:

SQL_ATTR_QUERY_TIMEOUT (DB2 CLI v2)

A 32-bit integer value that is the number of seconds to wait for an
SQL statement to execute between returning to the application. This
option can be set and used to terminate long running queries. The
value of 0 means there is no time out. DB2 CLI supports non-zero
values for all platforms that support multithreading.

Appendix C. DB2 CLI and ODBC

The following is a new section added to this appendix.

ODBC Unicode Applications

A Unicode ODBC application sends and retrieves character data primarily in
UCS-2. It does this by calling Unicode versions of the ODBC functions (those
with a 'W' suffix) and by indicating Unicode data types. The application does
not explicitly specify a local code page. The application can still call the ANSI
functions and pass local code page strings.

For example, the application may call SQLConnectW() and pass the DSN,
User ID and Password as Unicode arguments. It may then call
SQLExecDirectW() and pass in a Unicode SQL statement string, and then bind
a combination of ANSI local code page buffers (SQL_C_CHAR) and Unicode
buffers (SQL_C_WCHAR). The database data types may or may not be
Unicode.

If a CLI application calls SQLSetConnectAttr with SQL_ATTR_ANSI_APP set
to SQL_AA_FALSE or calls SQLConnectW without setting the value of
SQL_ATTR_ANSI_APP, then the application is considered a Unicode
application. This means all CHAR data is sent and received from a Unicode

database in UTF-8 format. The application can then fetch CHAR data into SQL_C_CHAR buffers in local code page (with possible data loss), or into SQL_C_WCHAR buffers in UCS-2 without any data loss.

If the application does not do either of the two calls above, CHAR data is converted to the applications local code page at the server. This means CHAR data fetched into SQL_C_WCHAR may suffer data loss.

If the DB2CODEPAGE instance variable is set (using db2set) to code page 1208 (UTF-8), the application will receive all CHAR data in UTF-8 since this is now the local code page. The application must also ensure that all CHAR input data is also in UTF-8. ODBC also assumes that all SQL_C_WCHAR data is in the native endian format. CLI will perform any required byte-reversal for SQL_C_WCHAR.

ODBC Unicode Versus Non-Unicode Applications
This release of DB2 Universal Database contains the SQLConnectW() API. A Unicode driver must export SQLConnectW in order to be recognized as a Unicode driver by the driver manager. It is important to note that many ODBC applications (such as Microsoft Access and Visual Basic) call SQLConnectW(). In previous releases of DB2 Universal Database, DB2 CLI has not supported this API, and thus was not recognized as a Unicode driver by the ODBC driver manager. This caused the ODBC driver manager to convert all Unicode data to the application's local code page. With the added support of the SQLConnectW() function, these applications will now connect as Unicode applications and DB2 CLI will take care of all required data conversion.

DB2 CLI now accepts Unicode APIs (with a suffix of "W") and regular ANSI APIs. ODBC defines a set of functions with a suffix of "A", but the driver manager does not pass ANSI functions with the "A" suffix to the driver. Instead, it converts these functions to ANSI function calls without the suffix, and then passes them to the driver.

An ODBC application that calls the SQLConnectW() API is considered a Unicode application. Since the ODBC driver manager will always call the SQLConnectW() API regardless of what version the application called, ODBC introduced the SQL_ATTR_ANSI_APP connect attribute to notify the driver if the application should be considered an ANSI or UNICODE application. If SQL_ATTR_ANSI_APP is set to SQL_AA_TRUE, the DB2 CLI converts all Unicode data to the local code page before sending it to the server.

Appendix D. Extended Scalar Functions

Date and Time Functions

The following functions are missing from the Date and Time Functions section of Appendix D "Extended Scalar Functions":

DAYOFWEEK_ISO(*date_exp* **)**
> Returns the day of the week in *date_exp* as an integer value in the range 1-7, where 1 represents Monday. Note the difference between this function and the DAYOFWEEK() function, where 1 represents Sunday.

WEEK_ISO(*date_exp* **)**
> Returns the week of the year in *date_exp* as an integer value in the range of 1-53. Week 1 is defined as the first week of the year to contain a Thursday. Therefore, Week1 is equivalent to the first week that contains Jan 4, since Monday is considered to be the first day of the week.
>
> Note that WEEK_ISO() differs from the current definition of WEEK(), which returns a value up to 54. For the WEEK() function, Week 1 is the week containing the first Saturday. This is equivalent to the week containing Jan. 1, even if the week contains only one day.

DAYOFWEEK_ISO() and WEEK_ISO() are automatically available in a database created in Version 7. If a database was created prior to Version 7, these functions may not be available. To make DAYOFWEEK_ISO() and WEEK_ISO() functions available in such a database, use the **db2updb** system command. For more information about **db2updb**, see the "Command Reference" section in these Release Notes.

Appendix K. Using the DB2 CLI/ODBC/JDBC Trace Facility

The sections within this appendix have been updated. See the "Traces" chapter in the *Troubleshooting Guide* for the most up-to-date information on this trace facility.

Message Reference

Update Available

The Message Reference was updated as part of FixPak 4. The latest PDF is available for download online at http://www.ibm.com/software/data/db2/udb/winos2unix/support. All updated documentation is also available on CD. This CD can be ordered through DB2 service using the PTF number U478862. Information on contacting DB2 Service is available at http://www.ibm.com/cgi-bin/db2www/data/db2/udb/winos2unix/support/help.d2w/report.

Message Updates

The following table indicates the messages that have changed since the last publication of the Message Reference. Instructions for viewing the message text online are included in these notes.

Table 16. New and Changed Messages

Message Code	Nature of Change
SQL2554N	New Reason Code
SQL0490N	New Message
SQL20214N	New Message
SQL20211N	New Message

The followng table indicates the SQL states that have changed since the last publication of the Message Reference. Instructions for viewing the message text online are included in these notes.

Table 17. New SQLSTATE Messages

SQLSTATE Value	Meaning
428B7	A number specified in an SQL statement is out of the valid range.
428FI	ORDER OF was specified, but the table-designator does not contain an ORDER BY clause.
428FJ	ORDER BY is not allowed in the outer fullselect of a view or summary table.

Reading Message Text Online

It is assumed that you are familiar with the functions of the operating system where DB2 is installed.

The following DB2 messages are accessible from the operating system command line:

Prefix Description

ASN messages generated by DB2 Replication

CCA messages generated by the Client Configuration Assistant

CLI messages generated by Call Level Interface

DBA messages generated by the Control Center and the Database Administration Utility

DBI messages generated by installation and configuration

DB2 messages generated by the command line processor

DWC messages generated by the Data Warehouse Center

FLG messages and reason codes generated by the Information Catalog Manager

GSE messages generated by the DB2 Spatial Extender

SAT messages generated by DB2 Satellite

SPM messages generated by the sync point manager

SQJ messages generated by Embedded SQL in Java (SQLJ)

SQL messages generated by the database manager when a warning or error condition has been detected.

As well, the message text associated with SQLSTATE values is available on-line.

Message identifiers consist of a three character message prefix (see above list), followed by a four or five digit message number. The single digit letter at the end which describes the severity of the error message is optional.

To access help on these error messages, enter the following at the operating system command prompt:

```
db2 "? XXXnnnnn"

where XXX represents the message prefix
and where nnnnn represents the message number.
```

Note: The message identifier accepted as a parameter of the **db2** command is not case sensitive, and the terminating letter is not required.

Therefore, the following commands will produce the same result:
- db2 "? SQL0000N"
- db2 "? sql0000"
- db2 "? SQL0000n"

If the message text is too long for your screen, use the following command (on unix-based systems and others which support 'more'):

```
db2 "? XXXnnnnn" | more
```

Help can also invoked in the interactive input mode. To enter the interactive input mode, enter the following at the operating system command prompt:

```
db2
```

Once in the interactive input mode, you can enter commands at the following command prompt:

```
db2 =>
```

To get DB2 message help in this mode, type the following at the command prompt:

```
? XXXnnnnn
```

Note: If the message text exceeds the length of the screen, users with non-graphical workstations can pipe the output to the 'more' (on unix-based systems) program or redirect the output to a file which can then be browsed.

The message text associated with a given SQLSTATE value can be retrieved by issuing:

```
db2 "? nnnnn"
```

  or

```
db2 "? nn"
```

where *nnnnn* is a five digit SQLSTATE (alphanumeric) and *nn* is the two digit SQLSTATE class code (first two digits of the SQLSTATE value).

SQL Reference

The SQL Reference has been updated and the latest .pdf is available for download online at http://www.ibm.com/software/data/db2/udb/winos2unix/support. The information in these notes is in addition to the updated reference. All updated documentation is also available on CD. This CD can be ordered through DB2 service using the PTF number U478862. Information on contacting DB2 Service is available at http://www.ibm.com/cgi-bin/db2www/data/db2/udb/winos2unix/support/help.d2w/report.

Enabling the New Functions and Procedures

Version 7 FixPaks deliver new SQL built-in scalar functions. Refer to the *SQL Reference* updates for a description of these new functions. The new functions are not automatically enabled on each database when the database server code is upgraded to the new service level. To enable these new functions, the system administrator must issue the command **db2updv7**, specifying each database at the server. This command makes an entry in the database that ensures that database objects created prior to executing this command use existing function signatures that may match the new function signatures.

For information on enabling the MQSeries functions (those defined in the DB2MQ schema), see "MQSeries" on page 193.

SET SERVER OPTION – Documentation Error

The Notes section for the SET SERVER OPTION statement contains misleading information. The current note reads:

- SET SERVER OPTION currently only supports the password, fold_id, and fold_pw server options.

This statement is not true. SET SERVER OPTION supports all server options, including server options for wrappers not provided by IBM. The note should be ignored.

Correction to CREATE TABLESPACE Container-clause, and Container-string Information

Remote resources (such as LAN-redirected drives or NFS-mounted file systems) are currently supported only when using NEC iStorage S4100 and S2100, Network Appliance Filers, IBM iSCSI, or IBM Network Attached Storage. This is a correction to the current documentation, which indicates that remote resources are unsupported.

Correction to CREATE TABLESPACE EXTENTSIZE information

The CREATE TABLESPACE statement cannot accept an EXTENTSIZE value specified in gigabytes.

GRANT (Table, View, or Nickname Privileges) – Documentation Error

The Notes section for the GRANT (Table, View, or Nickname Privileges) statement contains a misleading bullet. The current note reads:

- DELETE, INSERT, SELECT and UPDATE privileges are not defined for nicknames since operations on nicknames depend on the privileges of the authorization ID used at the data source when the statement referencing the nickname is processed.

This text should be ignored. As the remaining text in the section is accurate without it.

MQSeries Information

Scalar Functions

MQPUBLISH
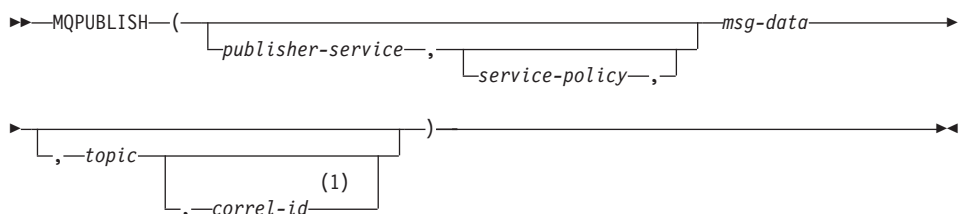
```
►►─MQPUBLISH─(──────────────────────────────────── msg-data ──────►
                └─publisher-service─,─┘
                          └─service-policy─,─┘

►──────────────────────────────)──────────────────────────────────►◄
  └─,─topic────────────────────┘
         └─,─correl-id─────┘
              (1)
```

Notes:

1   The *correl-id* cannot be specified unless a *service* and a *policy* are previously defined.

The schema is DB2MQ.

The MQPUBLISH function publishes data to MQSeries. This function requires the installation of either MQSeries Publish/Subscribe or MQSeries Integrator. Please consult www.ibm.com/software/MQSeries for further details.

The MQPUBLISH function publishes the data contained in *msg-data* to the MQSeries publisher specified in *publisher-service*, and using the quality of service policy defined by *service-policy*. An optional topic for the message can be specified, and an optional user-defined message correlation identifier may also be specified. The function returns a value of '1' if successful or a '0' if unsuccessful.

publisher-service
A string containing the logical MQSeries destination where the message is to be sent. If specified, the *publisher-service* must refer to a publisher Service Point defined in the AMT.XML repository file. A service point is a logical end-point from which a message is sent or received. Service point definitions include the name of the MQSeries Queue Manager and Queue. See the MQSeries Application Messaging Interface for further details. If *publisher-service* is not specified, then the DB2.DEFAULT.PUBLISHER will be used. The maximum size of *publisher-service* is 48 bytes.

service-policy
A string containing the MQSeries AMI Service Policy to be used in handling of this message. If specified, the *service-policy* must refer to a Policy defined in the AMT.XML repository file. A Service Policy defines a set of quality of service options that should be applied to this messaging operation. These options include message priority and message persistence. See the MQSeries Application Messaging Interface manual for further details. If *service-policy* is not specified, then the default DB2.DEFAULT.POLICY will be used. The maximum size of *service-policy* is 48 bytes.

msg-data
A string expression containing the data to be sent via MQSeries. The maximum size if the string of type VARCHAR is 4000 bytes. If the string is a CLOB, it can be up to 1MB in size.

topic
A string expression containing the topic for the message publication. If no topic is specified, none will be associated with the message. The maximum size of *topic* is 40 bytes. Multiple topics can be specified in one string (up to 40 characters long). Each topic must be separated by a colon. For example, "t1:t2:the third topic" indicates that the message is associated with all three topics: t1, t2, and "the third topic".

correl-id
An optional string expression containing a correlation identifier to be associated with this message. The *correl-id* is often specified in request and

reply scenarios to associate requests with replies. If not specified, no correlation id will be added to the message. The maximum size of *correl-id* is 24 bytes.

Examples

Example 1: This example publishes the string ″Testing 123″ to the default publisher service (DB2.DEFAULT.PUBLISHER) using the default policy (DB2.DEFAULT.POLICY). No correlation identifier or topic is specified for the message.
VALUES MQPUBLISH('Testing 123')

Example 2: This example publishes the string ″Testing 345″ to the publisher service ″MYPUBLISHER″ under the topic ″TESTS″. The default policy is used and no correlation identifier is specified.
VALUES MQPUBLISH('MYPUBLISHER','Testing 345', 'TESTS')

Example 3: This example publishes the string ″Testing 678″ to the publisher service ″MYPUBLISHER″ using the policy ″MYPOLICY″ with a correlation identifier of ″TEST1″. The message is published with topic ″TESTS″.
VALUES MQPUBLISH('MYPUBLISHER','MYPOLICY','Testing 678','TESTS','TEST1')

Example 4: This example publishes the string ″Testing 901″ to the publisher service ″MYPUBLISHER″ under the topic ″TESTS″ using the default policy (DB2.DEFAULT.POLICY) and no correlation identifier.
VALUES MQPUBLISH('Testing 901','TESTS')

All examples return the value '1' if successful.

MQREADCLOB

►►──MQREADCLOB──(─────────────────────────────────)──────────────►◄
　　　　　　　　　└─*receive-service*─┬──────────────────┬─┘
　　　　　　　　　　　　　　　　　　　└─,──*service-policy*─┘
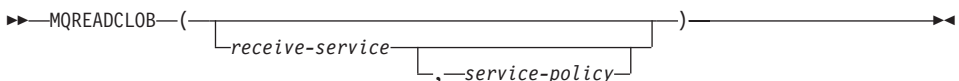
The schema is DB2MQ.

The MQREADCLOB function returns a message from the MQSeries location specified by *receive-service*, using the quality of service policy defined in *service-policy*. Executing this operation does not remove the message from the queue associated with *receive-service*, but instead returns the message at the head of the queue. The return value is a CLOB of 1MB maximum length, containing the message. If no messages are available to be returned, a NULL is returned.

receive-service
> A string containing the logical MQSeries destination from where the message is to be received. If specified, the *receive-service* must refer to a Service Point defined in the AMT.XML repository file. A service point is a logical end-point from where a message is sent or received. Service points definitions include the name of the MQSeries Queue Manager and Queue. See the MQSeries Application Messaging Interface for further details. If *receive-service* is not specified, then the DB2.DEFAULT.SERVICE will be used. The maximum size of *receive-service* is 48 bytes.

service-policy
> A string containing the MQSeries AMI Service Policy used in handling this message. If specified, the *service-policy* must refer to a Policy defined in the AMT.XML repository file. A Service Policy defines a set of quality of service options that should be applied to this messaging operation. These options include message priority and message persistence. See the MQSeries Application Messaging Interface manual for further details. If *service-policy* is not specified, then the default DB2.DEFAULT.POLICY will be used. The maximum size of *service-policy* is 48 bytes.

Examples:

Example 1: This example reads the message at the head of the queue specified by the default service (DB2.DEFAULT.SERVICE), using the default policy (DB2.DEFAULT.POLICY).

VALUES MQREADCLOB()

Example 2: This example reads the message at the head of the queue specified by the service "MYSERVICE" using the default policy (DB2.DEFAULT.POLICY).

VALUES MQREADCLOB('MYSERVICE')

Example 3: This example reads the message at the head of the queue specified by the service "MYSERVICE", and using the policy "MYPOLICY".

VALUES MQREADCLOB('MYSERVICE','MYPOLICY')

All of these examples return the contents of the message as a CLOB with a maximum size of 1MB, if successful. If no messages are available, then a NULL is returned.

MQRECEIVECLOB

▶▶──MQRECEIVECLOB───▶

```
►►─(─┬─────────────────┬──────────────────────────────────────────────)──────────►◄
     └─receive-service─┘  └─,─service-policy─┬───────────────────┘
                                             └─,─correl-id─┘
```
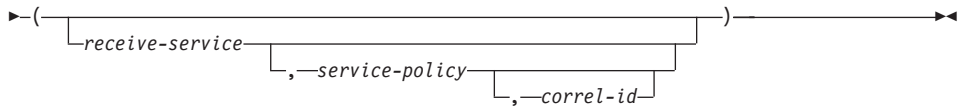
The schema is DB2MQ.

The MQRECEIVECLOB function returns a message from the MQSeries location specified by *receive-service*, using the quality of service policy *service-policy*. Performing this operation removes the message from the queue associated with *receive-service*. If the *correl-id* is specified, then the first message with a matching correlation identifier will be returned. If *correl-id* is not specified, then the message at the head of the queue will be returned. The return value is a CLOB with a maximum length of 1MB containing the message. If no messages are available to be returned, a NULL is returned.

receive-service
A string containing the logical MQSeries destination from which the message is received. If specified, the *receive-service* must refer to a Service Point defined in the AMT.XML repository file. A service point is a logical end-point from which a message is sent or received. Service points definitions include the name of the MQSeries Queue Manager and Queue. See the MQSeries Application Messaging Interface for further details. If *receive-service* is not specified, then the DB2.DEFAULT.SERVICE is used. The maximum size of *receive-service* is 48 bytes.

service-policy
A string containing the MQSeries AMI Service Policy to be used in the handling of this message. If specified, the *service-policy* must refer to a Policy defined in the AMT.XML repository file[1]. If *service-policy* is not specified, then the default DB2.DEFAULT.POLICY is used. The maximum size of *service-policy* is 48 bytes.

correl-id
A string containing an optional correlation identifier to be associated with this message. The *correl-id* is often specified in request and reply scenarios to associate requests with replies. If not specified, no correlation id will be used. The maximum size of *correl-id* is 24 bytes.

Examples:

---

1. A Service Policy defines a set of quality of service options that should be applied to this messaging operation. These options include message priority and message persistence. See the MQSeries Application Messaging Interface manual for further details.

Example 1: This example receives the message at the head of the queue specified by the default service (DB2.DEFAULT.SERVICE), using the default policy (DB2.DEFAULT.POLICY).

VALUES MQRECEIVECLOB()

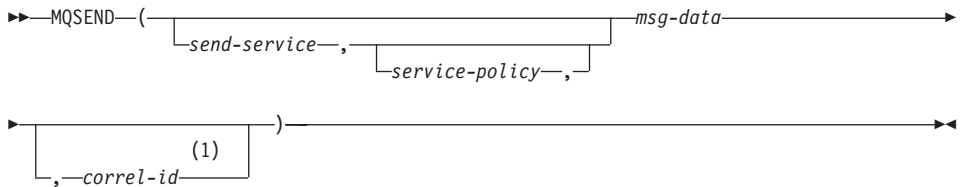Example 2: This example receives the message at the head of the queue specified by the service ″MYSERVICE″ using the default policy (DB2.DEFAULT.POLICY).

VALUES MQRECEIVECLOB('MYSERVICE')

Example 3: This example receives the message at the head of the queue specified by the service ″MYSERVICE″ using the policy ″MYPOLICY″.

VALUES MQRECEIVECLOB('MYSERVICE','MYPOLICY')

Example 4: This example receives the first message with a correlation id that matches '1234' from the head of the queue specified by the service ″MYSERVICE″ using the policy ″MYPOLICY″.

VALUES MQRECEIVECLOB('MYSERVICE',MYPOLICY','1234')

All these examples return the contents of the message as a CLOB with a maximum size of 1MB, if successful. If no messages are available, a NULL will be returned.

MQSEND

```
►►──MQSEND──(─────────────────────────────── msg-data ───────────►
              └─send-service─,─┬──────────────────┬─┘
                               └─service-policy─,─┘

►─┬─────────────────────┬─)─────────────────────────────────────►◄
  │             (1)     │
  └─,─correl-id─────────┘
```

Notes:

1   The *correl-id* cannot be specified unless a *service* and a *policy* are previously defined.

The schema is DB2MQ.

The MQSEND function sends the data contained in *msg-data* to the MQSeries location specified by *send-service*, using the quality of service policy defined by *service-policy*. An optional user defined message correlation identifier may be specified by *correl-id*. The function returns a value of '1' if successful or a '0' if unsuccessful.

msg-data
> A string expression containing the data to be sent via MQSeries. The maximum size is 4000 bytes if the data is of type VARCHAR, and 1MB if the data is of type CLOB.

send-service
> A string containing the logical MQSeries destination where the message is to be sent. If specified, the *send-service* refers to a service point defined in the AMT.XML repository file. A service point is a logical end-point from which a message may be sent or received. Service point definitions include the name of the MQSeries Queue Manager and Queue. See the MQSeries Application Messaging Interface manual for further details. If *send-service* is not specified, then the value of DB2.DEFAULT.SERVICE is used. The maximum size of *send-service* is 48 bytes.

service-policy
> A string containing the MQSeries AMI Service Policy used in handling of this message. If specified, the *service-policy* must refer to a service policy defined in the AMT XML repository file. A Service Policy defines a set of quality of service options that should be applied to this messaging operation. These options include message priority and message persistence. See the MQSeries Application Messaging Interface manual for further details. If *service-policy* is not specified, then a default value of DB2.DEFAULT.POLICY will be used. The maximum size of *service-policy* is 48 bytes.

correl-id
> An optional string containing a correlation identifier associated with this message. The *correl-id* is often specified in request and reply scenarios to associate requests with replies. If not specified, no correlation id will be sent. The maximum size of *correl-id* is 24 bytes.

Examples:

Example 1: This example sends the string "Testing 123" to the default service (DB2.DEFAULT.SERVICE), using the default policy (DB2.DEFAULT.POLICY), with no correlation identifier.

VALUES MQSEND('Testing 123')

Example 2: This example sends the string "Testing 345" to the service "MYSERVICE", using the policy "MYPOLICY", with no correlation identifier.

VALUES MQSEND('MYSERVICE','MYPOLICY','Testing 345')

Example 3: This example sends the string "Testing 678" to the service "MYSERVICE", using the policy "MYPOLICY", with correlation identifier "TEST3".

VALUES MQSEND('MYSERVICE','MYPOLICY','Testing 678','TEST3')

Example 4: This example sends the string "Testing 901" to the service "MYSERVICE", using the default policy (DB2.DEFAULT.POLICY), and no correlation identifier.

VALUES MQSEND('MYSERVICE'**,**'Testing 901')

All examples return a scalar value of '1' if successful.

Table Functions

MQREADALLCLOB

```
►►──MQREADALLCLOB──(──────────────────────────────────────)──►◄
                    └─receive-service─┘          └─num-rows─┘
                         └─,──service-policy─┘
```
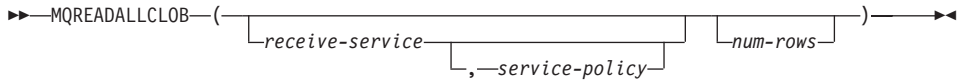
The schema is DB2MQ.

The MQREADALLCLOB function returns a table containing the messages and message metadata from the MQSeries location specified by *receive-service*, using the quality of service policy *service-policy*. Performing this operation does not remove the messages from the queue associated with *receive-service*.

If *num-rows* is specified, then a maximum of *num-rows* messages will be returned. If *num-rows* is not specified, then all available messages will be returned. The table returned contains the following columns:
- MSG - a CLOB column containing the contents of the MQSeries message.
- CORRELID - a VARCHAR(24) column holding a correlation ID used to relate messages.
- TOPIC - a VARCHAR(40) column holding the topic that the message was published with, if available.
- QNAME - a VARCHAR(48) column holding the queue name where the message was received.
- MSGID - a CHAR(24) column holding the assigned MQSeries unique identifier for this message.
- MSGFORMAT - a VARCHAR(8) column holding the format of the message, as defined by MQSeries. Typical strings have an MQSTR format.

receive-service
    A string containing the logical MQSeries destination from which the message is read. If specified, the *receive-service* must refer to a service point defined in the AMT.XML repository file. A service point is a logical end-point from which a message is sent or received. Service point definitions include the name of the MQSeries Queue Manager and Queue. See the MQSeries Application Messaging Interface for further details. If

receive-service is not specified, then the DB2.DEFAULT.SERVICE will be used. The maximum size of *receive-service* is 48 bytes.

service-policy

A string containing the MQSeries AMI Service Policy used in the handling of this message. If specified, the *service-policy* refers to a Policy defined in the AMT XML repository file. A service policy defines a set of quality of service options that should be applied to this messaging operation. These options include message priority and message persistence. See the MQSeries Application Messaging Interface manual for further details. If *service-policy* is not specified, then the default DB2.DEFAULT.POLICY will be used. The maximum size of *service-policy* is 48 bytes.

num-rows

A positive integer containing the maximum number of messages to be returned by the function.

Examples:

Example 1: This example receives all the messages from the queue specified by the default service (DB2.DEFAULT.SERVICE), using the default policy (DB2.DEFAULT.POLICY). The messages and all the metadata are returned as a table.

```
SELECT *
    FROM table (MQREADALLCLOB()) T
```

Example 2: This example receives all the messages from the head of the queue specified by the service MYSERVICE, using the default policy (DB2.DEFAULT.POLICY). Only the MSG and CORRELID columns are returned.

```
SELECT T.MSG, T.CORRELID
    FROM table (MQREADALLCLOB('MYSERVICE')) T
```

Example 3: This example reads the head of the queue specified by the default service (DB2.DEFAULT.SERVICE), using the default policy (DB2.DEFAULT.POLICY). Only messages with a CORRELID of '1234' are returned. All columns are returned.

```
SELECT *
    FROM table (MQREADALLCLOB()) T
    WHERE T.CORRELID = '1234'
```

Example 4: This example receives the first 10 messages from the head of the queue specified by the default service (DB2.DEFAULT.SERVICE), using the default policy (DB2.DEFAULT.POLICY). All columns are returned.

```
SELECT *
    FROM table (MQREADALLCLOB(10)) T
```

MQRECEIVEALLCLOB

```
►►──MQRECEIVEALLCLOB──(──────────────────────────────────────────────────────►

►─┬─────────────────────────────────────────────────────────┬──┬──────────────┬─►
  └─receive-service─┬─────────────────────────────────────┬─┘  └─,──num-rows──┘
                    └─,──service-policy─┬───────────────┬─┘
                                        └─,──correl-id──┘

►─)─────────────────────────────────────────────────────────────────────────►◄
```
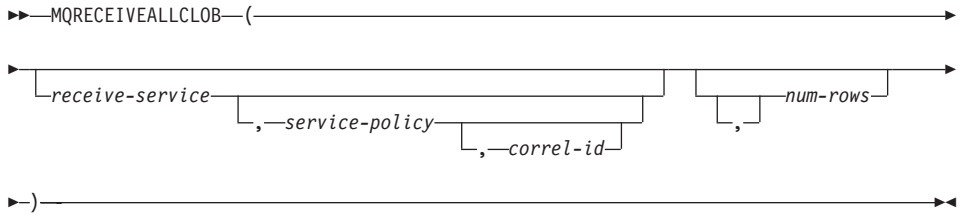
The schema is DB2MQ.

The MQRECEIVEALLCLOB function returns a table containing the messages and message metadata from the MQSeries location specified by *receive-service*, using the quality of service policy *service-policy*. Performing this operation removes the messages from the queue associated with *receive-service*.

If a *correl-id* is specified, then only those messages with a matching correlation identifier will be returned. If *correl-id* is not specified, then the message at the head of the queue will be returned.

If *num-rows* is specified, then a maximum of *num-rows* messages will be returned. If *num-rows* is not specified, then all available messages are returned. The table returned contains the following columns:

- MSG - a CLOB column containing the contents of the MQSeries message.
- CORRELID - a VARCHAR(24) column holding a correlation ID used to relate messages.
- TOPIC - a VARCHAR(40) column holding the topic that the message was published with, if available.
- QNAME - a VARCHAR(48) column holding the queue name where the message was received.
- MSGID - a CHAR(24) column holding the assigned MQSeries unique identifier for this message.
- MSGFORMAT - a VARCHAR(8) column holding the format of the message, as defined by MQSeries. Typical strings have an MQSTR format.

receive-service
> A string containing the logical MQSeries destination from which the message is received. If specified, the *receive-service* must refer to a service point defined in the AMT.XML repository file. A service point is a logical end-point from which a message is sent or received. Service point definitions include the name of the MQSeries Queue Manager and Queue. See the MQSeries Application Messaging Interface manual for further details. If *receive-service* is not specified, then the DB2.DEFAULT.SERVICE will be used. The maximum size of *receive-service* is 48 bytes.

service-policy
> A string containing the MQSeries AMI Service Policy used in the handling of this message. If specified, the *service-policy* refers to a Policy defined in the AMT XML repository file. A service policy defines a set of quality of service options that should be applied to this messaging operation. These options include message priority and message persistence. See the MQSeries Application Messaging Interface manual for further details. If *service-policy* is not specified, then the default DB2.DEFAULT.POLICY will be used. The maximum size of *service-policy* is 48 bytes.

correl-id
> An optional string containing a correlation identifier associated with this message. The *correl-id* is often specified in request and reply scenarios to associate requests with replies. If not specified, no correlation id is specified. The maximum size of correl-id is 24 bytes.

num-rows
> A positive integer containing the maximum number of messages to be returned by the function.

Examples:

Example 1: This example receives all the messages from the queue specified by the default service (DB2.DEFAULT.SERVICE), using the default policy (DB2.DEFAULT.POLICY). The messages and all the metadata are returned as a table.
```
SELECT *
   FROM table (MQRECEIVEALLCLOB()) T
```

Example 2: This example receives all the messages from the head of the queue specified by the service MYSERVICE, using the default policy (DB2.DEFAULT.POLICY). Only the MSG and CORRELID columns are returned.
```
SELECT T.MSG, T.CORRELID
   FROM table (MQRECEIVEALLCLOB('MYSERVICE')) T
```

Example 3: This example receives all of the message from the head of the queue specified by the service "MYSERVICE", using the policy "MYPOLICY". Only messages with a CORRELID of '1234' are returned. Only the MSG and CORRELID columns are returned.
```
SELECT T.MSG, T.CORRELID
   FROM table (MQRECEIVEALLCLOB('MYSERVICE','MYPOLICY','1234')) T
```

Example 4: This example receives the first 10 messages from the head of the queue specified by the default service (DB2.DEFAULT.SERVICE), using the default policy (DB2.DEFAULT.POLICY). All columns are returned.

```
SELECT *
   FROM table (MQRECEIVEALLCLOB(10)) T
```

CLOB data now supported in MQSeries functions

The MQSeries functions (those defined in the DB2MQ schema) now include functionality that allow them to be used with CLOB data in addition to VARCHAR data. In some cases, a new function now exists to handle the CLOB data type, in others, the already existing function now handles both CLOB and VARCHAR data. In either case, the syntax of the CLOB function is identical to that of its VARCHAR equivalent. The functions that support the use of CLOB data, and their equivalent VARCHAR functions, are listed in the following table:

Table 18. MQSeries Functions that support the CLOB data type

Function to use for VARCHAR data	Function to use for CLOB data
MQPUBLISH	MQPUBLISH
MQREAD	MQREADCLOB
MQRECEIVE	MQRECEIVECLOB
MQSEND	MQSEND
MQREADALL	MQREADALLCLOB
MQRECEIVEALL	MQRECEIVEALLCLOB

For information on enabling the MQSeries functions (those defined in the DB2MQ schema), see "MQSeries" on page 193.

Data Type Information

Promotion of Data Types

In this section table 5 shows the precedence list for each data type. Please note:

1. For a Unicode database, the following are considered to be equivalent data types:
   - CHAR and GRAPHIC
   - VARCHAR and VARGRAPHIC
   - LONG VARCHAR and LONG VARGRAPHIC
   - CLOB and DBCLOB

2. In a Unicode database, it is possible to create functions where the only difference in the function signature is between equivalent CHAR and GRAPHIC data types, for example, foo(CHAR(8)) and foo(GRAPHIC(8)). We strongly recommend that you do not define such duplicate functions since migration to a future release will require one of them to be dropped before the migration will proceed.

If such duplicate functions do exist, the choice of which one to invoke is determined by a two-pass algorithm. The first pass attempts to find a match using the same algorithm as is used for resolving functions in a non-Unicode database. If no match is found, then a second pass will be done taking into account the following promotion precedence for CHAR and GRAPHIC strings:

```
GRAPHIC–>CHAR–>VARGRAPHIC–>VARCHAR–>LONG VARGRAPHIC–>LONG VARCHAR–>
  –>DBCLOB–>CLOB
```

Casting between Data Types

The following entry has been added to the list introduced as: "The following casts involving distinct types are supported":

- For a Unicode database, cast from a VARCHAR or VARGRAPHIC to distinct type DT with a source data type CHAR or GRAPHIC.

The following are updates to "Table 6. Supported Casts between Built-in Data Types". Only the affected rows of the table are included.

Table 19. Supported Casts between Built-in Data Types

Target Data Type → Source Data Type ↓	CHAR	VARCHAR	LONGVARCHAR	CLOB	GRAPHIC	VARGRAPHIC	LONGVARGRAPHIC	DBCLOB
CHAR	Y	Y	Y	Y	Y^1	Y^1	-	-
VARCHAR	Y	Y	Y	Y	Y^1	Y^1	-	-
LONGVARCHAR	Y	Y	Y	Y	-	-	Y^1	Y^1
CLOB	Y	Y	Y	Y	-	-	-	Y^1
GRAPHIC	Y^1	Y^1	-	-	Y	Y	Y	Y
VARGRAPHIC	Y^1	Y^1	-	-	Y	Y	Y	Y
LONGVARGRAPHIC	-	-	Y^1	Y^1	Y	Y	Y	Y
DBCLOB	-	-	Y^2	Y^1	Y	Y	Y	Y

[1]   Cast is only supported for Unicode databases.

Assignments and Comparisons

Assignments and comparisons involving both character and graphic data are only supported when one of the strings is a literal. For function resolution, graphic literals and character literals will both match character and graphic function parameters.

The following are updates to "Table 7. Data Type Compatibility for Assignments and Comparisons". Only the affected rows of the table, and the new footnote 6, are included:

Operands	Binary Integer	Decimal Number	Floating Point	Character String	Graphic String	Date	Time	Time-stamp	Binary String	UDT
Character String	No	No	No	Yes	Yes [6]	[1]	[1]	[1]	No [3]	[2]
Graphic String	No	No	No	Yes [6]	Yes	No	No	No	No	[2]

[6]   Only supported for Unicode databases.

String Assignments
Storage Assignment

The last paragraph of this subsection is modified as follows:

When a string is assigned to a fixed-length column and the length of the string is less than the length attribute of the target, the string is padded to the right with the necessary number of single-byte, double-byte, or UCS-2[2] blanks. The pad character is always a blank even for columns defined with the FOR BIT DATA attribute.

Retrieval Assignment

The third paragraph of this subsection is modified as follows:

When a character string is assigned to a fixed-length variable and the length of the string is less than the length attribute of the target, the string is padded to the right with the necessary number of single-byte, double-byte, or UCS-2[2] blanks. The pad character is always a blank even for strings defined with the FOR BIT DATA attribute.

²   UCS-2 defines several SPACE characters with different properties. For a Unicode database, the database manager always uses the ASCII

SPACE at position x'0020' as UCS-2 blank. For an EUC database, the
IDEOGRAPHIC SPACE at position x'3000' is used for padding
GRAPHIC strings.

Conversion Rules for String Assignments

The following paragraph has been added to the end of this subsection:

For Unicode databases, character strings can be assigned to a graphic column,
and graphic strings can be assigned to a character column.

DBCS Considerations for Graphic String Assignments

The first paragraph of this subsection has been modified as follows:

Graphic string assignments are processed in a manner analogous to that for
character strings. For non-Unicode databases, graphic string data types are
compatible only with other graphic string data types, and never with numeric,
character string, or datetime data types. For Unicode databases, graphic string
data types are compatible with character string data types.

String Comparisons
Conversion Rules for Comparison

This subsection has been modified as follows:

When two strings are compared, one of the strings is first converted, if
necessary, to the encoding scheme and code page of the other string. For
details, see the "Rules for String Conversions" section of "Chapter 3.
Language Elements" in the *SQL Reference*.

Rules for Result Data Types

Character and Graphic Strings in a Unicode Database
This is a new subsection inserted after the subsection "Graphic Strings".

In a Unicode database, character strings and graphic strings are compatible.

If one operand is...	And the other operand is...	The data type of the result is...
GRAPHIC(x)	CHAR(y) or GRAPHIC(y)	GRAPHIC(z) where z = max(x,y)
VARGRAPHIC(x)	CHAR(y) or VARCHAR(y)	VARGRAPHIC(z) where z = max(x,y)
VARCHAR(x)	GRAPHIC(y) or VARGRAPHIC	VARGRAPHIC(z) where z = max(x,y)

If one operand is...	And the other operand is...	The data type of the result is...
LONG VARGRAPHIC	CHAR(y) or VARCHAR(y) or LONG VARCHAR	LONG VARGRAPHIC
LONG VARCHAR	GRAPHIC(y) or VARGRAPHIC(y)	LONG VARGRAPHIC
DBCLOB(x)	CHAR(y) or VARCHAR(y) or CLOB(y)	DBCLOB(z) where z = max(x,y)
DBCLOB(x)	LONG VARCHAR	DBCLOB(z) where z = max(x,16350)
CLOB(x)	GRAPHIC(y) or VARGRAPHIC(y)	DBCLOB(z) where z = max(x,y)
CLOB(x)	LONG VARGRAPHIC	DBCLOB(z) where z = max(x,16350)

Rules for String Conversions

The third point has been added to the following list in this section:

For each pair of code pages, the result is determined by the sequential application of the following rules:

- If the code pages are equal, the result is that code page.
- If either code page is BIT DATA (code page 0), the result code page is BIT DATA.
- In a Unicode database, if one code page denotes data in an encoding scheme different from the other code page, the result is UCS-2 over UTF-8 (that is, the graphic data type over the character data type).[1]
- Otherwise, the result code page is determined by Table 8 of the "Rules for String Conversions" section of "Chapter 3. Language Elements" in the *SQL Reference*. An entry of "first" in the table means the code page from the first operand is selected and an entry of "second" means the code page from the second operand is selected.

[1]   In a non-Unicode database, conversion between different encoding schemes is not supported.

Expressions

The following has been added:

In a Unicode database, an expression that accepts a character or graphic string will accept any string types for which conversion is supported.

With the Concatenation Operator

The following has been added to the end of this subsection:

In a Unicode database, concatenation involving both character string operands and graphic string operands will first convert the character operands to graphic operands. Note that in a non-Unicode database, concatenation cannot involve both character and graphic operands.

Predicates

The following entry has been added to the list introduced by the sentence: "The following rules apply to all types of predicates":

- In a Unicode database, all predicates that accept a character or graphic string will accept any string types for which conversion is supported.

Unicode Information

Scalar Functions and Unicode

In a Unicode database, all scalar functions that accept a character or graphic string will accept any string types for which conversion is supported.

GRAPHIC type and DATE/TIME/TIMESTAMP compatibility

In the following sections, references to datetime values having "character string" representations have been changed to "string" representations. DB2 now supports, for Unicode databases only, "graphic string" representations of datetime values.

String representations of datetime values

Values whose data types are DATE, TIME, or TIMESTAMP are represented in an internal form that is transparent to the user. Date, time, and timestamp values can, however, also be represented by strings. This is useful because there are no constants or variables whose data types are DATE, TIME, or TIMESTAMP. Before it can be retrieved, a datetime value must be assigned to a string variable. The CHAR function or the GRAPHIC function (for Unicode databases only) can be used to change a datetime value to a string representation. The string representation is normally the default format of datetime values associated with the country/region code of the database, unless overridden by specification of the DATETIME option when the program is precompiled or bound to the database.

No matter what its length, a large object string, a LONG VARCHAR value, or a LONG VARGRAPHIC value cannot be used to represent a datetime value (SQLSTATE 42884).

When a valid string representation of a datetime value is used in an operation with an internal datetime value, the string representation is converted to the internal form of the date, time, or timestamp value before the operation is performed.

Date, time and timestamp strings must contain only characters and digits.

Date strings, time strings, and datetime strings
The definitions of these terms have been changed slightly. References to "character string" representations have been changed to "string" representations.

Casting between data types

DATE, TIME, and TIMESTAMP can now be cast to GRAPHIC and VARGRAPHIC. GRAPHIC and VARGRAPHIC can now be cast to DATE, TIME, and TIMESTAMP. Graphic string support is only available for Unicode databases.

Assignments and comparisons

There is now data type compatibility for assignments and comparisons between graphic strings and DATE, TIME, and TIMESTAMP values. Graphic string support is only available for Unicode databases.

Datetime assignments

The basic rule for datetime assignments is that a DATE, TIME, or TIMESTAMP value can only be assigned to a column with a matching data type (whether DATE, TIME, or TIMESTAMP) or to a fixed- or varying-length string variable or string column. The assignment must not be to a LONG VARCHAR, CLOB, LONG VARGRAPHIC, DBCLOB, or BLOB variable or column.

When a datetime value is assigned to a string variable or string column, conversion to a string representation is automatic. Leading zeros are not omitted from any part of the date, time, or timestamp. The required length of the target will vary, depending on the format of the string representation. If the length of the target is greater than required, and the target is a fixed-length string, it is padded on the right with blanks. If the length of the target is less than required, the result depends on the type of datetime value involved, and on the type of target.

When the target is a host variable, the following rules apply:

- **DATE:** If the variable length is less than 10 characters, an error occurs.
- **TIME:** If the USA format is used, the length of the variable must not be less than 8 characters; in other formats the length must not be less than 5 characters.

If ISO or JIS formats are used, and if the length of the host variable is less than 8 characters, the seconds part of the time is omitted from the result and assigned to the indicator variable, if provided. The SQLWARN1 field of the SQLCA is set to indicate the omission.

- **TIMESTAMP:** If the host variable is less than 19 characters, an error occurs. If the length is less than 26 characters, but greater than or equal to 19 characters, trailing digits of the microseconds part of the value are omitted. The SQLWARN1 field of the SQLCA is set to indicate the omission.

DATE

►►──DATE─(─*expression*─)──►◄

The schema is SYSIBM.

The DATE function returns a date from a value.

The argument must be a date, timestamp, a positive number less than or equal to 3 652 059, a valid string representation of a date or timestamp, or a string of length 7 that is not a LONG VARCHAR, CLOB, LONG VARGRAPHIC, DBCLOB, or BLOB.

Only Unicode databases support an argument that is a graphic string representation of a date or a timestamp.

If the argument is a string of length 7, it must represent a valid date in the form *yyyynnn*, where *yyyy* are digits denoting a year, and *nnn* are digits between 001 and 366, denoting a day of that year.

The result of the function is a date. If the argument can be null, the result can be null; if the argument is null, the result is the null value.

The other rules depend on the data type of the argument:
- If the argument is a date, timestamp, or valid string representation of a date or timestamp:
  - The result is the date part of the value.
- If the argument is a number:
  - The result is the date that is *n*-1 days after January 1, 0001, where *n* is the integral part of the number.
- If the argument is a string with a length of 7:
  - The result is the date represented by the string.

Examples:

Assume that the column RECEIVED (timestamp) has an internal value equivalent to '1988-12-25-17.12.30.000000'.

- This example results in an internal representation of '1988-12-25'.
    DATE(RECEIVED)
- This example results in an internal representation of '1988-12-25'.
    DATE('1988-12-25')
- This example results in an internal representation of '1988-12-25'.
    DATE('25.12.1988')
- This example results in an internal representation of '0001-02-04'.
    DATE(35)

GRAPHIC

►►──GRAPHIC──(──*graphic-expression*──┬──────────────┬──)──────────────────►◄
                                       └──,──*integer*──┘

The schema is SYSIBM.

The GRAPHIC function returns a GRAPHIC representation of a graphic string type or a GRAPHIC representation of a datetime type.

graphic-expression
> An *expression* that returns a value that is a graphic string.

integer
> An integer value specifying the length attribute of the resulting GRAPHIC data type. The value must be between 1 and 127. If *integer* is not specified, the length of the result is the same as the length of the first argument.

The result of the function is a GRAPHIC. If the argument can be null, the result can be null; if the argument is null, the result is the null value.

Datetime to Graphic:

►►──GRAPHIC──(──*datetime-expression*──┬──────────────┬──)────────────────►◄
                                       └──,──┬──ISO───┬──┘
                                             ├──USA───┤
                                             ├──EUR───┤
                                             ├──JIS───┤
                                             └──LOCAL─┘

Datetime to Graphic

> *datetime-expression*
> > An expression that is one of the following three data types
>
> > **date**   The result is the graphic string representation of the date

in the format specified by the second argument. The length of the result is 10. An error occurs if the second argument is specified and is not a valid value (SQLSTATE 42703).

time The result is the graphic string representation of the time in the format specified by the second argument. The length of the result is 8. An error occurs if the second argument is specified and is not a valid value (SQLSTATE 42703).

timestamp
The second argument is not applicable and must not be specified (SQLSTATE 42815). The result is the graphic string representation of the timestamp. The length of the result is 26.

The code page of the string is the code page of the database at the application server.

TIME

```
►►──TIME──(──expression──)──────────────────────────────────────────►◄
```

The schema is SYSIBM.

The TIME function returns a time from a value.

The argument must be a time, timestamp, or a valid string representation of a time or timestamp that is not a LONG VARCHAR, CLOB, LONG VARGRAPHIC, DBCLOB, or BLOB.

Only Unicode databases support an argument that is a graphic string representation of a time or a timestamp.

The result of the function is a time. If the argument can be null, the result can be null; if the argument is null, the result is the null value.

The other rules depend on the data type of the argument:
• If the argument is a time:
  – The result is that time.
• If the argument is a timestamp:
  – The result is the time part of the timestamp.
• If the argument is a string:
  – The result is the time represented by the string.

Example:

- Select all notes from the IN_TRAY sample table that were received at least one hour later in the day (any day) than the current time.

  ```
  SELECT * FROM IN_TRAY
    WHERE TIME(RECEIVED) >= CURRENT TIME + 1 HOUR
  ```

TIMESTAMP

```
►►──TIMESTAMP──(──expression──┬─────────────────┬──)────────────────────────►◄
                              └─,expression─────┘
```

The schema is SYSIBM.

The TIMESTAMP function returns a timestamp from a value or a pair of values.

Only Unicode databases support an argument that is a graphic string representation of a date, a time, or a timestamp.

The rules for the arguments depend on whether the second argument is specified.

- If only one argument is specified:
  - It must be a timestamp, a valid string representation of a timestamp, or a string of length 14 that is not a LONG VARCHAR, CLOB, LONG VARGRAPHIC, DBCLOB, or BLOB.

    A string of length 14 must be a string of digits that represents a valid date and time in the form *yyyyxxddhhmmss*, where *yyyy* is the year, *xx* is the month, *dd* is the day, *hh* is the hour, *mm* is the minute, and *ss* is the seconds.
- If both arguments are specified:
  - The first argument must be a date or a valid string representation of a date and the second argument must be a time or a valid string representation of a time.

The result of the function is a timestamp. If either argument can be null, the result can be null; if either argument is null, the result is the null value.

The other rules depend on whether the second argument is specified:

- If both arguments are specified:
  - The result is a timestamp with the date specified by the first argument and the time specified by the second argument. The microsecond part of the timestamp is zero.
- If only one argument is specified and it is a timestamp:
  - The result is that timestamp.

- If only one argument is specified and it is a string:
  - The result is the timestamp represented by that string. If the argument is a string of length 14, the timestamp has a microsecond part of zero.

Example:

- Assume the column START_DATE (date) has a value equivalent to 1988-12-25, and the column START_TIME (time) has a value equivalent to 17.12.30.

    TIMESTAMP(START_DATE, START_TIME)

  Returns the value '1988-12-25-17.12.30.000000'.

VARGRAPHIC

Character to Vargraphic:

▶▶──VARGRAPHIC──(──*character-string-expression*──)──────────────────────────▶◀

Datetime to Vargraphic:

▶▶──VARGRAPHIC──(──*datetime-expression*──)───────────────────────────────────▶◀

Graphic to Vargraphic:

▶▶──VARGRAPHIC──(──*graphic-string-expression*──┬──────────────┬──)──────────▶◀
                                                 └─,──*integer*─┘

The schema is SYSIBM.

The VARGRAPHIC function returns a graphic string representation of a:

- character string value, converting single byte characters to double byte characters,
- datetime value (only supported on Unicode databases)
- graphic string value, if the first argument is any type of graphic string.

The result of the function is a varying length graphic string (VARGRAPHIC data type). If the first argument can be null, the result can be null; if the first argument is null, the result is the null value.

Character to Vargraphic

character-string-expression
    An expression whose value must be of a character string data type other than LONG VARCHAR or CLOB, and whose maximum length must not be greater than 16 336 bytes.

The length attribute of the result is equal to the length attribute of the argument.

Let S denote the value of the *character-string-expression*. Each single-byte character in S is converted to its equivalent double-byte representation or to the double-byte substitution character in the result; each double-byte character in S is mapped 'as-is'. If the first byte of a double-byte character appears as the last byte of S, it is converted into the double-byte substitution character. The sequential order of the characters in S is preserved.

The following are additional considerations for the conversion.

- For a Unicode database, this function converts the character string from the code page of the operand into UCS-2. Every character of the operand, including DBCS characters, is converted. If the second argument is given, it specifies the desired length (number of UCS-2 characters) of the resulting UCS-2 string.
- The conversion to double-byte code points by the VARGRAPHIC function is based on the code page of the operand.
- Double-byte characters of the operand are not converted. All other characters are converted to their corresponding double-byte depiction. If there is no corresponding double-byte depiction, the double-byte substitution character for the code page is used.
- No warning or error code is generated if one or more double-byte substitution characters are returned in the result.

Datetime to Vargraphic

datetime-expression
   An expression whose value must be of the DATE, TIME, or TIMESTAMP data type.

Graphic to Vargraphic

graphic-string-expression
   An expression that returns a value that is a graphic string.

integer
   The length attribute for the resulting varying length graphic string. The value must be between 0 and 16 336. If this argument is not specified, the length of the result is the same as the length of the argument.

If the length of the *graphic-string-expression* is greater than the length attribute of the result, truncation is performed and a warning is returned (SQLSTATE 01004), unless the truncated characters were all blanks and the *graphic-string-expression* was not a long string (LONG VARGRAPHIC or DBCLOB).

Larger Index Keys for Unicode Databases

ALTER TABLE

The length of variable length columns that are part of any index, including primary and unique keys, defined when the registry variable DB2_INDEX_2BYTEVARLEN was on, can be altered to a length greater than 255 bytes. The fact that a variable length column is involved in a foreign key will no longer prevent the length of that column from being altered to larger than 255 bytes, regardless of the registry variable setting. However, data with length greater than 255 cannot be inserted into the table unless the column in the corresponding primary key has length greater than 255 bytes, which is only possible if the primary key was created with the registry variable ON.

CREATE INDEX

Indexes can be defined on variable length columns whose length is greater than 255 bytes if the registry variable DB2_INDEX_2BYTEVARLEN is ON.

CREATE TABLE

Primary and unique keys with variable keyparts can have a size greater than 255 if the registry variable DB2_INDEX_2BYTEVARLEN is ON. Foreign keys can be defined on variable length columns whose length is greater than 255 bytes.

ALLOCATE CURSOR Statement Notes Section Incorrect

The two bulleted items in the Notes section of the ALLOCATE CURSOR Statement were printed in error. Disregard the information contained in these items.

Additional Options in the GET DIAGNOSTICS Statement

GET DIAGNOSTICS Statement

The GET DIAGNOSTICS statement is used to obtain information about the previously executed SQL statement. The syntax of this statement has been updated as follows.

Command Syntax

```
►►──GET DIAGNOSTICS──┬─SQL-variable-name──=──┬─ROW_COUNT──────┬──────────────────►◄
                     │                       └─RETURN_STATUS──┘
                     └─┤ condition-information ├─┘
```

condition-information:

```
                      ┌─,──────────────────────────────────────┐
├──EXCEPTION──1──▼─SQL-variable-name──=──┬─MESSAGE_TEXT──────┬──┴──────────────┤
                                         └─DB2_TOKEN_STRING──┘
```
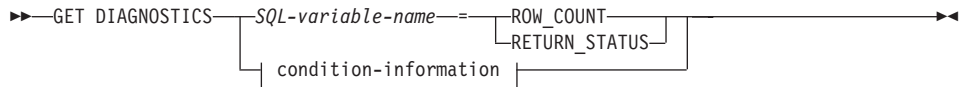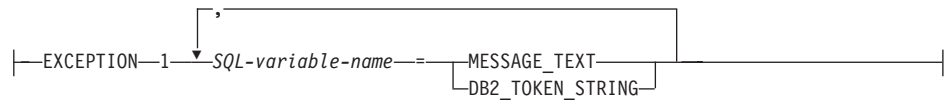
Command Parameters

SQL-variable-name
> Identifies the variable that is the assignment target. If ROW_COUNT or RETURN_STATUS is specified, the variable must be an integer variable. Otherwise, the variable must be CHAR or VARCHAR. SQL variables can be defined in a compound statement.

ROW_COUNT
> Identifies the number of rows associated with the previous SQL statement. If the previous SQL statement is a DELETE, INSERT, or UPDATE statement, ROW_COUNT identifies the number of rows deleted, inserted, or updated by that statement, excluding rows affected by triggers or referential integrity constraints. If the previous statement is a PREPARE statement, ROW_COUNT identifies the *estimated* number of result rows in the prepared statement.

RETURN_STATUS
> Identifies the status value returned from the stored procedure associated with the previously executed SQL statement, provided that the statement was a CALL statement invoking a procedure that returns a status. If the previous statement is not such a statement, then the value returned has no meaning and could be any integer.

condition-information
> Specifies that the error or warning information for the previously executed SQL statement is to be returned. If information about an error is needed, the GET DIAGNOSTICS statement must be the first statement specified in the handler that will handle the error. If information about a warning is needed, and if the handler will get control of the warning condition, the

GET DIAGNOSTICS statement must be the first statement specified in that handler. If the handler will *not* get control of the warning condition, the GET DIAGNOSTICS statement must be the next statement executed.

MESSAGE_TEXT

Identifies any error or warning message text returned from the previously executed SQL statement. The message text is returned in the language of the database server where the statement is processed. If the statement completes with an SQLCODE of zero, an empty string or blanks are returned.

DB2_TOKEN_STRING

Identifies any error or warning message tokens returned from the previously executed SQL statement. If the statement completes with an SQLCODE of zero, or if the SQLCODE has no tokens, then an empty string or blanks is returned.

ORDER BY in Subselects

DB2 now supports ORDER BY in subselects and fullselects.

fullselect

Following is a partial syntax diagram of the modified fullselect showing the location of the order-by-clause.

```
>>─┬─subselect─────┬─┬───────────────────┬─┬─subselect─────┬─>
   ├─(fullselect)──┤ │ ┌───────────────┐ │ ├─(fullselect)──┤
   └─ values-clause ┘ ▼ ├─UNION─────────┤ │ └─ values-clause ┘
                        ├─UNION ALL─────┤
                        ├─EXCEPT────────┤
                        ├─EXCEPT ALL────┤
                        ├─INTERSECT─────┤
                        └─INTERSECT ALL─┘

>─┬──────────────────┬─><
  └─order-by-clause──┘
```
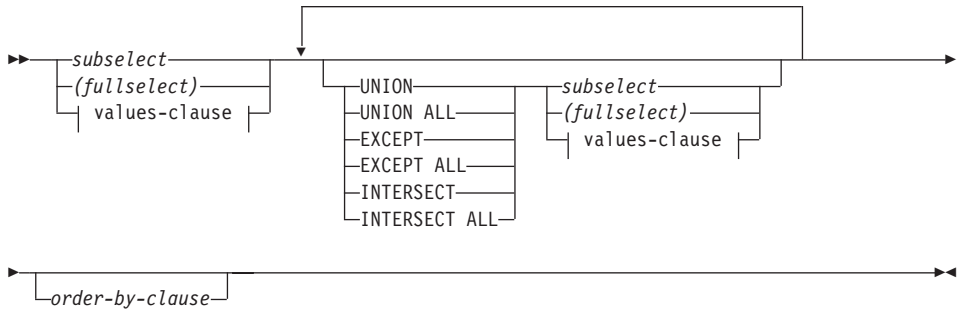
A fullselect that contains an ORDER BY clause cannot be specified in:

- A summary table
- The outermost fullselect of a view (SQLSTATE 428FJ SQLCODE -20211)

An ORDER BY clause in a fullselect does not affect the order of the rows returned by a query. An ORDER BY clause only affects the order of the rows returned if it is specified in the outermost fullselect.

subselect

Following is the complete syntax diagram of the modified subselect showing the location of the order-by-clause.

```
>>──select-clause──from-clause──┬────────────────┬─┬─────────────────┬─>
                                └─where-clause───┘ └─group-by-clause─┘

>─┬────────────────┬─┬─────────────────┬─><
  └─having-clause──┘ └─order-by-clause─┘
```
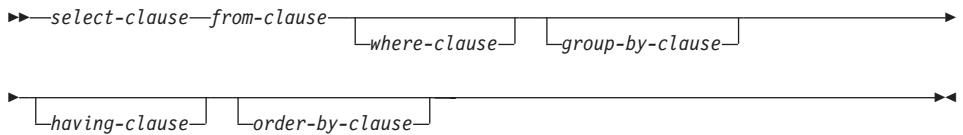
The clauses of the subselect are processed in the following sequence:

1. FROM clause
2. WHERE clause
3. GROUP BY clause
4. HAVING clause
5. SELECT clause

6. ORDER BY clause

A subselect that contains an ORDER BY cannot be specified:

- In the outermost fullselect of a view
- In a summary table
- Unless the subselect is enclosed in parentheses

For example, the following is not valid (SQLSTATE 428FJ SQLCODE -20211):

```
SELECT * FROM T1
   ORDER BY C1
UNION
SELECT * FROM T2
   ORDER BY C1
```

The following example *is* valid:

```
(SELECT * FROM T1
   ORDER BY C1)
UNION
(SELECT * FROM T2
   ORDER BY C1)
```

An ORDER BY clause in a subselect does not affect the order of the rows returned by a query. An ORDER BY clause only affects the order of the rows returned if it is specified in the outermost fullselect.

order-by-clause

Following is the complete syntax diagram of the modified order-by-clause.
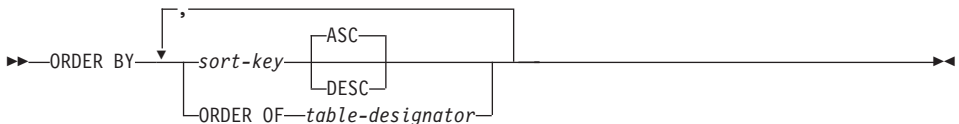
sort-key:

ORDER OF *table-designator*

Specifies that the same ordering used in *table-designator* should be applied to the result table of the subselect. There must be a table reference matching *table-designator* in the FROM clause of the subselect that specifies this clause (SQLSTATE 42703). The subselect (or fullselect) corresponding to the specified *table-designator* must include an ORDER BY clause that is dependant on the data (SQLSTATE 428FI SQLCODE -20210). The ordering

that is applied is the same as if the columns of the ORDER BY clause in the nested subselect (or fullselect) were included in the outer subselect (or fullselect), and these columns were specified in place of the ORDER OF clause. For more information on table designators, see "Column Name Qualifiers to Avoid Ambiguity" in the *SQL Reference*.

Note that this form is not allowed in a fullselect (other than the degenerative form of a fullselect). For example, the following is not valid:

```
(SELECT C1 FROM T1
   ORDER BY C1)
UNION
SELECT C1 FROM T2
   ORDER BY ORDER OF T1
```

The following example *is* valid:

```
SELECT C1 FROM
   (SELECT C1 FROM T1
      UNION
    SELECT C1 FROM T2
    ORDER BY C1 ) AS UTABLE
ORDER BY ORDER OF UTABLE
```

select-statement

Following is the complete syntax diagram of the modified select-statement:

```
►►─┬──────────────────────────────────┬──fullselect──fetch-first-clause──►
   │      ┌─,──────────────────────┐   │
   └─WITH─▼─common-table-expression─┴──┘

►──●─┬─────────────────────┬──●─┬──────────────────────┬──●─┬──────────────────┬──►◄
     ├─read-only-clause────┤    └─optimize-for-clause──┘    └─WITH─┬─RR─┬──────┘
     │           (1)       │                                       ├─RS─┤
     └─update-clause───────┘                                       ├─CS─┤
                                                                   └─UR─┘
```
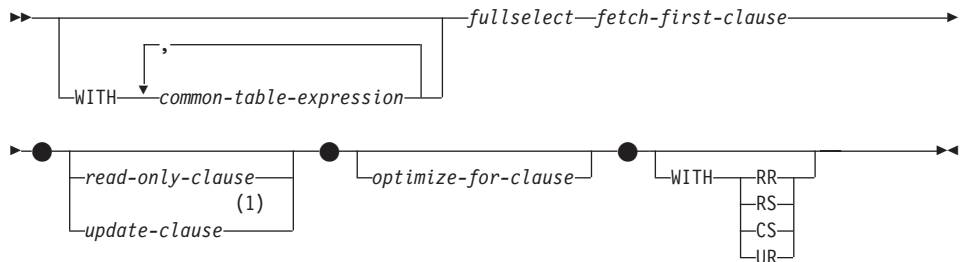
Notes:

1 The update-clause cannot be specified if the fullselect contains an order-by-clause.

SELECT INTO statement

Syntax

```
>>--select-clause--INTO----host-variable----from-clause-------------------------->
                       ^------,--------|                 |--where-clause--|

>--------------------------------------------------------------------------><
    |--group-by-clause--|  |--having-clause--|  |--order-by-clause--|  |--WITH----RR----|
                                                                                |--RS--|
                                                                                |--CS--|
                                                                                |--UR--|
```
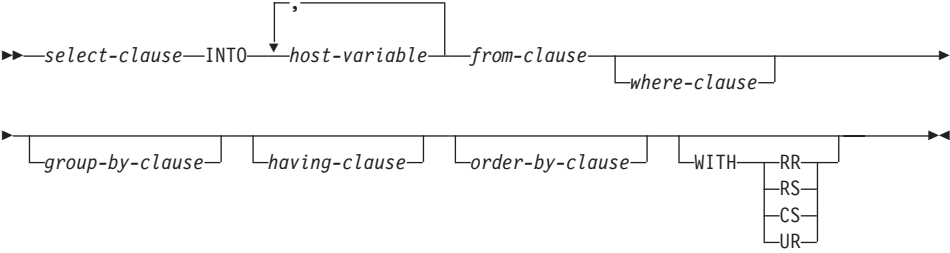
OLAP Functions (window-order-clause)

Following is a partial syntax diagram for the OLAP functions showing the modified window-order-clause.

window-order-clause:
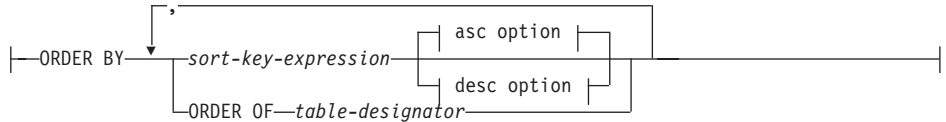
```
                        ,
                   ┌─────────────────┐
                   ▼                 │    ┌─ asc option ─┐
├──ORDER BY──┬───sort-key-expression─┴────┤              ├──────────────────────┤
            │                             └─ desc option ┘
            └──ORDER OF──table-designator──────────────────────────────────────
```

asc option:

```
            ┌─NULLS LAST──┐
├──ASC──────┤             ├───────────────────────────────────────────────────┤
            └─NULLS FIRST─┘
```

desc option:

```
            ┌─NULLS FIRST─┐
├──DESC─────┤             ├───────────────────────────────────────────────────┤
            └─NULLS LAST──┘
```

ORDER BY (*sort-key-expression***,...)**
> Defines the ordering of rows within a partition that determines the value of the OLAP function or the meaning of the ROW values in the window-aggregation-group-clause (it does not define the ordering of the query result set).

sort-key-expression
> An expression used in defining the ordering of the rows within a window partition. Each column name referenced in a sort-key-expression must unambiguously reference a column of the result set of the subselect, including the OLAP function (SQLSTATE 42702 or 42703). The length of each sort-key-expression must not be more than 255 bytes (SQLSTATE 42907). A sort-key-expression cannot include a scalar fullselect (SQLSTATE 42822) or any function that is not deterministic or that has an external action (SQLSTATE 42845). This clause is required for the RANK and DENSE_RANK functions (SQLSTATE 42601).

ASC
> Uses the values of the sort-key-expression in ascending order.

DESC
> Uses the values of the sort-key-expression in descending order.

NULLS FIRST
The window ordering considers null values *before* all non-null values in the sort order.

NULLS LAST
The window ordering considers null values *after* all non-null values in the sort order.

ORDER OF *table-designator*
Specifies that the same ordering used in *table-designator* should be applied to the result table of the subselect. There must be a table reference matching *table-designator* in the FROM clause of the subselect that specifies this clause (SQLSTATE 42703). The subselect (or fullselect) corresponding to the specified *table-designator* must include an ORDER BY clause that is dependent on the data (SQLSTATE 428FI SQLCODE -20210). The ordering that is applied is the same as if the columns of the ORDER BY clause in the nested subselect (or fullselect) were included in the outer subselect (or fullselect), and these columns were specified in place of the ORDER OF clause. For more information on table designators, see "Column Name Qualifiers to Avoid Ambiguity" in the *SQL Reference*.

New Input Argument for the GET_ROUTINE_SAR Procedure

This procedure now supports *hide_body_flag*, an input argument of type INTEGER that specifies (using one of the following values) whether or not the routine body should be hidden when the routine text is extracted from the catalogs:

0   Leave the routine text intact. This is the default value.

1   Replace the routine body with an empty body when the routine text is extracted from the catalogs.

```
►►──GET_ROUTINE_SAR──────────────────────────────────────────────────►

►─(──sarblob──,──type──,──routine_name_string───────────────────)──────►◄
                                                └─,──hide_body_flag─┘
```

Required Authorization for the SET INTEGRITY Statement

When this statement is used to *turn off* integrity checking, the privileges of the authorization ID of the statement must include at least one of the following:

- CONTROL privilege on:
  - The specified tables, and
  - The descendent foreign key tables that will have integrity checking turned off by the statement, and
  - The descendent immediate summary tables that will have integrity checking turned off by the statement
- SYSADM or DBADM authority
- LOAD authority

Appendix N. Exception Tables

In the table "Exception Table Message Column Structure", in rows 2 and 6, which describe the characters that indicate the type of the first and the next constraint violations found, respectively, there is a missing reference to:

```
'D' - Delete Cascade violation
```

Unicode Updates

Introduction

The Unicode standard is a universal character encoding scheme for written characters and text. It defines a character set very precisely, as well as a small number of encodings for it. It defines a consistent way of encoding multilingual text that enables the exchange of text data internationally and creates the foundation for global software.

Two of the encoding schemes provided by Unicode are UTF-16 and UTF-8.

The default encoding scheme is UTF-16, which is a 16-bit encoding format. UCS-2 is a subset of UTF-16 which uses two bytes to represent a character. UCS-2 is generally accepted as the universal code page capable of representing all the necessary characters from all existing single and double byte code pages. UCS-2 is registered in IBM as code page 1200.

The other Unicode encoding format is UTF-8, which is byte-oriented and has been designed for ease of use with existing ASCII-based systems. UTF-8 uses a varying number of bytes (usually 1-3, sometimes 4) to store each character. The invariant ASCII characters are stored as single bytes. All other characters are stored using multiple bytes. In general, UTF-8 data can be treated as extended ASCII data by code that was not designed for multi-byte code pages. UTF-8 is registered in IBM as code page 1208.

It is important that applications take into account the requirements of data as it is converted between the local code page, UCS-2 and UTF-8. For example, 20 characters will require exactly 40 bytes in UCS-2 and somewhere between 20 and 60 bytes in UTF-8, depending on the original code page and the characters used.

DB2 Unicode Databases and Applications

A DB2 Universal database for Unix, Windows, or OS/2 created specifying a code set of UTF-8 can be used to store data in both UCS-2 and UTF-8 formats. Such a database is referred to as a Unicode database. SQL character data is encoded using UTF-8 and SQL graphic data is encoded using UCS-2. This means that MBCS characters, including both single-byte and double-byte characters, are stored in character columns, and DBCS characters are stored in graphic columns.

The code page of an application may not match the code page that DB2 uses to store data. In a non-Unicode database, when the code pages are not the

381

same, the database manager converts character and graphic (pure DBCS) data that is transferred between client and server. In a Unicode database, the conversion of character data between the client code page and UTF-8 is automatically performed by the database manager, but all graphic (UCS-2) data is passed without any conversion between the client and the server.

Code Page Conversions Performed by the Database Manager
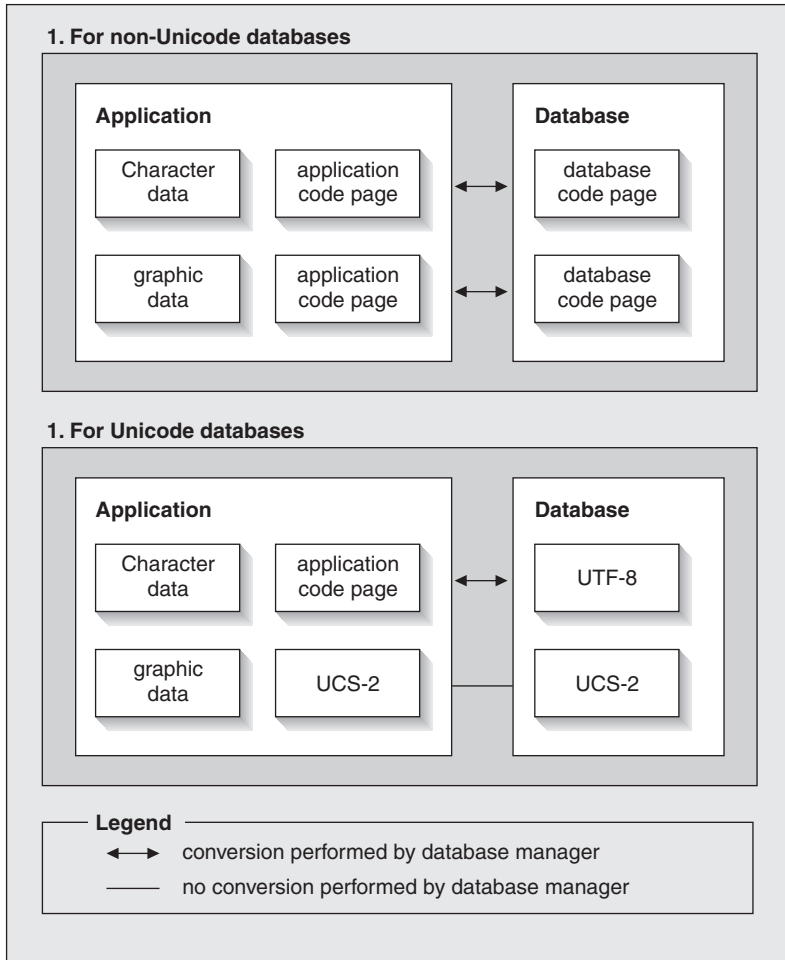
Figure 1. Code Page Conversions Performed by the Database Manager

Notes:

1. When connecting to Unicode Databases, if the application sets `DB2CODEPAGE=1208`, the local code page is UTF-8, so no code page conversion is needed.

2. When connected to a Unicode Database, CLI applications can also receive character data as graphic data, and graphic data as character data.

It is possible for an application to specify a UTF-8 code page, indicating that it will send and receive all graphic data in UCS-2 and character data in UTF-8. This application code page is only supported for Unicode databases.

Other points to consider when using Unicode:

1. The database code page is determined at the time the database is created, and by default its value is determined from the operating system locale (or code page). The CODESET and TERRITORY keywords can be used to explicitly create a Unicode DB2 database. For example:
   ```
   CREATE DATABASE unidb USING CODESET UTF-8 TERRITORY US
   ```
2. The application code page also defaults to the local code page, but this can be overridden by UTF-8 in one of two ways:
   - Setting the application code page to UTF-8 (1208) with this command:
     ```
     db2set DB2CODEPAGE=1208
     ```
   - For CLI/ODBC applications, by calling SQLSetConnectAttr() and setting the SQL_ATTR_ANSI_APP to SQL_AA_FALSE. The default setting is SQL_AA_TRUE.
3. Data in GRAPHIC columns will take exactly two bytes for each Unicode character, whereas data in CHAR columns will take from 1 to 3 bytes for each Unicode character. SQL limits in terms of characters for GRAPHIC columns are generally half of those as for CHAR columns, but they are equal in terms of bytes. For example, the maximum character length for a CHAR column is 254, and the maximum character length for a graphic column is 127. For more information, see MAX in the "Functions" chapter of the *SQL Reference*.
4. A graphic literal is differentiated from a character literal by a G prefix. For example:
   ```
   SELECT * FROM mytable WHERE mychar = 'utf-8 data'
                                   AND mygraphic = G'ucs-2 data'
   ```

   Note: The G prefix is optional for Unicode databases.
   See "Literals in Unicode Databases" on page 323 for more information and updated support.
5. Support for CLI/ODBC and JDBC applications differ from the support for Embedded applications. For information specific to CLI/ODBC support, see "CLI Guide and Reference" on page 317.
6. The byte ordering of UCS-2 data may differ between platforms. Internally, DB2 uses big-endian format.

Documentation Updates

These release notes include updates to the following information on using Unicode with DB2 Version 7.1:

- *SQL Reference*:
    - Chapter 3. Language Elements
    - Chapter 4. Functions
    - Chapter 6. SQL Statements
- *CLI Guide and Reference*:
    - Chapter 3. Using Advanced Features
    - Appendix C. DB2 CLI and ODBC
- *Data Movement Utilities Guide and Reference*, Appendix C. Export/Import/Load Utility File Formats

For more information on using Unicode with DB2 refer to the *Administration Guide*, National Language Support (NLS) appendix: "Unicode Support in DB2 UDB".

Part 7. Connecting to Host Systems

DB2 Connect User's Guide

Increasing DB2 Connect data transfer rate

While the blocking of rows for a query result set is nothing new, DB2 for z/OS (formerly called DB2 for OS/390) since its Version 6.1 release has had the capability of returning multiple query blocks in response to an OPEN or FETCH request to a remote client, such as DB2 Connect. Rather than repeatedly sending requests to the DB2 for z/OS server requesting one block of row data at a time, the client can now optionally request that the server send back an additional number of query blocks. Such additional query blocks are called extra query blocks.

This new feature allows the client to minimize the number of network line turnarounds, which has a major impact on network performance. The decrease in the number of requests sent by the client to the server for query blocks translates into a significant performance boost because switching between a send and receive is an expensive operation in terms of performance. DB2 Connect can now exploit this performance enhancement by requesting extra query blocks by default from a DB2 for z/OS server.

To take full advantage of the return of extra query blocks (each can be up to 32K bytes long) for the preferred network protocol of TCP/IP, Window Scaling extensions are also enabled as architected under RFC-1323 in DB2 Connect. This feature allows TCP/IP to dynamically and efficiently adjust the send and receive window sizes to accommodate the potentially large amounts of data returned by way of the extra query blocks.

Extra Query Blocks

Extra query block support in DB2 for z/OS servers on Versions 6.1 or later is configured via the EXTRA BLOCKS SRV parameter on the DB2 DDF installation panel. This parameter controls the maximum number of extra query blocks that DB2 can send back to a client for a request and can be set to a value between 0 and 100. Setting the parameter value to 0 disables the return of extra query blocks. The default value of 100 should be used to get the most benefit out of this feature, barring any idiosyncrasies in the network that would render this setting less than ideal.

On the client side where the application accesses DB2 for z/OS either directly through a co-located DB2 Connect installation, or through a separate DB2 Connect server installation, there are various means for activating the corresponding DB2 Connect support on a per cursor or statement basis through the use of:

387

- A query rowset size for a cursor
- The 'OPTIMIZE for N ROWS' clause on the select statement associated with a cursor
- The 'FETCH FIRST N ROWS ONLY' clause on the select statement associated with a cursor.

Option 1 is not covered under in this section because it was already implemented as part of DB2 for z/OS Scrollable Support in DB2 Connect Version 7.1 FixPak 2. Our focus is on the use of options 2 and 3 instead to enabling extra query block support using different SQL APIs as follows:

1. Embedded SQL
   - Invoke extra query block support for a query by specifying the 'OPTIMIZE for N ROWS' clause and/or the 'FETCH FIRST N ROWS ONLY' clause on the select statement itself.
   - With the 'OPTIMIZE for N ROWS' clause, DB2 for z/OS will attempt to block the desired number of rows to return to DB2 Connect, subject to the EXTRA BLOCKS SRV DDF installation parameter setting. The application can choose to fetch beyond N rows as DB2 for z/OS does not limit the total number of rows that could ultimately be returned for the query result set to N.
   - The 'FETCH FIRST N ROWS ONLY' clause works similarly, except that the query result set is limited to N rows by DB2 for z/OS. Fetching beyond N rows would result in SQL code +100 (end of data).

2. CLI/ODBC
   - Invoke extra query block support for a query through its SQL_MAX_ROWS statement attribute.
   - DB2 Connect will tag on the 'OPTIMIZE for N ROWS' clause for a DB2 for z/OS 6.x server. Even though the number of rows that could ultimately be returned for the query result set is not limited to N by DB2 for z/OS, CLI/ODBC would return SQL_NO_DATA_FOUND to the application if an attempt is made to fetch beyond N rows.
   - The 'FETCH FIRST N ROWS ONLY' clause is used instead for a DB2 for z/OS 7.1 or above server. Similar to the embedded SQL case, the query result set is limited to N rows by DB2 for z/OS. Fetching beyond N rows would result in SQL_NO_DATA_FOUND.

3. JDBC
   - Invoke extra query block support for a query through the setMaxRows method. Similar to the CLI/ODBC enablement, DB2 Connect will tag on the 'OPTIMIZE for N ROWS' clause for a DB2 for z/OS server Version 6.x , and the 'FETCH FIRST N ROWS ONLY' clause for a DB2 for z/OS server Version 7.1 or later.

RFC-1323 Window Scaling

Window Scaling is supported as of FixPak 4 on all Windows and UNIX platforms that support the RFC-1323 extensions for TCP/IP. This feature can be enabled on DB2 for Windows and UNIX via the DB2 registry variable DB2SORCVBUF. To enable Window Scaling, set the DB2 registry variable DB2SORCVBUF to any value above 64K (for example, on DB2 for Windows or UNIX, you can issue db2set DB2SORCVBUF =65537). The maximum send and receive buffer sizes are dependent on the specific operating system. To ensure that buffer sizes configured have been accepted, the user can set the database manager configuration parameter DIAGLEVEL to 4 (informational) and check the db2diag.log file for messages.

For Window Scaling to take effect, it must be enabled on both ends of a connection. For example, to enable Window Scaling between the DB2 Connect workstation and the host, this feature must be active on both the workstation and the host, either directly through the operating system TCP/IP stack, or indirectly through the DB2 product. For instance, for DB2 for z/OS, Window Scaling can currently only be activated through the operating system by setting TCPRCVBUFRSIZE to any value above 64K.

If a remote DB2 client is used for accessing host DB2 through a DB2 Connect server workstation, Window Scaling can be enabled on the client also. By the same token, Window Scaling can also be enabled between a remote DB2 client and a workstation DB2 server when no host DB2 is involved.

While Window Scaling is designed to enhance network performance, the expected network performance improvement does not always materialize. Interaction among factors such as the frame size used for the Ethernet or token ring LAN adapter, the IP MTU size, and other settings at routers throughout the communication link could even result in performance degradation once Window Scaling has been enabled. By default, Window Scaling is disabled with both the send and receive buffers set to 64K. The user should be prepared to assess the impact of turning on Window Scaling and perform any necessary adjustments to the network. For an introduction to tuning the network for improved network performance, refer to the white paper at http://www.networking.ibm.com/per/per10.html.

DB2 Connect Support for Loosely Coupled Transactions

The support within DB2 Connect for loosely coupled transactions is intended to be used by user who implement XA distributed applications that access DB2 for OS/390 Version 6 or later. This support allows different branches of the same global transaction to share lock space on DB2 for OS/390. This feature reduces the window where one branch of a distributed transaction encounters lock timeout or deadlock as a result of another branch within the same global transaction. DB2 for OS/390 Version 6 shares the lock space in

this situation provided DB2 Connect sends the XID on each connection
serving different branches of the same global transaction.

Kerberos support

DB2 Universal Database currently supports the Kerberos security protocol as a
means to authenticate users in the non-DRDA environment. Since DB2/390
V7.1 will start to support Kerberos security, DB2 Connect will add DRDA AR
functionality to allow the use of Kerberos authentication to connect to
DB2/390.

The Kerberos authentication layer which handles the ticketing system is
integrated into the Win2K Active Directory mechanism. The client and server
sides of an application communicate with the Kerberos SSP (Security Support
Provider) client and server modules respectively. The Security Support
Provider Interface (SSPI) provides a high level interface to the Kerberos SSP
and other security protocols

Communication protocol support:

For SNA connection, you must use SECURITY=NONE when cataloging the
APPC node

Typical setup:

The procedure to configure DB2 to use Kerberos authentication involves
setting up the following:
- An authorization policy for DB2 (as a service) in the Active Directory that is
  shared on a network, and
- Trust relationship between Kerberos Key Distribution Centers (KDCs)

In the simplest scenario, there is at least one KDC trust relationship to
configure, that is, the one between the KDC controlling the client workstation,
and the OS/390 system. OS/390 R10 provides Kerberos ticket processing
through its RACF facility which allows the host to act as an UNIX KDC.

DB2 Connect provides as usual the router functionality in the 3-tier setting. It
does not assume any role in authentication when Kerberos security is used.
Instead, it merely passes the client's security token to DB2/390. Thus there is
no need for the DB2 Connect gateway to be a member of the client or the
host's Kerberos realm.

To use Kerberos, both the DB2 Connect gateway must catalog its connection
with authentication type KERBEROS. The client can either catalog with

authentication NOT_SPEC or Kerberos. Any other combinations of authentication types on the client and the gateway results in sqlcode -1401 (Authentication type mismatch).

Downlevel compatibility:

DB2 requirements for Kerberos support:

DB2 UDB Client:
Version 7.1 (OS: Win2K)

DB2 Connect:
Version 7.1 + Fix Pack 1 (OS: Any)

DB2/390:
Version 7.1

DB2/390 also have a requirement to be run on OS/390 Version 2 Release 10 or later. There are additional implied requirements on downlevel DB2/390 systems when connecting from DB2 Connect Version 7.1 clients. Although these DB2/390 systems do not support Kerberos, they do not respond properly to unsupported DRDA SECMECs. To solve this problem, apply the proper PTF:
- UQ41941 (for DB2/390 V5.1)
- UQ41942 (for DB2/390 V6.1)

Connectivity Supplement

Setting Up the Application Server in a VM Environment

Add the following sentence after the first (and only) sentence in the section "Provide Network Information", subsection "Defining the Application Server":

```
The RDB_NAME is provided on the SQLSTART EXEC as the DBNAME parameter.
```

CLI/ODBC/JDBC Configuration PATCH1 and PATCH2 Settings

The CLI/ODBC/JDBC driver can be configured through the Client Configuration Assistant or the ODBC Driver Manager (if it is installed on the system), or by manually editing the db2cli.ini file. For more details, see either the *Installation and Configuration Supplement*, or the *CLI Guide and Reference*.

The DB2 CLI/ODBC driver default behavior can be modified by specifying values for both the PATCH1 and PATCH2 keyword through either the db2cli.ini file or through the SQLDriverConnect() or SQLBrowseConnect() CLI API.

The PATCH1 keyword is specified by adding together all keywords that the user wants to set. For example, if patch 1, 2, and 8 were specified, then PATCH1 would have a value of 11. Following is a description of each keyword value and its effect on the driver:

1   This makes the driver search for "count(exp)" and replace it with "count(distinct exp)". This is needed because some versions of DB2 support the "count(exp)" syntax, and that syntax is generated by some ODBC applications. Needed by Microsoft applications when the server does not support the "count(exp)" syntax.

2   Some ODBC applications are trapped when SQL_NULL_DATA is returned in the SQLGetTypeInfo() function for either the LITERAL_PREFIX or LITERAL_SUFFIX column. This forces the driver to return an empty string instead. Needed by Impromptu 2.0.

4   This forces the driver to treat the input time stamp data as date data if the time and the fraction part of the time stamp are zero. Needed by Microsoft Access.

8   This forces the driver to treat the input time stamp data as time data if the date part of the time stamp is 1899-12-30. Needed by Microsoft Access.

16   Not used.

32	This forces the driver to not return information about SQL_LONGVARCHAR, SQL_LONGVARBINARY, and SQL_LONGVARGRAPHIC columns. To the application it appears as though long fields are not supported. Needed by Lotus 123.
64	This forces the driver to NULL terminate graphic output strings. Needed by Microsoft Access in a double byte environment.
128	This forces the driver to let the query "SELECT Config, nValue FROM MSysConf" go to the server. Currently the driver returns an error with associated SQLSTATE value of S0002 (table not found). Needed if the user has created this configuration table in the database and wants the application to access it.
256	This forces the driver to return the primary key columns first in the SQLStatistics() call. Currently, the driver returns the indexes sorted by index name, which is standard ODBC behavior.
512	This forces the driver to return FALSE in SQLGetFunctions() for both SQL_API_SQLTABLEPRIVILEGES and SQL_API_SQLCOLUMNPRIVILEGES.
1024	This forces the driver to return SQL_SUCCESS instead of SQL_NO_DATA_FOUND in SQLExecute() or SQLExecDirect() if the executed UPDATE or DELETE statement affects no rows. Needed by Visual Basic applications.
2048	Not used.
4096	This forces the driver to not issue a COMMIT after closing a cursor when in autocommit mode.
8192	This forces the driver to return an extra result set after invoking a stored procedure. This result set is a one row result set consisting of the output values of the stored procedure. Can be accessed by Powerbuild applications.
32768	This forces the driver to make Microsoft Query applications work with DB2 MVS synonyms.
65536	This forces the driver to manually insert a "G" in front of character literals which are in fact graphic literals. This patch should always be supplied when working in an double byte environment.
131072	This forces the driver to describe a time stamp column as a CHAR(26) column instead, when it is part of an unique index. Needed by Microsoft applications.
262144	This forces the driver to use the pseudo-catalog table db2cli.procedures instead of the SYSCAT.PROCEDURES and SYSCAT.PROCPARMS tables.

524288 his forces the driver to use SYSTEM_TABLE_SCHEMA instead of TABLE_SCHEMA when doing a system table query to a DB2/400 V3.x system. This results in better performance.

1048576

This forces the driver to treat a zero length string through SQLPutData() as SQL_NULL_DATA.

The PATCH2 keyword differs from the PATCH1 keyword. In this case, multiple patches are specified using comma separators. For example, if patch 1, 4, and 5 were specified, then PATCH2 would have a value of "1,4,5". Following is a description of each keyword value and its effect on the driver:

```
1 - This forces the driver to convert the name of the stored procedure
     in a CALL statement to uppercase.

2 - Not used.

3 - This forces the driver to convert all arguments to schema calls to
     uppercase.

4 - This forces the driver to return the Version 2.1.2 like result set
     for schema calls (that is, SQLColumns(), SQLProcedureColumns(), and
     so on), instead of the Version 5 like result set.

5 - This forces the driver to not optimize the processing of input VARCHAR
     columns, where the pointer to the data and the pointer to the length
     are consecutive in memory.

6 - This forces the driver to return a message that scrollable cursors
     are not supported. This is needed by Visual Basic programs if the
     DB2 client is Version 5 and the server is DB2 UDB Version 5.

7 - This forces the driver to map all GRAPHIC column data types to the
     CHAR column data type. This is needed in a double byte environment.

8 - This forces the driver to ignore catalog search arguments in schema
     calls.
9 - Do not commit on Early Close of a cursor
10 - Not Used
11 - Report that catalog name is supported, (VB stored procedures)
12 - Remove double quotes from schema call arguments, (Visual Interdev)
13 - Do not append keywords from db2cli.ini to output connection string
14 - Ignore schema name on SQLProcedures() and SQLProcedureColumns()
15 - Always use period for decimal separator in character output
16 - Force return of describe information for each open
17 - Do not return column names on describe
18 - Attempt to replace literals with parameter markers
19 - Currently, DB2 MVS V4.1 does not support the ODBC syntax where
      parenthesis are allowed in the ON clause in an Outer join clause.
      Turning on this PATCH2 will cause IBM DB2 ODBC driver to strip
      the parenthesis when the outer join clause is in an ODBC escape
      sequence. This PATCH2 should only be used when going against
      DB2 MVS 4.1.
```

```
20 - Currently, DB2 on MVS does not support BETWEEN predicate with
     parameter markers as both operands (expression ? BETWEEN ?).
     Turning on this patch will cause the IBM ODBC Driver to rewrite
     the predicate to (expression >= ? and expression <= ?).
21 - Set all OUTPUT only parameters for stored procedures to
     SQL_NULL_DATA
22 - This PATCH2 causes the IBM ODBC driver to report OUTER join as
     not supported. This is for application that generates SELECT DISTINCT
     col1 or ORDER BY col1 when using outer join statement where col1 has
     length greater than 254 characters and causes DB2 UDB to return
     an error (since DB2 UDB does not support greater-than-254 byte
     column in this usage
23 - Do not optimize input for parameters bound with cbColDef=0
24 - Access workaround for mapping Time values as Characters
25 - Access workaround for decimal columns - removes trailing zeros in
     char representation
26 - Do not return sqlcode 464 to application - indicates result sets
     are returned
27 - Force SQLTables to use TABLETYPE keyword value, even if the application
     specifies a valid value
28 - Describe real columns as double columns
29 - ADO workaround for decimal columns - removes leading zeroes
     for values x, where 1 > x > -1 (Only needed for
     some MDAC versions)
30 - Disable the Stored Procedure caching optimization
31 - Report statistics for aliases on SQLStatistics call
32 - Override the sqlcode -727 reason code 4 processing
33 - Return the ISO version of the time stamp when converted to char
     (as opposed to the ODBC version)
34 - Report CHAR FOR BIT DATA columns as CHAR
35 - Report an invalid TABLENAME when SQL_DESC_BASE_TABLE_NAME
     is requested - ADO readonly optimization
36 - Reserved
37 - Reserved
```

Part 8. Additional Information

Additional Information

DB2 Everywhere is Now DB2 Everyplace

The name of DB2 Everywhere has changed to DB2 Everyplace.

Accessibility Features of DB2 UDB Version 7

The DB2 UDB family of products includes a number of features that make the products more accessible for people with disabilities. These features include:

- Features that facilitate keyboard input and navigation
- Features that enhance display properties
- Options for audio and visual alert cues
- Compatibility with assistive technologies
- Compatibility with accessibility features of the operating system
- Accessible documentation formats

Keyboard Input and Navigation

Keyboard Input
The DB2 Control Center can be operated using only the keyboard. Menu items and controls provide access keys that allow users to activate a control or select a menu item directly from the keyboard. These keys are self-documenting, in that the access keys are underlined on the control or menu where they appear.

Keyboard Focus
In UNIX-based systems, the position of the keyboard focus is highlighted, indicating which area of the window is active and where the user's keystrokes will have an effect.

Features for Accessible Display

The DB2 Control Center has a number of features that enhance the user interface and improve accessibility for users with low vision. These accessibility enhancements include support for high-contrast settings and customizable font properties.

High-Contrast Mode
The Control Center interface supports the high-contrast-mode option provided by the operating system. This feature assists users who require a higher degree of contrast between background and foreground colors.

Font Settings

The Control Center interface allows users to select the color, size, and font for the text in menus and dialog windows.

Non-dependence on Color

Users do not need to distinguish between colors in order to use any of the functions in this product.

Alternative Alert Cues

The user can opt to receive alerts through audio or visual cues.

Compatibility with Assistive Technologies

The DB2 Control Center interface is compatible with screen reader applications such as Via Voice. When in application mode, the Control Center interface has the properties required for these accessibility applications to make onscreen information available to blind users.

Accessible Documentation

Documentation for the DB2 family of products is available in HTML format. This allows users to view documentation according to the display preferences set in their browsers. It also allows the use of screen readers and other assistive technologies.

Mouse Required

For all platforms except Windows, a mouse is required to use the tools.

Attempting to Bind from the DB2 Run-time Client Results in a ″Bind files not found″ Error

Because the DB2 Run-time Client does not have the full set of bind files, the binding of GUI tools cannot be done from the DB2 Run-time Client, and can only be done from the DB2 Administration Client.

Search Discovery

Search discovery is only supported on broadcast media. For example, search discovery will not function through an ATM adapter. However, this restriction does not apply to known discovery.

Memory Windows for HP-UX 11

Memory windows is for users on large HP 64-bit machines, who want to take advantage of greater than 1.75 GB of shared memory for 32-bit applications. Memory windows is not required if you are running the 64-bit version of DB2. Memory windows makes available a separate 1 GB of shared memory per process or group of processes. This allows an instance to have its own 1

GB of shared memory, plus the 0.75 GB of global shared memory. If users want to take advantage of this, they can run multiple instances, each in its own window. Following are prerequisites and conditions for using memory windows:

- DB2 EE environment
  - Patches: Extension Software 12/98, and PHKL_17795.
  - The $DB2INSTANCE variable must be set for the instance.
  - There must be an entry in the /etc/services.window file for each DB2 instance that you want to run under memory windows. For example:

    ```
    db2instance1 50
    db2instance2 60
    ```

    Note: There can only be a single space between the name and the ID.
  - Any DB2 commands that you want to run on the server, and that require more than a single statement, must be run using a TCP/IP loopback method. This is because the shell will terminate when memory windows finishes processing the first statement. DB2 Service knows how to accomplish this.
  - Any DB2 command that you want to run against an instance that is running in memory windows must be prefaced with db2win (located in sqllib/bin). For example:

    ```
    db2win db2start
    db2win db2stop
    ```
  - Any DB2 command that is run outside of memory windows (but when memory windows is running) should return a 1042. For example:

    ```
    db2win db2start <== OK
    db2 connect to db  <==SQL1042
    db2stop <==SQL1042
    db2win db2stop   <== OK
    ```

- DB2 EEE environment
  - Patches: Extension Software 12/98, and PHKL_17795.
  - The $DB2INSTANCE variable must be set for the instance.
  - The DB2_ENABLE_MEM_WINDOWS registry variable must be set to TRUE.
  - There must be an entry in the /etc/services.window file for each logical node of each instance that you want to run under memory windows. The first field of each entry should be the instance name concatenated with the port number. For example:

    ```
    === $HOME/sqllib/db2nodes.cfg for db2instance1 ===
    5 host1 0
    7 host1 1
    9 host2 0

    === $HOME/sqllib/db2nodes.cfg for db2instance2 ===
    ```

```
1 host1 0
2 host2 0
3 host2 1

=== /etc/services.window on host1 ===
db2instance10 50
db2instance11 55
db2instance20 60

=== /etc/services.window on host2 ===
db2instance10 30
db2instance20 32
db2instance21 34
```
  – You must not preface any DB2 command with db2win, which is to be
    used in an EE environment only.

Uninstalling DB2 DFS Client Enabler

Before the DB2 DFS Client Enabler is uninstalled, root should ensure that no
DFS file is in use, and that no user has a shell open in DFS file space. As root,
issue the command:

```
stop.dfs dfs_cl
```

Check that /... is no longer mounted:

```
mount | grep -i dfs
```

If this is not done, and DB2 DFS Client Enabler is uninstalled, the machine
will need to be rebooted.

Client Authentication on Windows NT

A new DB2 registry variable DB2DOMAINLIST is introduced to complement
the existing client authentication mechanism in the Windows NT environment.
This variable is used on the DB2 for Windows NT server to define one or
more Windows NT domains. Only connection or attachment requests from
users belonging to the domains defined in this list will be accepted.

This registry variable should only be used under a pure Windows NT domain
environment with DB2 servers and clients running at Version 7 (or higher).

For information about setting this registry variable, refer to the "DB2 Registry
and Environment Variables" section in the *Administration Guide: Performance*.

Federated Systems Restrictions

Following are restrictions that apply to federated systems:

- The Oracle data types NCHAR, NVARCHAR2, NCLOB, and BFILE are not supported in queries involving nicknames.
- The Create Server Option, Alter Server Option, and Drop Server Option commands are not supported from the Control Center. To issue any of these commands, you must use the command line processor (CLP).
- For queries involving nicknames, DB2 UDB does not always abide by the DFT_SQLMATHWARN database configuration option. Instead, DB2 UDB returns the arithmetic errors or warnings directly from the remote data source regardless of the DFT_SQLMATHWARN setting.
- The CREATE SERVER statement does not allow the COLSEQ server option to be set to 'I' for data sources with case-insensitive collating sequences.
- The ALTER NICKNAME statement returns SQL0901N when an invalid option is specified.
- For Oracle, Microsoft SQL Server, and Sybase data sources, numeric data types cannot be mapped to DB2's BIGINT data type. By default, Oracle's number(p,s) data type, where 10 <= p <= 18, and s = 0, maps to DB2's DECIMAL data type.

Federated Limitations with MPP Partitioned Tables

When you attempt to use one SQL statement to select data from a data source and insert, update, or delete the data directly in an MPP partitioned table on your DB2 federated server, you will receive the SQL0901N error. The federated functionality does not allow you to select from a nickname and insert into an MPP partitioned table.

Once you apply FixPak 4 (or above), you can use these steps to select data and insert the data into an MPP partitioned table:

1. In the customer application environment, export the DB2NODE environment variable to designate the node to which the application should always connect.

       EXPORT DB2NODE=x

   where *x* is a node number.
2. Create a nodegroup which contains only the designated node.

       CREATE NODEGROUP nodegroup_name ON NODE(x)

   where *x* is the node number.
3. Create a tablespace in the nodegroup.

   CREATE TABLESPACE tablespace_name IN NODEGROUP nodegroup_name
4. Create a temporary table in the tablespace.

   CREATE TABLE temp_table_name IN tablespace_name
5. Divide the INSERT operation in the application into two steps:

- INSERT INTO temp_table_name SELECT * FROM nickname
- INSERT INTO MPP_partitioned_table SELECT * from temp_table_name

Dividing the INSERT statement into two statements changes the statement level commit and rollback semantics. For example, instead of rolling back one statement, you will now have to rollback two statements. Additionally, if you change the node number associated with the DB2NODE environment variable, you must invalidate the application package and rebind.

These steps allow you to select data from data sources and insert the data into an MPP partitioned table. You will still receive the SQL0901N error when you attempt to use one statement to select data from a data source and update or delete the data in an MPP partitioned table. This restriction will be elimiated in DB2 Universal Database Version 8.

DataJoiner Restriction

Distributed requests issued within a federated environment are limited to read-only operations.

Hebrew Information Catalog Manager for Windows NT

The Information Catalog Manager component is available in Hebrew and is provided on the DB2 Warehouse Manager for Windows NT CD.

The Hebrew translation is provided in a zip file called IL_ICM.ZIP and is located in the DB2\IL directory on the DB2 Warehouse Manager for Windows NT CD.

To install the Hebrew translation of Information Catalog Manager, first install the English version of DB2 Warehouse Manager for Windows NT and all prerequisites on a Hebrew Enabled version of Windows NT.

After DB2 Warehouse Manager for Windows NT has been installed, unzip the IL_ICM.ZIP file from the DB2\IL directory into the same directory where DB2 Warehouse Manager for Windows NT was installed. Ensure that the correct options are supplied to the unzip program to create the directory structure in the zip file.

After the file has been unzipped, the global environment variable LC_ALL must be changed from En_US to Iw_IL. To change the setting:

1. Open the Windows NT Control Panel and double click on the **System** icon.
2. In the **System Properties** window, click on the **Environment** tab, then locate the variable **LC_ALL** in the **System Variables** section.

3. Click on the variable to display the value in the **Value** edit box. Change the value from **En_US** to **Iw_IL**.

4. Click on the **Set** button.

5. Close the **System Properties** window and the **Control Panel**.

The Hebrew version of Information Catalog Manager should now be installed.

DB2's SNA SPM Fails to Start After Booting Windows

If you are using Microsoft SNA Server Version 4 SP3 or later, please verify that DB2's SNA SPM started properly after a reboot. Check the \sqllib\<instance name>\db2diag.log file for entries that are similar to the following:

```
2000-04-20-13.18.19.958000   Instance:DB2   Node:000
PID:291(db2syscs.exe)   TID:316   Appid:none
common_communication  sqlccspmconnmgr_APPC_init    Probe:19
SPM0453C  Sync point manager did not start because Microsoft SNA Server has not
been started.

2000-04-20-13.18.23.033000   Instance:DB2   Node:000
PID:291(db2syscs.exe)   TID:302   Appid:none
common_communication  sqlccsna_start_listen    Probe:14
DIA3001E "SNA SPM" protocol support was not successfully started.

2000-04-20-13.18.23.603000   Instance:DB2   Node:000
PID:291(db2syscs.exe)   TID:316   Appid:none
common_communication  sqlccspmconnmgr_listener    Probe:6
DIA3103E Error encountered in APPC protocol support. APPC verb "APPC(DISPLAY 1
BYTE)". Primary rc was "F004". Secondary rc was "00000000".
```

If such entries exist in your db2diag.log, and the time stamps match your most recent reboot time, you must:

1. Invoke db2stop.

2. Start the SnaServer service (if not already started).

3. Invoke db2start.

Check the db2diag.log file again to verify that the entries are no longer appended.

Service Account Requirements for DB2 on Windows NT and Windows 2000

During the installation of DB2 for Windows NT or Windows 2000, the setup program creates several Windows services and assigns a service account for each service. To run DB2 properly, the setup program grants the following user rights to the service account that is associated with the DB2 service:

• Act as part of the operating system

• Create a token object

- Increase quotas
- Log on as a service
- Replace a process level token.

If you want to use a different service account for the DB2 services, you must grant these user rights to the service account.

In addition to these user rights, the service account must also have write access to the directory where the DB2 product is installed.

The service account for the DB2 Administration Server service (DB2DAS00 service) must also have the authority to start and stop other DB2 services (that is, the service account must belong to the Power Users group) and have DB2 SYSADM authority against any DB2 instances that it administers.

Need to Commit all User-defined Programs That Will Be Used in the Data Warehouse Center (DWC)

If you want to use a stored procedure built by the DB2 Stored Procedure Builder as a user-defined program in the Data Warehouse Center (DWC), you must insert the following statement into the stored procedure before the `con.close();` statement:

```
con.commit();
```

If this statement is not inserted, changes made by the stored procedure will be rolled back when the stored procedure is run from the DWC.

For all user-defined programs in the DWC, it is necessary to *explicitly* commit any included DB2 functions for the changes to take effect in the database; that is, you must add the COMMIT statements to the user-defined programs.

Client-side Caching on Windows NT

If a user tries to access a READ PERM DB file residing on a Windows NT server machine where DB2 Datalinks is installed using a shared drive using a valid token, the file opens as expected. However, after that, subsequent open requests using the same token do not actually reach the server, but are serviced from the cache on the client. Even after the token expires, the contents of the file continue to be visible to the user, since the entry is still in the cache. However, this problem does not occur if the file resides on a Windows NT workstation.

A solution would be to set the registry entry \\HKEY_LOCAL_MACHINE\SYSTEM \CurrentControlSet\Services\Lanmanserver\Parameters\EnableOpLocks to zero on the Windows NT server. With this registry setting, whenever a file

residing on the server is accessed from a client workstation through a shared drive, the request will always reach the server, instead of being serviced from the client cache. Therefore, the token is re-validated for all requests.

The negative impact of this solution is that this affects the overall performance for all file access from the server over shared drives. Even with this setting, if the file is accessed through a shared drive mapping on the server itself, as opposed to from a different client machine, it appears that the request is still serviced from the cache. Therefore, the token expiry does not take effect.

Note: In all cases, if the file access is a local access and not through a shared drive, token validation and subsequent token expiry will occur as expected.

Life Sciences Data Connect

New Wrappers

In FixPak 4, two new wrappers were added to Life Sciences Data Connect. One for Documentum on AIX, and one for Excel on Windows NT. Additionally, the table-structured file wrapper was ported from AIX to Windows NT, Solaris, Linux, and HP-UX systems.

For FixPak 5, The BLAST wrapper on AIX has been added to DB2 Life Sciences Data Connect. The Documentum wrapper has been ported from AIX to Windows NT, Windows 2000, and Solaris Operating Environment.

For FixPak 6, The BLAST wrapper has been ported from AIX to Windows NT, Windows 2000, HP-UX, and Solaris Operating Environment.

Notices–

Life Sciences Data Connect includes code from The Apache Software and ICU. The code is provided on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. Further, no IBM obligation of indemnification applies.

The Apache Software License, Version 1.1

Copyright (c) 1999-2001 The Apache Software Foundation. All rights reserved.

ICU 1.8.1 and later

Copyright (c) 1995-2001 International Business Machines Corporation and others All rights reserved.

Enhancement to SQL Assist

The SQL Assist tool now allows the user to specify a join operator other than
″=″ for table joins. The Join Type dialog, which is launched by clicking the
Join Type button on the Joins page of the SQL Assist tool, has been enhanced
to include a drop-down list of join operators.

The available operators are ″=″, ″<>″, ″<″, ″>″, ″<=″, and ″>=″. SQL Assist is a
tool that assists the user in creating simple SQL statements. It is available
from the Command Center (Interactive tab), the Control Center (Create View
and Create Trigger dialogs), the Stored Procedure Builder (″Inserting SQL
Stored Procedure″ wizard), and the Data Warehouse Center (SQL Process
step).

Help for Backup and Restore Commands

Incorrect information appears when you type db2 ? backup. The correct
output is:

```
BACKUP DATABASE database-alias [USER username [USING password]]
[TABLESPACE (tblspace-name [ {,tblspace-name} ... ])] [ONLINE]
[INCREMENTAL [DELTA]] [USE TSM [OPEN num-sess SESSIONS]] |
TO dir/dev [ {,dir/dev} ... ] | LOAD lib-name [OPEN num-sess SESSIONS]]
[WITH num-buff BUFFERS] [BUFFER buffer-size] [PARALLELISM n]
[WITHOUT PROMPTING]
```

Incorrect information appears when you type db2 ? restore. The correct
output is:

```
RESTORE DATABASE source-database-alias { restore-options | CONTINUE | ABORT }";

restore-options:";
  [USER username [USING password]] [{TABLESPACE [ONLINE] |";
  TABLESPACE (tblspace-name [ {,tblspace-name} ... ]) [ONLINE] |";
  HISTORY FILE [ONLINE]}] [INCREMENTAL [ABORT]]";
  [{USE TSM [OPEN num-sess SESSIONS] |";
  FROM dir/dev [ {,dir/dev} ... ] | LOAD shared-lib";
  [OPEN num-sess SESSIONS]}] [TAKEN AT date-time] [TO target-directory]";
  [INTO target-database-alias] [NEWLOGPATH directory]";
  [WITH num-buff BUFFERS] [BUFFER buffer-size]";
  [DLREPORT file-name] [REPLACE EXISTING] [REDIRECT] [PARALLELISM n]";
  [WITHOUT ROLLING FORWARD] [WITHOUT DATALINK] [WITHOUT PROMPTING]";
```

″Warehouse Manager″ Should Be ″DB2 Warehouse Manager″

All occurrences of the phrase ″Warehouse Manager″ in product screens and in
product documentation should read ″DB2 Warehouse Manager″.

Appendix. Notices

IBM may not offer the products, services, or features discussed in this document in all countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

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

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

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make

improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

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

IBM Canada Limited
Office of the Lab Director
1150 Eglinton Ave. East
North York, Ontario
M3C 1H7
CANADA

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

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

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

COPYRIGHT LICENSE:

This information may contain sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Each copy or any portion of these sample programs or any derivative work must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

Trademarks

The following terms, which may be denoted by an asterisk(*), are trademarks of International Business Machines Corporation in the United States, other countries, or both.

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

The following terms are trademarks or registered trademarks of other companies:

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

Java or all Java-based trademarks and logos, and Solaris are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Tivoli and NetView are trademarks of Tivoli Systems Inc. in the United States, other countries, or both.

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

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

Index

A

accessibility
  keyboard shortcuts   218
accessibility features   399
accessing help   338
Adobe Acrobat Reader
  accessing with non-english locales   11
ADT transforms   318
AIX
  DB2 installation on   17
  installation errors   30
AIX 4.3.3
  AutoLoader hanging after a fork   164
ARCHIVE LOG
  usage notes   134
AS/400
  capture and apply for   170
assigning strings to columns, rules for   355
assignments
  datetime to string value   359
  datetime values, rules for   359
  strings, basic rules for   355
authentication
  by domain using DB2DOMAINLIST   402
AutoLoader
  hanging during fork   164

B

backup
  when LOGRETAIN=CAPTURE   158
backup method
  for Data Links Manager   159
bidirectional language support
  with Control Center on Windows NT   217
binding database utilities using Run-Time Client   317
buffer pool blocking   107
  examples   109
buffer pool IDs   109
buffer pools
  block-based   107
  contiguous pages   107
  page-based   107

C

CAPTURE (LOGRETAIN setting)
  backup and restore behavior   158
capture and apply
  error messages   174
  on AS/400   170

capture and apply *(continued)*
  starting from within an application   175
character conversion
  rules for operations combining strings   357
  rules when comparing strings   357
character string
  assignment, overview   355
character strings
  double-byte character string   364
  VARGRAPHIC scalar function   364
CLI
  advantages of   319
  configuration keywords
    CURRENTFUNCTIONPATH   329
    SKIPTRACE   329
  stored procedures
    multiple result sets   319
  writing a Unicode application   319
CLI functions
  SQLBindFileToParam   330
  SQLColAttribute   330
  SQLGetInfo   331
  SQLGetLength   331
  SQLNextResult   331
  SQLSetStmtAttr   334
CLI stored procedures
  autobinding   328
  limitations   328
CLI0637E   316
clients
  downloading   17
  three-tier environment   325
CLOB data
  MQSeries functions   353
clone databases
  creating   155
column
  string assignment, basic rules for   355
commands
  CREATE DATABASE   152
compound SQL, using   327
concatenation
  operator   358
connections
  increasing data transfer rates   387
Control Center
  bidirectional support
    on Windows NT   217
  errors with Internet Explorer   5

IBM ®

Printed in U.S.A.