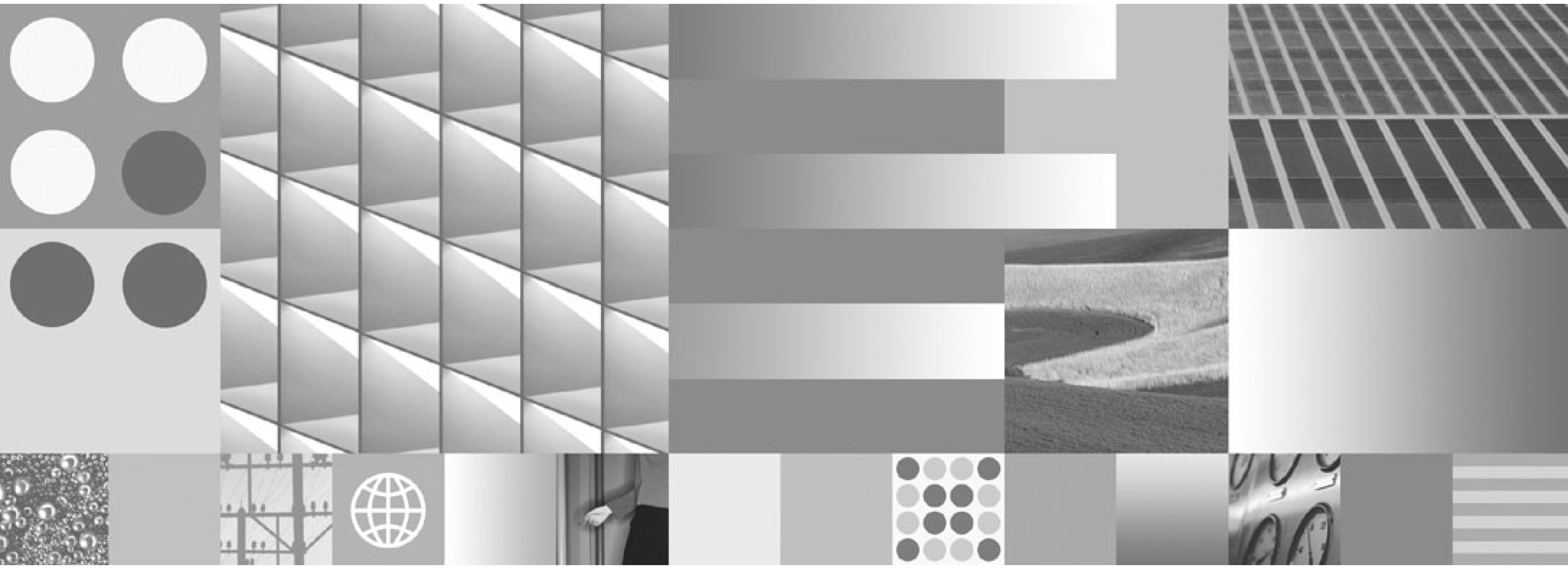
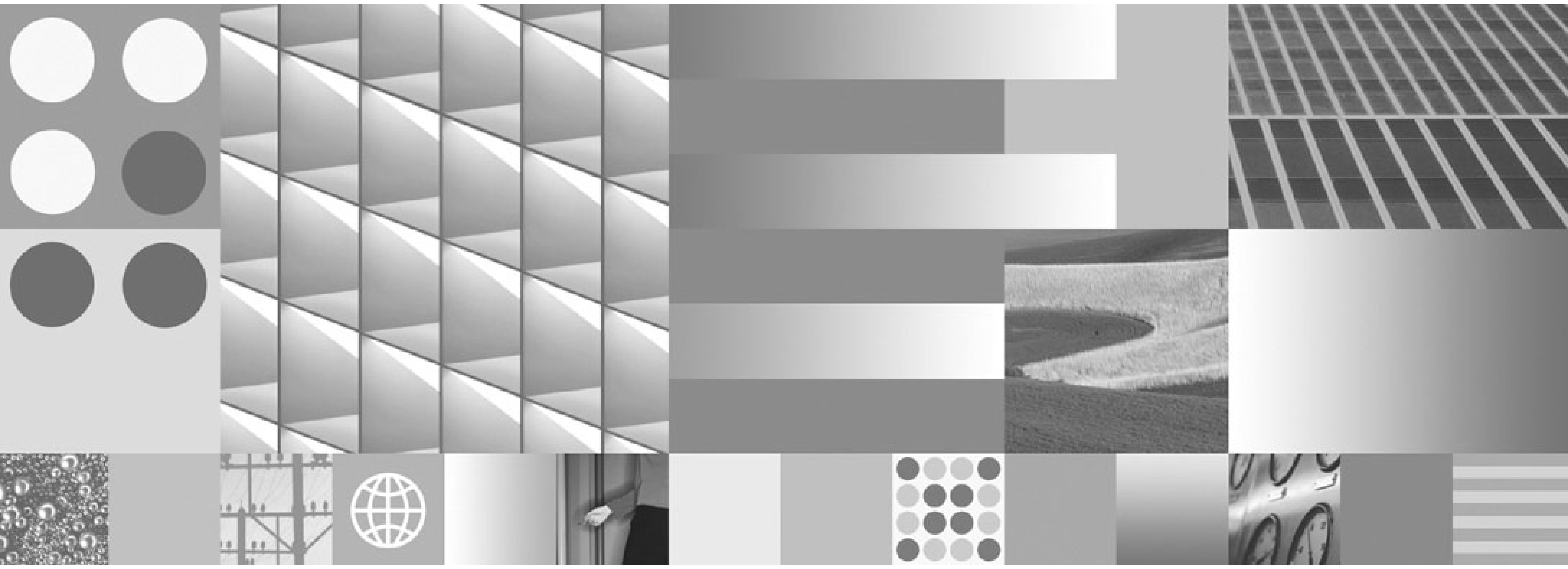
DB2 Version 9.5
for Linux, UNIX, and Windows



Getting Started with Database Application Development
Updated April, 2009

GC23-5856-02

DB2 Version 9.5
for Linux, UNIX, and Windows



Getting Started with Database Application Development
Updated April, 2009

GC23-5856-02

Note

Before using this information and the product it supports, read the general information under Appendix B, "Notices," on page 89.

Edition Notice

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

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

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

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

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

Contents

About this book v

Chapter 1. The DB2 database application development environment 1

Chapter 2. Support for elements of the database application development environment 3

Chapter 3. Supported operating systems for database application development . . 5
Installation requirements for DB2 servers and IBM data server clients (AIX) 5
Installation requirements for DB2 servers and IBM data server clients (HP-UX) 7
Installation requirements for DB2 servers and IBM data server clients (Linux) 10
Installation requirements for DB2 servers and IBM data server clients (Solaris Operating Environment) . 11
Installation requirements for DB2 servers and IBM data server clients (Windows) 12

Chapter 4. IBM Data Server Client support for database application development 15

Chapter 5. Supported database application programming interfaces . . 17
ADO.NET development for IBM Data Servers . . . 17
IBM Data Server Provider for .NET 17
ODBC .NET Data Provider 18
OLE DB .NET Data Provider 19
Introduction to DB2 Call Level Interface and ODBC 20
Introduction to embedded SQL 21
Supported drivers for JDBC and SQLJ 22
IBM OLE DB Provider for DB2 24
Perl DBI 25
Introduction to PHP application development for DB2 25
The IBM_DB Ruby driver and Rails adapter . . . 26

Chapter 6. Supported programming languages and compilers for database application development 29
Supported .NET development software 29
Support for database application development in C 30
Support for database application development in C++ 31
Support for database application development in COBOL. 33
Support for database application development in Fortran 34
Java software support for DB2 products 34
Support for database application development in Perl 36
Support for database application development in PHP 37
Support for database application development in Ruby/Ruby on Rails 38
Support for database application development in REXX 38

Chapter 7. Supported transaction managers 41

Chapter 8. Supported database application development tools 43

Chapter 9. Configuring the database application development environment . 45
Configuring the operating system for database application development 46
Setting up the UNIX application development environment 46
Setting up the Windows application development environment 47
Configuring the IBM Data Server Client for database application development 48
Configuring the development environment for application programming interfaces 48
Setting up your application development environment to run DB2 CLI and ODBC applications 49
Installing the IBM Data Server Driver for JDBC and SQLJ 50
Transaction managers 53
X/Open distributed transaction processing model 54
Setting up the unixODBC driver manager . . . 75

Appendix A. Overview of the DB2 technical information 79
DB2 technical library in hardcopy or PDF format . . 79
Ordering printed DB2 books. 82
Displaying SQL state help from the command line processor 83
Accessing different versions of the DB2 Information Center 83
Displaying topics in your preferred language in the DB2 Information Center 83
Updating the DB2 Information Center installed on your computer or intranet server 84
DB2 tutorials 86
DB2 troubleshooting information 86
Terms and Conditions 86

Appendix B. Notices 89

Index 93

About this book

The Getting Started with Application Development Guide provides key concepts for supporting and configuring your database applications in the following key areas:

- database application programming interfaces
- database application programming languages
- compilers
- transaction managers
- development tools
- operating systems

The information presented in this book will help to provide you with the necessary resources for successful database application development.

Chapter 1. The DB2 database application development environment

The DB2® database application development environment is composed of several software elements:

- **operating system**
- **IBM® Data Server Client**
- **database application programming interface (API)**
- **programming language**
- **transaction manager**
- **development tools**

For many of these elements, there are multiple choices available. For example: there might be several operating systems available for your hardware; DB2 database supports several APIs; and any XA-compliant transaction manager can be used to manage distributed transactions with DB2 databases.

Chapter 2. Support for elements of the database application development environment

The following support is provided for database application development on DB2 products:

- Supported operating systems
- IBM data server client support
- Chapter 5, “Supported database application programming interfaces,” on page 17
- Supported programming languages and compilers
- Chapter 7, “Supported transaction managers,” on page 41
- Chapter 8, “Supported database application development tools,” on page 43

Chapter 3. Supported operating systems for database application development

You can develop DB2 database applications on the following operating systems:

- AIX®
- HP-UX
- Linux®
- Solaris
- Windows®

There are restrictions about when you can use available 32-bit or 64-bit architectures:

- With Linux on x86 and Windows operating systems on x86, only 32-bit DB2 database instances are supported. On 64-bit Windows operating system, both the 32-bit and 64-bit instances are supported. On all other operating systems, only 64-bit DB2 database instances are supported.
- You can run 32-bit database applications with either a 32-bit or 64-bit version of one of the IBM data server clients and develop with either the 32-bit or 64-bit IBM Data Server Client
- The steps for building applications in a 32-bit environment are sometimes different from the steps for building applications in a 64-bit environment.

Installation requirements for DB2 servers and IBM data server clients (AIX)

Before you install DB2 database products on AIX operating systems, ensure that the system you choose meets the necessary operating system, hardware, software, and communications requirements.

To install a DB2 database product, the following requirements must be met:

Table 1. AIX installation requirements

Operating System	**Hardware**
AIX Version 5.3 • 64-bit AIX kernel is required • AIX 5.3 Technology Level (TL) 6 and Service Pack (SP) 2 plus APAR IZ03063 • Minimum C++ runtime level requires the xlC.rte 9.0.0.1 and xlC.aix50.rte 9.0.0.1 filesets. These filesets are included in the August 2007 IBM C++ Runtime Environment Components for AIX package. AIX Version 6.1[2] • 64-bit AIX kernel is required • Minimum C++ runtime level requires the xlC.rte 9.0.0.1 and xlC.aix61.rte 9.0.0.1 filesets. These filesets are included in the October 2007 IBM C++ Runtime Environment Components for AIX package.	64-bit Common Hardware Reference Platform (CHRP) architecture[1] All processors that are capable of running the supported AIX operating systems.

- [1]To verify that it is a CHRP architecture system, issue the command lscfg and look for the following output: `Model Architecture: chrp`
- [2]In AIX 6.1 there are two types of Workload Partitions (WPARs): System WPARs and Application WPARs. DB2 installation is supported only on a System WPAR. AIX 6.1 also supports the ability to encrypt a JFS2 file system or set of files. This feature is not supported if you are using multiple partition instances.

Software considerations

- For application development and runtime considerations, see the topics in Supported programming languages and compilers for database application development.
- You can download the latest IBM C++ Runtime Environment Components for AIX at the IBM AIX XL C and C++ support web site.
- (Clients only) If you plan to use Kerberos Authentication, you require IBM Network Authentication Service client v1.4 or later. The NAS client can be downloaded from https://www6.software.ibm.com/dl/dm/dm-nas-p.
- Use the bosboot command to switch to the 64-bit kernel.

  To switch to a 64-bit kernel, you require root authority and should enter the following commands:

  ```
  ln -sf /usr/lib/boot/unix_64 /unix
  ln -sf /usr/lib/boot/unix_64 /usr/lib/boot/unix
  bosboot -a
  shutdown -Fr
  ```
- One of the following browsers is required to view online help and to run First Steps (db2fs):
  - Mozilla 1.4 and up
  - Firefox 1.0 and up
  - Netscape 7.0 and up

- An X Window System software capable of rendering a graphical user interface is required if you want to use the DB2 Setup wizard to install a DB2 product on Linux or UNIX® operating systems
- For details regarding known AIX issues, see www.ibm.com/support/docview.wss?&uid=swg21165448

Installation requirements for DB2 servers and IBM data server clients (HP-UX)

To install a DB2 product, the following operating system, hardware, and communications requirements must be met:

Table 2. HP-UX installation requirements

Operating System	Hardware
DB2 products are supported on: • HP-UX 11iv2 (11.23.0505) with: – May 2005 Base Quality (QPKBASE) bundle – May 2005 Applications Quality (QPKAPPS) bundle • HP-UX 11iv3 (11.31)	Itanium® based HP Integrity Series Systems

Kernel configuration considerations

A system restart is required if you update the kernel configuration parameters. The kernel configuration parameters are set in /etc/system. Depending on the values of your kernel configuration parameters, you might need to modify some of them before you install the Version 9 client or DB2 server products. If the kernel parameter being modified is not listed as dynamic, a system reboot is required to make the changes to /etc/system take effect.

Software considerations

- One of the following browsers is required to view online help and to run First Steps (db2fs):
  - Mozilla 1.4 and up
  - Firefox 1.0 and up
  - Netscape 7.0 and up
- An X Window System software capable of rendering a graphical user interface is required if:
  - you want to use the DB2 Setup wizard to install a DB2 product on Linux or UNIX operating systems
- For details regarding known HP-UX issues, see www.ibm.com/support/docview.wss?&uid=swg21257602

Installation requirements for DB2 servers and IBM data server clients (Linux)

Before you install DB2 database products on Linux operating systems, ensure that the system you choose meets the necessary operating system, hardware, software, and communications requirements.

For the latest information on supported Linux distributions, point your browser to http://www.ibm.com/software/data/db2/linux/validate/.

DB2 database products are supported on the following hardware:

- x86 (Intel® Pentium®, Intel Xeon™, and AMD) 32-bit Intel and AMD processors
- x64 (64-bit AMD64 and Intel EM64T processors)
- POWER™ (IBM eServer™ OpenPower®, System i™ or pSeries® systems that support Linux)
- eServer System z™ or System z9®

The supported operating systems for Linux include:

- Red Hat Enterprise Linux (RHEL) 4 Update 4
- Red Hat Enterprise Linux (RHEL) 5
- SUSE Linux Enterprise Server (SLES) 9 Service Pack 3
- SUSE Linux Enterprise Server (SLES) 10 Service Pack 1

Note: POWER requires a minimum of SLES 10 Service Pack 1 or RHEL 5

Multithreaded architecture limitations

If you are installing a DB2 Version 9.5 32-bit database product on a Linux operating system, consider upgrading to a 64-bit operating system and installing the DB2 Version 9.5 64-bit database product instead. The multithreaded architecture generally simplifies memory configuration. However, this could affect the memory configuration of 32-bit DB2 database servers. For example:

- Private memory for agent threads is allocated within a single process. The aggregate of all private memory allocations for database agents might not fit in a single process memory space.
- Support for multiple databases is limited because all database shared memory segments for all databases are allocated in a single process. You might need to reduce the memory usage for some databases in order to activate all databases successfully at the same time. However, the database manager performance might be impacted. Another alternative is to create multiple instances and catalog the databases across the instances. However, sufficient system resources is required to support this configuration.

Distribution Requirements

You should update your kernel configuration parameters in preparation for your Linux distribution. The default values for particular kernel parameters might not be sufficient when running a DB2 database system.

You might also have other products or applications that require Linux system resources. You should modify the kernel configuration parameters based on the needs of your Linux system working environment.

The kernel configuration parameters are set in /etc/sysctl.conf.

Refer to your operating system manual for information on setting and activating these parameters using the sysctl command.

Package requirements

The following tables list the package requirements for SLES and RHEL distributions for DB2 Version 9.5:

- libaio.so.1 is required for DB2 servers using asynchronous i/o.
- libstdc++so.5 is required for DB2 servers and clients.

Package requirements for SLES and RHEL

Package name	Description
libaio	contains the asynchronous library required for DB2 servers.
compat-libstdc++	contains libstdc++so.5 (not required for Linux on POWER)

The following tables list the package requirements for SUSE Linux and Red Hat distributions for DB2 Version 9.5 partitioned servers.

- The ksh93 Korn shell is required for SUSE10 and RHEL5 systems. The pdksh Korn Shell package is required for all other DB2 database systems.
- A remote shell utility is required for partitioned database systems. DB2 supports the following remote shell utilities:
  - rsh
  - ssh

  By default, DB2 uses rsh when executing commands on remote DB2 nodes, for example, when starting a remote DB2 database partition. To use the DB2 default, the rsh-server package must be installed (see table below). More information on rsh and ssh is available in the DB2 Information Center.

  If you choose to use the rsh remote shell utility, inetd (or xinetd) must be installed and running as well. If you choose to use the ssh remote shell utility, you need to set the **DB2RSHCMD** communication variable immediately after the DB2 installation is complete. If this registry variable is not set, rsh is used.
- The nfs-utils Network File System support package is required for partitioned database systems.

All required packages should be installed and configured before continuing with the DB2 setup. For general Linux information, see your Linux distribution documentation.

Package requirements for SUSE Linux

Package name	Description
pdksh or ksh93	Korn Shell. This package is required for partitioned database environments.
openssh	This package contains a set of server programs which allow users to run commands on (and from) remote computers via a secure shell. This package is not required if you use the default configuration of DB2 with rsh.
rsh-server	This package contains a set of server programs which allow users to run commands on remote computers, login in to other computers, and copy files between computers (rsh, rexec, rlogin, and rcp). This package is not required if you configure DB2 to use ssh.
nfs-utils	Network File System support package. It allows access to local files from remote computers.

Package requirements for Red Hat

Directory	Package name	Description
/System Environment/Shell	pdksh or ksh93	Korn Shell. This package is required for partitioned database environments.
/Applications/Internet	openssh	This package contains a set of client programs which allow users to run commands on a remote computer via a secure shell. This package is not required if you use the default configuration of DB2 with rsh.
/System Environment/ Daemons	openssh-server	This package contains a set of server programs which allow users to run commands from a remote computer via a secure shell. This package is not required if you use the default configuration of DB2 with rsh.
/System Environment/ Daemons	rsh-server	This package contains a set of programs which allow users to run commands on a remote computer. Required for partitioned database environments. This package is not required if you configure DB2 to use ssh.
/System Environment/ Daemons	nfs-utils	Network File System support package. It allows access to local files from remote computers.

Software considerations

- (Clients only) If you plan to use Kerberos Authentication, you require IBM Network Authentication Service client v1.4 or later. The NAS client can be downloaded from https://www6.software.ibm.com/dl/dm/dm-nas-p.
- One of the following browsers is required to view online help and to run First Steps (db2fs):
  - Mozilla 1.4 and up
  - Firefox 1.0 and up
  - Netscape 7.0 and up
- An X Window System software capable of rendering a graphical user interface is required if:
  - you want to use the DB2 Setup wizard to install a DB2 product on Linux or UNIX operating systems, or
  - you want to use any DB2 graphical tools on Linux for x86 and Linux on AMD 64/EM64T.

Security-enhanced Linux considerations

On RHEL 4 and RHEL 5 systems, if Security-enhanced Linux (SELinux) is enabled and in enforcing mode, the installer might fail due to SELinux restrictions.

To determine if SELinux is installed and in enforcing mode, you can do one of the following:

- check the /etc/sysconfig/selinux file
- run the **sestatus** command

- check the /var/log/messages file for SELinux notices (Notice format might differ between RHEL 4 and RHEL 5.)

To disable SELinux, you can do one of the following:

- set it in permissive mode and run the **setenforce 0** command as a superuser
- modify /etc/sysconfig/selinux and reboot the machine.

If your DB2 product installs successfully on a RHEL 4 or RHEL 5 system, DB2 processes will run in the unconfined domain. To assign DB2 processes to their own domains, modify the policy. A sample SELinux policy is provided in the sqllib/samples directory.

Installation requirements for DB2 servers and IBM data server clients (Solaris Operating Environment)

To install a DB2 database product, the following operating system, hardware, and communications requirements must be met:

Table 3. Solaris installation requirements

Operating System	**Hardware**
Solaris 9 • 64- bit kernel • Patches 111711-12 and 111712-12 • If raw devices are used, patch 122300-11 • 64-bit Fujitsu PRIMEPOWER and Solaris 9 Kernel Update Patch 112233-01 or later to get the fix for patch 912041-01 Solaris 10 • 64- bit kernel • If raw devices are used, patch 125100-07	UltraSPARC or SPARC64 processors
Solaris 10 • 64- bit kernel • Patch 118855-33 • If raw devices are used, patch 125101-07	Solaris x64 (Intel 64 or AMD64)

Kernel configuration considerations

The kernel configuration parameters are set in /etc/system. If the kernel parameter being modified is not listed as dynamic, a system reboot is required to make the changes to /etc/system take effect. These parameters must be set before you install an IBM data server client.

Software considerations

- (Clients only) If you plan to use Kerberos Authentication, you require Solaris 9 or higher with IBM Network Authentication Service (NAS) client v1.4 or higher. The NAS client can be downloaded from Web site: https://www6.software.ibm.com/dl/dm/dm-nas-p.
- One of the following browsers is required to view online help and to run First Steps (db2fs):
  - Mozilla 1.4 and up

- Firefox 1.0 and up
- Netscape 7.0 and up

- An X Window System software capable of rendering a graphical user interface is required if:
  - you want to use the DB2 Setup wizard to install a DB2 database product on Linux or UNIX operating systems
- For details regarding known Solaris issues, see www.ibm.com/support/docview.wss?&uid=swg21257606

Security patches can be obtained from the http://sunsolve.sun.com Web site. From the SunSolve Online Web site, click on the ″Patches″ menu item in the left panel.

The Java2 Standard Edition (J2SE) Solaris Operating System Patch Clusters and the SUNWlibC software are also required and can be obtained from the http://sunsolve.sun.com Web site.

For DB2 on 64-bit Fujitsu PRIMEPOWER systems, you require the following:

- Solaris 9 Kernel Update Patch 112233-01 or later to get the fix for patch 912041-01.

The Fujitsu PRIMEPOWER patches for the Solaris Operating Environment can be downloaded from FTSI at: http://download.ftsi.fujitsu.com/.

Installation requirements for DB2 servers and IBM data server clients (Windows)

Before you install a DB2 database product on Windows operating systems, ensure that the system you choose meets the necessary operating system, hardware, and software requirements.

Table 4. Windows workstation platforms

Operating System	Prerequisites	Hardware
Windows XP Professional (32-bit and 64-bit) Windows Vista Business (32-bit and 64-bit) Windows Vista Enterprise (32-bit and 64-bit) Windows Vista Ultimate (32-bit and 64-bit)	Windows XP Service Pack 2 or later IBM Data Server Provider for .NET client applications and CLR server-side procedures require .NET 1.1 SP1 or .NET 2.0 framework runtime 64-bit IBM data server provider for .NET applications are supported All Windows Vista service packs are supported.	All Intel and AMD processors capable of running the supported Windows operating systems (32-bit and x64 based systems).

Table 5. Windows server platforms

Operating System	Prerequisites	Hardware
Windows Server 2003 Datacenter Edition (32-bit and 64-bit) Windows Server 2003 Enterprise Edition (32-bit and 64-bit) Windows Server 2003 Standard Edition (32-bit and 64-bit)	Service Pack 1 or later. R2 is also supported IBM data server provider for .NET client applications and CLR server-side procedures require .NET 1.1 SP1 or .NET 2.0 framework runtime 64-bit IBM data server provider for .NET applications are supported	All Intel and AMD processors capable of running the supported Windows operating systems (32-bit and x64 based systems).
Windows Server 2008 Datacenter Edition (32-bit and 64-bit) Windows Server 2008 Enterprise Edition (32-bit and 64-bit) Windows Server 2008 Standard Edition (32-bit and 64-bit)	IBM data server provider for .NET client applications and CLR server-side procedures require .NET 1.1 SP1 or .NET 2.0 framework runtime 64-bit IBM data server provider for .NET applications are supported All Windows Server 2008 service packs are supported.	

Note:

- DB2 database products support the hardware-enforced Data Execution Prevention (DEP) feature that is built into some Windows operating systems.
- Federation (federated systems, servers and databases) is currently not supported by DB2 database products on Windows Server 2008.
- To use Windows Server 2008 Failover Clusters for failover support of partitioned DB2 database systems, you must install DB2 Version 9.5 Fix Pack 3 (or later fix packs).

Additional software considerations

- Windows Installer 3.0 is required. It will be installed by the installer if is it not detected.
- IBM Data Server Provider for .NET client applications and CLR server-side procedures require .NET 1.1 SP1 or .NET 2.0 framework runtime. In an x64 environment, 32-bit IBM data server provider for .NET applications will run in the WOW64 emulation mode.
- MDAC 2.8 is required. The DB2 Setup wizard will install MDAC 2.8 if it is not already installed.

  Note: If a previous version of MDAC (for example, 2.7) is already installed, DB2 install will upgrade MDAC to 2.8. For a typical install, MDAC 2.8 is installed. For a custom install, MDAC 2.8 is installed but only if you have not deselected the default which is to install it. If you deselect MDAC as part of a custom install, it will not be installed.
- If you plan to use LDAP (Lightweight Directory Access Protocol), you should use either a Microsoft® LDAP client or the IBM Tivoli® Directory Server v6 client (also known as the IBM LDAP client which is included

with DB2 products). Prior to installation of the Microsoft Active Directory, you will need to extend your directory schema using the db2schex utility, which can be found on the installation media under the db2\Windows\utilities directory.

The Microsoft LDAP client is included with Windows operating systems.

- One of the following browsers is required to view online help, run the DB2 install launchpad (setup.exe), and to run First Steps (db2fs):
  - Internet Explorer 6 and up
  - Mozilla 1.4 and up
  - Firefox 1.0 and up
  - Netscape 7.0 and up

Chapter 4. IBM Data Server Client support for database application development

The Data Server Client includes the following:

- **Precompilers for C/C++, COBOL, and Fortran**, (providing the language is supported for that platform).
- **Embedded SQL application support**, including programming libraries, include files and code samples.
- **ODBC and DB2 Call Level Interface (DB2 CLI) application support**, including programming libraries, include files, and code samples for developing applications which are easily ported to ODBC and compiled with an ODBC SDK. An ODBC SDK is available from Microsoft for Windows operating systems, and from various other vendors for many of the other supported platforms. On Windows operating systems, the ODBC and CLI driver is installed by default with the DB2 Client, supporting applications developed with the Microsoft ODBC Software Developer's Kit. For all other platforms, the ODBC and CLI driver can optionally be installed with the DB2 Client, supporting applications that can be developed with an ODBC SDK for that platform, if one exists.
- **The IBM Data Server Driver for JDBC and SQLJ**, which includes:
  - Support for JDBC 3 and JDBC 4 compliant applications
  - SQLJ support, for developing Java™ applications with static SQL
- **Java 5** environment is embedded in DB2 servers to support server-side Java application artifacts including stored procedures and user-defined functions
- **ActiveX Data Objects (ADO) and Object Linking and Embedding (OLE) automation routines (UDFs and Stored Procedures)** on Windows operating systems, including code samples implemented in Microsoft Visual Basic and Microsoft Visual C++.
- **Object Linking and Embedding Database (OLE DB) table functions** on Windows operating systems.
- **C# and Visual Basic .NET applications and CLR .NET routines** on Windows operating systems.
- **Ruby and Ruby on Rails application support** , including IBM_DB gem for easy install and code samples. Ruby is a fully-integrated object-oriented programming language used to develop web applications. Ruby on Rails (RoR), also known as Rails, is a framework for developing web applications that access databases according to the Model-View-Control architectural framework
- **PHP Hypertext Preprocessor application support** , including programming libraries and code samples. PHP is a scripting language used to develop web applications. It also supports object-oriented programming
- **IBM Data Studio** is a comprehensive suite of integrated Eclipse tools for the database developer and development DBA. IBM Data Studio reduces the time to perform day-to-day administration tasks, create, deploy, and debug SQL and Java stored procedures, deploy data-centric web services, and create queries for relational and XML data using SQL and XQuery for DB2 and Informix data servers.
- **Interactive SQL** through the Command Editor or Command Line Processor (CLP) to prototype SQL statements or to perform ad hoc queries against the database.

- **A set of documented APIs** to enable other application development tools to implement precompiler support for DB2 directly within their products. For example, IBM COBOL on AIX uses this interface.

Chapter 5. Supported database application programming interfaces

You can use any of the following programming interfaces to manage or access DB2 databases. You can:

- Use DB2 APIs to perform administrative functions such as backing up and restoring databases.
- Embed static and dynamic SQL statements in your applications programmed in C, C++, COBOL, FORTRAN, and REXX™.
- Code DB2 Call Level Interface (DB2 CLI) function calls in your applications to invoke dynamic SQL statements.
- Develop Java applications and applets using the Java Database Connectivity application programming interface (JDBC API) or embedded SQL for Java (SQLJ).
- Develop Microsoft Visual Basic and Visual C++ applications using ActiveX Data Object (ADO).
- Develop ADO.NET applications using IBM Data Server Provider for .NET (DB2 .NET), OLE DB .NET Data Provider or ODBC .NET Data Provider.
- Develop applications using IBM or third-party tools such as Excel, Perl, and Open Database Connectivity (ODBC) end-user tools such as Lotus® Approach®, and its programming language, LotusScript.
- Develop web applications using PHP scripting language and Ruby on Rails (RoR) which a framework for developing web applications that access databases.
- Develop web applications using Python, which is a general purpose, high-level scripting language that is well-suited for rapid application development.

ADO.NET development for IBM Data Servers

The IBM Data Server Provider for .NET is an extension of the ADO.NET interface that allows .NET applications to access an IBM data server through a secure connection, execute commands, and retrieve result sets.

Reference documentation is included with the IBM Data Server Provider for .NET when installing the IBM Data Server Client, presenting detailed information about all the IBM Data Server Provider for .NET objects and their members. During the DB2 installation process, this documentation is registered with Microsoft Visual Studio .NET. To view the IBM Data Server Provider for .NET documentation from Microsoft Visual Studio .NET, select the `Help` menu option, and `Contents`. Once the help viewer opens, filter by *IBM Database Documentation*.

IBM Data Server Provider for .NET

The IBM Data Server Provider for .NET extends DB2 data server support for the ADO.NET interface. The IBM Data Server Provider for .NET delivers high-performing, secure access to IBM data servers.

The IBM Data Server Provider for .NET allows your .NET applications to access the following database management systems:

- DB2 Version 9 (or later) for Linux, UNIX, and Windows
- DB2® Universal Database™ Version 8 for Windows, UNIX, and Linux

- DB2 for z/OS® and OS/390® Version 6 (or later), through DB2® Connect™
- DB2 for i5/OS® Version 5 (or later), through DB2 Connect
- DB2 Universal Database Version 7.3 (or later) for VSE & VM, through DB2 Connect
- IBM Informix® Dynamic Server, Version 11.10 or later
- IBM UniData®, Version 7.1.11 or later
- IBM UniVerse®, Version 10.2 or later

To develop and run applications that use Data Server Provider for .NET you need the .NET Framework

In addition to the IBM Data Server Provider for .NET, the IBM Database Development Add-Ins enable you to quickly and easily develop .NET applications for IBM data servers using Microsoft Visual Studio. You can also use the Add-Ins to create database objects such as indexes and tables, and develop server-side objects, such as stored procedures and user-defined functions.

ODBC .NET Data Provider

The ODBC .NET Data Provider makes ODBC calls to a DB2 data source using the DB2 CLI Driver. Therefore, the connection string keywords supported by the ODBC .NET Data Provider are the same as those supported by the DB2 CLI driver. Also, the ODBC .NET Data Provider has the same restrictions as the DB2 CLI driver. There are additional restrictions for the ODBC .NET Data Provider, which are identified in the topic: ODBC .NET Data Provider restrictions.

In order to use the ODBC .NET Data Provider, you must have the .NET Framework Version 1.1, 2.0, or 3.0 installed. For DB2 Universal Database for AS/400® and iSeries®, the following fix is required on the server: APAR II13348.

The following are the supported connection keywords for the ODBC .NET Data Provider:

Table 6. Useful ConnectionString keywords for the ODBC .NET Data Provider

Keyword	Value	Meaning
`DSN`	database alias	The DB2 database alias as cataloged in the database directory
`UID`	user ID	The user ID used to connect to the DB2 server
`PWD`	password	The password for the user ID used to connect to the DB2 server

Note: For the full list of ConnectionString keywords, see the Microsoft documentation.

The following is an example of creating an `OdbcConnection` to connect to the SAMPLE database:

```
[Visual Basic .NET]
Dim con As New OdbcConnection("DSN=sample;UID=userid;PWD=password;")
con.Open()
```

```
[C#]
OdbcConnection con = new OdbcConnection("DSN=sample;UID=userid;PWD=password;");
con.Open()
```

OLE DB .NET Data Provider

The OLE DB .NET Data Provider uses the IBM DB2 OLE DB Driver, which is referred to in a ConnectionString object as IBMDADB2. The connection string keywords supported by the OLE DB .NET Data Provider are the same as those supported by the IBM OLE DB Provider for DB2. Also, the OLE DB .NET Data Provider has the same restrictions as the IBM DB2 OLE DB Provider. There are additional restrictions for the OLE DB .NET Data Provider, which are identified in the topic: OLE DB .NET Data Provider restrictions.

In order to use the OLE DB .NET Data Provider, you must have the .NET Framework Version 1.1, 2.0, or 3.0 installed.

For DB2 Universal Database for AS/400 and iSeries, the following fix is required on the server: APAR ii13348.

The following are all the supported connection keywords for the OLE DB .NET Data Provider:

Table 7. Useful ConnectionString keywords for the OLE DB .NET Data Provider

Keyword	Value	Meaning
PROVIDER	IBMDADB2	Specifies the IBM OLE DB Provider for DB2 (required)
DSN or Data Source	database alias	The DB2 database alias as cataloged in the database directory
UID	user ID	The user ID used to connect to the DB2 data server
PWD	password	The password for the user ID used to connect to the DB2 data server

Note: For the full list of ConnectionString keywords, see the Microsoft documentation.

The following is an example of creating an OleDbConnection to connect to the SAMPLE database:

```
[Visual Basic .NET]
Dim con As New OleDbConnection("Provider=IBMDADB2;" +
    "Data Source=sample;UID=userid;PWD=password;")
con.Open()

[C#]
OleDbConnection con = new OleDbConnection("Provider=IBMDADB2;" +
                "Data Source=sample;UID=userid;PWD=password;" );
con.Open()
```

Introduction to DB2 Call Level Interface and ODBC

DB2 Call Level Interface (DB2 CLI) is IBM’s callable SQL interface to the DB2 family of database servers. It is a ’C’ and ’C++’ application programming interface for relational database access that uses function calls to pass dynamic SQL statements as function arguments. It is an alternative to embedded dynamic SQL, but unlike embedded SQL, DB2 Call Level Interface does not require host variables or a precompiler.

DB2 Call Level Interface is based on the Microsoft** Open Database Connectivity** (ODBC) specification, and the International Standard for SQL/CLI. These specifications were chosen as the basis for the DB2 Call Level Interface in an effort to follow industry standards and to provide a shorter learning curve for those application programmers already familiar with either of these database interfaces. In addition, some DB2 specific extensions have been added to help the application programmer specifically exploit DB2 features.

The DB2 Call Level Interface driver also acts as an ODBC driver when loaded by an ODBC driver manager. It conforms to ODBC 3.51.

DB2 Call Level Interface Background information

To understand DB2 Call Level Interface or any callable SQL interface, it is helpful to understand what it is based on, and to compare it with existing interfaces.

The X/Open Company and the SQL Access Group jointly developed a specification for a callable SQL interface referred to as the *X/Open Call Level Interface*. The goal of this interface is to increase the portability of applications by enabling them to become independent of any one database vendor’s programming interface. Most of the X/Open Call Level Interface specification has been accepted as part of the ISO Call Level Interface International Standard (ISO/IEC 9075-3:1995 SQL/CLI).

Microsoft developed a callable SQL interface called Open Database Connectivity (ODBC) for Microsoft operating systems based on a preliminary draft of X/Open CLI.

The ODBC specification also includes an operating environment where database specific ODBC Drivers are dynamically loaded at run time by a driver manager based on the data source (database name) provided on the connect request. The application is linked directly to a single driver manager library rather than to each DBMS’s library. The driver manager mediates the application’s function calls at run time and ensures they are directed to the appropriate DBMS specific ODBC driver. Since the ODBC driver manager only knows about the ODBC-specific functions, DBMS-specific functions cannot be accessed in an ODBC environment. DBMS-specific dynamic SQL statements are supported via a mechanism called an escape clause.

ODBC is not limited to Microsoft operating systems; other implementations are available on various platforms.

The DB2 Call Level Interface load library can be loaded as an ODBC driver by an ODBC driver manager. For ODBC application development, you must obtain an ODBC Software Development Kit. For the Windows platform, the ODBC SDK is available as part of the Microsoft Data Access Components (MDAC) SDK, available for download from http://www.microsoft.com/data/. For non-Windows platforms, the ODBC SDK is provided by other vendors. When developing ODBC applications that may connect to DB2 servers, use the Call Level Interface Guide and Reference, Volume 1 and the Call Level Interface Guide and Reference, Volume 2 (for information on DB2 specific extensions and diagnostic information), in conjunction with the ODBC Programmer's Reference and SDK Guide available from Microsoft.

Applications written directly to DB2 Call Level Interface link directly to the DB2 Call Level Interface load library. DB2 Call Level Interface includes support for many ODBC and ISO SQL/CLI functions, as well as DB2 specific functions.

The following DB2 features are available to both ODBC and DB2 Call Level Interface applications:

- double byte (graphic) data types
- stored procedures
- Distributed Unit of Work (DUOW), two phase commit
- compound SQL
- user defined types (UDT)
- user defined functions (UDF)

Introduction to embedded SQL

Embedded SQL database applications connect to databases and execute embedded SQL statements. Embedded SQL statements are embedded within a host language application. Embedded SQL database applications support the embedding of SQL statements to be executed statically or dynamically.

You can develop embedded SQL applications for DB2 in the following host programming languages: C, C++, COBOL, FORTRAN, and REXX.

Note: Support for embedded SQL in FORTRAN and REXX has been deprecated and will remain at the DB2 Universal Database , Version 5.2 level.

Building embedded SQL applications involves two prerequisite steps prior to application compilation and linking.

- Preparing the source files containing embedded SQL statements using the DB2 precompiler.

  The `PREP` (`PRECOMPILE`) command is used to invoke the DB2 precompiler, which reads your source code, parses and converts the embedded SQL statements to DB2 run-time services API calls, and finally writes the output to a new modified source file. The precompiler produces access plans for the SQL statements, which are stored together as a package within the database.
- Binding the statements in the application to the target database.

  Binding is done by default during precompilation (the `PREP` command). If binding is to be deferred (for example, running the `BIND` command later), then the `BINDFILE` option needs to be specified at `PREP` time in order for a bind file to be generated.

Once you have precompiled and bound your embedded SQL application, it is ready to be compiled and linked using the host language-specific development tools.

To aid in the development of embedded SQL applications, you can refer to the embedded SQL template in C. Examples of working embedded SQL sample applications can also be found in the `%DB2PATH%\SQLLIB\samples` directory.

Note: %DB2PATH% refers to the DB2 installation directory

Static and dynamic SQL

SQL statements can be executed in one of two ways: statically or dynamically.

Statically executed SQL statements
For statically executed SQL statements, the syntax is fully known at precompile time. The structure of an SQL statement must be completely specified for a statement to be considered static. For example, the names for the columns and tables referenced in a statement must be fully known at precompile time. The only information that can be specified at run time are values for any host variables referenced by the statement. However, host variable information, such as data types, must still be precompiled. You precompile, bind, and compile statically executed SQL statements before you run your application. Static SQL is best used on databases whose statistics do not change a great deal.

Dynamically executed SQL statements
Dynamically executed SQL statements are built and executed by an application at run-time. An interactive application that prompts the end user for key parts of an SQL statement, such as the names of the tables and columns to be searched, is a good example of a situation suited for dynamic SQL.

Supported drivers for JDBC and SQLJ

The DB2 product includes support for two types of JDBC driver architecture.

According to the JDBC specification, there are four types of JDBC driver architectures:

Type 1
Drivers that implement the JDBC API as a mapping to another data access API, such as Open Database Connectivity (ODBC). Drivers of this type are generally dependent on a native library, which limits their portability. The DB2 database system does not provide a type 1 driver.

Type 2
Drivers that are written partly in the Java programming language and partly in native code. The drivers use a native client library specific to the data source to which they connect. Because of the native code, their portability is limited.

Type 3
Drivers that use a pure Java client and communicate with a database using a database-independent protocol. The database then communicates the client's requests to the data source. The DB2 database system does not provide a type 3 driver.

Type 4
Drivers that are pure Java and implement the network protocol for a specific data source. The client connects directly to the data source.

DB2 Database for Linux, UNIX, and Windows supports the following drivers:

Driver name	Packaged as	Driver type
DB2 JDBC Type 2 Driver for Linux, UNIX and Windows	db2java.zip	Type 2
IBM Data Server Driver for JDBC and SQLJ	• db2jcc.jar and sqlj.zip for JDBC 3.0 and earlier support • db2jcc4.jar and sqlj4.zip for support of some JDBC 4.0 functions	Type 2 and Type 4

IBM Data Server Driver for JDBC and SQLJ (type 2 and type 4)

The IBM Data Server Driver for JDBC and SQLJ is a single driver that includes JDBC type 2 and JDBC type 4 behavior. When an application loads the IBM Data Server Driver for JDBC and SQLJ, a single driver instance is loaded for type 2 and type 4 implementations. The application can make type 2 and type 4 connections using this single driver instance. The type 2 and type 4 connections can be made concurrently. IBM Data Server Driver for JDBC and SQLJ type 2 driver behavior is referred to as *IBM Data Server Driver for JDBC and SQLJ type 2 connectivity.* IBM Data Server Driver for JDBC and SQLJ type 4 driver behavior is referred to as *IBM Data Server Driver for JDBC and SQLJ type 4 connectivity.*

Two versions of the IBM Data Server Driver for JDBC and SQLJ are available. IBM Data Server Driver for JDBC and SQLJ version 3.5*x* is JDBC 3.0-compliant. IBM Data Server Driver for JDBC and SQLJ version 4.*x* is JDBC 3.0-compliant and supports some JDBC 4.0 functions.

The IBM Data Server Driver for JDBC and SQLJ supports these JDBC and SQLJ functions:

- All of the methods that are described in the JDBC 3.0 specifications. See ″Driver support for JDBC APIs″.
- Some methods that are described in the JDBC 4.0 specifications, if you install IBM Data Server Driver for JDBC and SQLJ version 4.0.
- SQLJ application programming interfaces, as defined by the SQLJ standards, for simplified data access from Java applications.
- Connections that are enabled for connection pooling. WebSphere® Application Server or another application server does the connection pooling.
- Connections to a database within Java user-defined functions and stored procedures (IBM Data Server Driver for JDBC and SQLJ type 2 connectivity only. Calling applications can use type 2 connectivity or type 4 connectivity.)

  The IBM Data Server Driver for JDBC and SQLJ is the default driver for Java routines.
- Support for distributed transaction management. This support implements the Java 2 Platform, Enterprise Edition (J2EE) Java Transaction Service (JTS) and Java Transaction API (JTA) specifications, which conform to the X/Open standard for distributed transactions (*Distributed Transaction Processing: The XA Specification*, available from http://www.opengroup.org).

DB2 JDBC Type 2 Driver for Linux, UNIX and Windows (DB2 JDBC type 2 driver) (deprecated)

The DB2 JDBC type 2 driver lets Java applications make calls to DB2 through JDBC. Calls to the DB2 JDBC type 2 driver are implemented with Java native methods. The DB2 JDBC Type 2 Driver uses the DB2 CLI interface to communicate with DB2 databases. The Java applications that use this driver must run on a DB2 client, through which JDBC requests flow to the DB2 database. DB2 Connect must be installed before the DB2 JDBC application driver can be used to access DB2 for i data sources or data sources in the DB2 for z/OS environments.

The DB2 JDBC type 2 driver supports these JDBC and SQLJ functions:

- Most of the methods that are described in the JDBC 1.2 specification, and some of the methods that are described in the JDBC 2.0 specification.
- SQLJ statements that perform equivalent operations to all JDBC methods
- Connection pooling
- Distributed transactions
- Java user-defined functions and stored procedures

The DB2 JDBC Type 2 Driver for Linux, UNIX and Windows will not be supported in future releases. You should therefore consider moving to the IBM Data Server Driver for JDBC and SQLJ.

IBM OLE DB Provider for DB2

The IBM OLE DB Provider for DB2 allows DB2 to act as a resource manager for the OLE DB provider. This support gives OLE DB-based applications the ability to extract or query DB2 data using the OLE interface.

Microsoft OLE DB is a set of OLE/COM interfaces that provides applications with uniform access to data stored in diverse information sources. The OLE DB architecture defines OLE DB consumers and OLE DB providers. An OLE DB consumer is any system or application that uses OLE DB interfaces; an OLE DB provider is a component that exposes OLE DB interfaces.

The IBM OLE DB Provider for DB2, whose provider name is IBMDADB2, enables OLE DB consumers to access data on a DB2 database server. If DB2 Connect is installed, these OLE DB consumers can also access data on a host DBMS such as DB2 for MVS™, DB2 for VM/VSE, or SQL/400®.

The IBM OLE DB Provider for DB2 offers the following features:

- Support level 0 of the OLE DB provider specification, including some additional level 1 interfaces.
- A free threaded provider implementation, which enables the application to create components in one thread and use those components in any other thread.
- An Error Lookup Service that returns DB2 error messages.

Note that the IBM OLE DB Provider resides on the client and is different from the OLE DB table functions, which are also supported by DB2 database systems.

Subsequent sections of this document describe the specific implementation of the IBM OLE DB Provider for DB2. For more information on the Microsoft OLE DB 2.0 specification, refer to the Microsoft OLE DB 2.0 Programmer's Reference and Data Access SDK, available from Microsoft Press.

Version Compliance

The IBM OLE DB Provider for DB2 complies with Version 2.7 or later of the Microsoft OLE DB specification.

System Requirements

Refer to the announcement letter for the IBM OLE DB Provider for DB2 data servers to see the supported Windows operating systems.

To install the IBM OLE DB Provider for DB2, you must first be running on one of the supported operating systems listed above. You also need to install the DB2 Client. This client includes Microsoft Data Access Components (MDAC).

Perl DBI

DB2 supports the Perl Database Interface (DBI) specification for data access through the DBD::DB2 driver. The DB2 Perl DBI website is located at:

```
http://www.ibm.com/software/data/db2/perl/
```

and contains the latest DBD::DB2 driver, and related information.

Perl is an interpreted language and the Perl DBI Module uses dynamic SQL. These properties make Perl an ideal language for quickly creating and revising DB2 applications. The Perl DBI Module uses an interface that is quite similar to the CLI and JDBC interfaces, which makes it easy to port Perl applications to CLI and JDBC, or to port CLI and JDBC applications to Perl.

Introduction to PHP application development for DB2

PHP: Hypertext Preprocessor (PHP) is an interpreted programming language primarily intended for the development of Web applications. The first version of PHP was created by Rasmus Lerdorf and contributed under an open source license in 1995. PHP was initially a very simple HTML templating engine, but over time the developers of PHP added database access functionality, rewrote the interpreter, introduced object-oriented support, and improved performance. Today, PHP has become a popular language for Web application development because of its focus on practical solutions and support for the most commonly required functionality in Web applications.

For the easiest install and configuration experience on Linux, UNIX, or Windows operating systems, you can download and install Zend Core for IBM for use in production systems. Paid support for Zend Core for IBM is available from Zend. On Windows, precompiled binary versions of PHP are available for download from http://php.net/. Most Linux distributions include a precompiled version of PHP. On UNIX operating systems that do not include a precompiled version of PHP, you can compile your own version of PHP.

PHP is a modular language that enables you to customize the available functionality through the use of extensions. These extensions can simplify tasks such as reading, writing, and manipulating XML, creating SOAP clients and servers, and encrypting communications between server and browser. The most popular extensions for PHP, however, provide read and write access to databases so that you can easily create a dynamic database-driven Web site.

We have built on our existing PHP support by developing a new extension called pdo_ibm for anyone who wishes to use the PHP Application Objects (PDO) interface. This new extension along with the existing ibm_db2 extension will now be conveniently included as part of the IBM Data Server Client. The most up to date versions of ibm_db2 and pdo_ibm are available from the PHP Extension Community Library (PECL) http://pecl.php.net/. You can use either extension to access data stored in a DB2 database through your PHP application. The differences between the extensions are detailed as follows:

- ibm_db2 is an extension written, maintained, and supported by IBM for access to DB2 databases. The ibm_db2 extension offers a procedural application programming interface (API) that, in addition to the normal create, read, update, and write database operations, also offers extensive access to the database metadata. You can compile the ibm_db2 extension with either PHP 4 or PHP 5.
- pdo_ibm is a driver for the PHP Data Objects (PDO) extension that offers access to DB2 databases through the standard object-oriented database interface introduced in PHP 5.1.

A third extension, Unified ODBC, has historically offered access to DB2 database systems. It is not recommended that you write new applications with this extension because ibm_db2 and pdo_ibm both offer significant performance and stability benefits over Unified ODBC. The ibm_db2 extension API makes porting an application that was previously written for Unified ODBC almost as easy as globally changing the `odbc_` function name to `db2_` throughout the source code of your application.

The IBM_DB Ruby driver and Rails adapter

With the introduction of support for the Ruby on Rails framework, Rails applications can now access data from IBM data servers.

Collectively known as the IBM_DB gem, the IBM_DB Ruby driver and Rails adapter allows Ruby applications to access the following database management systems:

- DB2 Database for Linux, UNIX, and Windows, Version 9.5
- DB2 Database for Linux, UNIX, and Windows, Version 9.1
- DB2 Universal Database Version 8 for Windows, UNIX, and Linux-based computers
- DB2 Universal Database Version 5, Release 1 (or later) for AS/400 and iSeries, through DB2 Connect
- DB2 z/OS Version 8 and Version 9, through DB2 Connect
- IBM Informix Dynamic Server, Version 11.10 or later

  Note: Clients should use IBM Data Server Client Version 9.5 when accessing Informix Dynamic Server Version 11.10. Previous versions are not supported. Clients may also use the IBM Data Server Runtime Client or the IBM Data Server Driver.

The IBM_DB Ruby driver can be used to connect to and access data from the IBM data servers mentioned above. The IBM_DB Ruby adapter allows any database-backed Ruby application to interface with IBM data servers.

For more information about IBM Ruby projects and the RubyForge open source community, refer to the following website: http://rubyforge.org/projects/rubyibm/

Click on the following link to see a list of installation requirements for DB2 database products: http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/topic/com.ibm.db2.luw.qb.server.doc/doc/r0025127.html

The following is a link to see a list of installation requirements for IDS: http://publib.boulder.ibm.com/infocenter/idshelp/v111/topic/com.ibm.expr.doc/ids_in_004x.html

Chapter 6. Supported programming languages and compilers for database application development

You can develop DB2 database applications in the following languages:

- C
- C++
- COBOL
- Fortran
- Java
- Perl
- PHP
- Python
- Ruby/Ruby on Rails
- REXX
- C#, VB .NET and other .NET languages

The supported APIs and programming languages for developing external routines are generally compatible with those for database application development, however there are some minor differences in prerequisites and support. To review the supported APIs and programming language for external routine development see: *Supported APIs and programming languages for external routine development*.

Supported .NET development software

To develop and deploy .NET applications that run against IBM data servers, you will need to use supported development software and operating systems.

Supported operating systems for developing and deploying .NET Framework 1.1 applications

- Windows 2000
- Windows XP (32-bit edition)
- Windows Server 2003 (32-bit edition)
- Windows Vista (32-bit editions)
- Windows Server 2008 (32-bit edition)

Supported operating systems for developing and deploying .NET Framework 2.0, 3.0 and 3.5 applications

- Windows 2000, Service Pack 3
- Windows XP, Service Pack 2 (32-bit and 64-bit editions)
- Windows Server 2003 (32-bit and 64-bit editions)
- Windows Vista (32-bit and 64-bit editions)
- Windows Server 2008 (32-bit and 64-bit editions)

Supported development software for .NET Framework applications

In addition to a DB2 client, you will need one of the following options to develop .NET Framework applications.

- Visual Studio 2003 (for .NET Framework 1.1 applications)
- Visual Studio 2005 (for .NET Framework 2.0 and 3.0 applications)
- Visual Studio 2008 (for .NET Framework 2.0, 3.0 and 3.5 applications)

Supported deployment software for .NET Framework applications

In addition to a DB2 runtime client , you will need one of the following three options to deploy .NET Framework applications. In most cases one of these are included with a Windows installation.

- .NET Framework Version 1.1 Redistributable Package (for .NET Framework 1.1 applications)
- .NET Framework Version 2.0 Redistributable Package (for .NET Framework 2.0 applications)
- .NET Framework Version 3.0 Redistributable Package (for .NET Framework 3.0 applications)
- .NET Framework Version 3.5 Redistributable Package (for .NET Framework 3.5 applications)

Support for database application development in C

The following table lists C compilers that are supported for DB2 database application development.

Unless otherwise noted, the compilers listed in the following table support both 32-bit and 64-bit application development.

Table 8. Supported C compilers

Operating System	Supported Compiler(s)
AIX	• IBM XL C/C++ Enterprise Edition Version 7.0 for AIX • IBM XL C/C++ Enterprise Edition Version 8.0 for AIX • IBM XL C/C++ Enterprise Edition Version 9.0 for AIX
HP-UX on IPF (IA64)	• For HP-UX 11i v2, use HP aC++ Compiler Version A.06.05 • For HP-UX 11i v3, use HP aC++ Compiler Version A.06.12
Linux on x86 (32-bit DB2 instances and compilers only)	• GNU/Linux gcc versions 3.3 and 3.4 • Intel C Compiler Version 9.1
Linux on AMD64/EM64T (x86-64, x64)	• GNU/Linux gcc versions 3.3 and 3.4 • Intel C Compiler Version 9.1
Linux on POWER (PowerPC®, iSeries, pSeries)	• GNU/Linux gcc versions 3.3 and 3.4 • IBM XL C/C++ Advanced Edition Version 8.0 for Linux
Linux on zSeries® (s/390x)	• GNU/Linux gcc versions 3.3 and 3.4

Table 8. Supported C compilers (continued)

Operating System	Supported Compiler(s)
Solaris on UltraSPARC	• Solaris Studio 10 • Forte C++ Version 6.2 • Sun ONE Studio 7, Compiler Collection
Windows on x86 (32-bit DB2 instances only)	• Intel Proton Compiler for Windows 32-bit applications, Version 9.0.021 or later • Microsoft Visual C++ 2005 • Microsoft Visual C++ .NET
Windows on X64 (x86-64, AMD64/EM64T)	• Microsoft Platform SDK for Windows Server 2003 SP1 (Build 1830 or later), C/C++ compiler for x64 • Intel Proton Compiler for Windows x64, Version 9.0.024 or later • Microsoft Visual C++ 2005

Support for database application development in C++

The following table lists C++ compilers that are supported for DB2 database application development.

Unless otherwise noted, the compilers listed in the following table support both 32-bit and 64-bit application development.

Table 9. Supported C++ compilers

Operating System	Supported Compiler(s)
AIX	• IBM XL C/C++ Enterprise Edition Version 7.0 for AIX • IBM XL C/C++ Enterprise Edition Version 8.0 for AIX • IBM XL C/C++ Enterprise Edition Version 9.0 for AIX **Important:** • The minimum C++ runtime level requires the following filesets: – For AIX Version 5.3: xlC.rte 9.0.0.1 and xlC.aix50.rte 9.0.0.1 (or later). These filesets are included in the August 2007 IBM C++ Runtime Environment Components for AIX package. – For AIX Version 6.1: xlC.rte 9.0.0.1 and xlC.aix61.rte 9.0.0.1 (or later). These filesets are included in the October 2007 IBM C++ Runtime Environment Components for AIX package. • 32-bit applications developed using version 9 of the C++ compiler require a minimum of xlC.rte.9.0.0.4 and one of xlC.aix50.rte.9.0.0.4 (AIX 5.3) or xlC.aix61.rte.9.0.0.4 (AIX 6.1). These filesets are available in the December 2007 IBM C++ Runtime Environment Components for AIX package. • You can download the latest IBM C++ Runtime Environment Components for AIX from the IBM AIX XL C and C++ support web site.

Table 9. Supported C++ compilers (continued)

Operating System	**Supported Compiler(s)**
HP-UX on IPF (IA64)	• For HP-UX 11i v2, use HP aC++ Compiler Version A.06.05 • For HP-UX 11i v3, use HP aC++ Compiler Version A.06.12
Linux on x86 (32-bit DB2 instances and compilers only)	• GNU/Linux g++ versions 3.4 and 3.3 on RHEL4 and SLES9 • GNU/Linux g++ versions 4.1.1 and 4.1.0 on RHEL5 and SLES10 **Note:** If you experience problems with the g++ version 4.1.1 compiler on RHEL5, revert back to the g++ version 3.4 compiler which is available as a compat package on RHEL 5.
Linux on AMD64/EM64T (x86-64, x64)	• GNU/Linux g++ versions 3.4 and 3.3 on RHEL4 and SLES9 • GNU/Linux g++ versions 4.1.1 and 4.1.0 on RHEL5 and SLES10 **Note:** If you experience problems with the g++ version 3.4 compiler on RHEL 4, revert back to the g++ version 3.2 compiler which is available as a compat package on RHEL 4. **Note:** If you experience problems with the g++ version 4.1.1 compiler on RHEL5, revert back to the g++ version 3.4 compiler which is available as a compat package on RHEL 5.
Linux on POWER (PowerPC, iSeries, pSeries)	• GNU/Linux g++ versions 4.1.1 and 4.1.0 on RHEL5 and SLES10 • IBM XL C/C++ Enterprise Edition Version 7.0 for Linux on pSeries. **Note:** If you experience problems with the g++ version 4.1.1 compiler on RHEL 5, revert back to the g++ version 3.4 compiler which is available as a compat package on RHEL 5.
Linux on zSeries (s/390x)	• GNU/Linux g++ versions 3.4 and 3.3 on RHEL4 and SLES9 • GNU/Linux g++ versions 4.1.1 and 4.1.0 on RHEL5 and SLES10 **Note:** If you experience problems with the g++ version 3.4 compiler on RHEL 4, revert back to the g++ version 3.2 compiler which is available as a compat package on RHEL 4. **Note:** If you experience problems with the g++ version 4.1.1 compiler on RHEL5, revert back to the g++ version 3.4 compiler which is available as a compat package on RHEL 5.
Solaris on UltraSPARC	• Solaris Studio 10 • Forte C++ Version 6.2 • Sun ONE Studio 7, Compiler Collection

Table 9. Supported C++ compilers (continued)

Operating System	Supported Compiler(s)
Windows on x86 (32-bit DB2 instances only)	• Intel Proton Compiler for Windows 32-bit applications, Version 9.0.021 or later • Microsoft Visual C++ 2005 • Microsoft Visual C++ .NET
Windows on X64 (x86-64, AMD64/EM64T)	• Microsoft Platform SDK for Windows Server 2003 SP1 (Build 1830 or later), C/C++ compiler for x64 • Intel Proton Compiler for Windows x64, Version 9.0.024 or later • Microsoft Visual C++ 2005

Support for database application development in COBOL

The following table lists COBOL compilers that are supported for DB2 database application development.

Unless otherwise noted, the compilers listed in the following table support both 32-bit and 64-bit application development.

Table 10. Supported COBOL compilers

Operating System	Supported Compilers
AIX	• IBM COBOL Set for AIX Version 2.0 (32-bit only) • Micro Focus COBOL Server Express Version 4.0 with Service Pack 2 • Micro Focus Server Express, Version 5.0
HP-UX on IPF (IA64)	• Micro Focus COBOL Server Express Version 4.0 with Service Pack 2 • Micro Focus Server Express, Version 5.0 with Wrap Pack 4
Linux on x86 (32-bit DB2 instances and compilers only)	• Micro Focus COBOL Server Express Version 4.0 with Service Pack 2 on SLES9 • Micro Focus COBOL Server Express Version 4.0 with Service Pack 2 on RHEL4 • Micro Focus Server Express, Version 5.0 • Micro Focus does not offer support on SLES11
Linux on zSeries (s/390x)	• Micro Focus COBOL Server Express Version 4.0 with Service Pack 2 on SLES9 • Micro Focus Server Express, Version 5.0
Solaris on UltraSPARC	• Micro Focus COBOL Server Express Version 4.0 with Service Pack 2 (32-bit only) • Micro Focus Server Express, Version 5.0 (32-bit only)
Windows on x86 (32-bit DB2 instances only)	• Micro Focus COBOL Net Express Version 3.1.0, Version 4.0, and Version 5.0 • IBM VisualAge® COBOL Version 3.0.4 or later

Support for database application development in Fortran

The following table lists Fortran compilers that are supported for DB2 database application development.

Unless otherwise noted, the compilers listed in the following table support both 32-bit and 64-bit application development.

Table 11. Supported Fortran compilers

Operating System	Supported Compiler(s)
AIX	• IBM XL Fortran for AIX Versions 7.1.1.3 (32-bit only) • IBM XL Fortran for AIX Versions 8.1.1.0
HP-UX on IPF (IA64)	• HP-UX F90 B.11.23
Solaris on UltraSPARC	• SPARCompiler Fortran Versions 4.2 and 5.0 (32-bit only)

Java software support for DB2 products

You require the appropriate level of IBM Software Development Kit (SDK) for Java listed below to use Java-based tools and to create and run Java applications, including stored procedures and user-defined functions.

If the IBM SDK for Java is required by a component being installed and the SDK for Java is not already installed in that path, the SDK for Java will be installed if you use either the DB2 Setup wizard or a response file to install the product.

The SDK for Java is not installed with the IBM Data Server Runtime Client or the IBM Data Server Driver for ODBC, CLI, and .NET.

The following table lists the installed SDK for Java levels for DB2 products according to operating system platform:

Operating System Platform	SDK for Java level
AIX	SDK 5 Service Release 5
HP-UX for Itanium-based systems	HP SDK for J2SE HP-UX 11i platform, adapted by IBM for IBM Software, Version 5 Service Release 5
Linux on x86	SDK 5 Service Release 5
Linux on AMD64/EM64T	SDK 5 Service Release 5
Linux on zSeries	SDK 5 Service Release 5
Linux on POWER	SDK 5 Service Release 5
Solaris Operating System	SDK 5 Service Release 5
Windows x86	SDK 5 Service Release 5
Windows x64	SDK 5 Service Release 5

Note:

1. The SDK for Java software can be downloaded from the developerWorks® web page at: http://www.ibm.com/developerworks/java/jdk/index.html . For a list of the supported levels of the SDK for Java, see the table below entitled DB2 Database for Linux, UNIX, and Windows support for SDKs for Java.

Note: For Windows operating system platforms, use the IBM Development Package for Eclipse downloads.

2. DB2 GUI tools only run on Linux on x86, Linux on AMD64/EM64T, Windows x86, and Windows x64.
3. On Windows x86 and Linux on x86:
   - the 32-bit SDK is installed
   - 32-bit applications and Java external routines are supported
4. On all supported platforms (except Windows x86, and Linux on x86):
   - 32-bit applications are supported
   - 32-bit Java external routines are not supported
   - 64-bit applications and Java external routines are supported

Supported Java application development software

The following table lists the supported levels of the SDK for Java. The listed levels and forward-compatible later versions of the same levels are supported.

Because there are frequent SDK for Java fixes and updates, not all levels and versions have been tested. If your database application has problems that are related to the SDK for Java, try the next available version of your SDK for Java at the given level.

Non-IBM versions of the SDK for Java are supported only for building and running stand-alone Java applications. For building and running Java stored procedures and user-defined functions, only the IBM SDK for Java that is included with the DB2 Database for Linux, UNIX, and Windows product is supported.

Table 12. DB2 Database for Linux, UNIX, and Windows support for SDKs for Java

	Supported levels of the SDK for Java	
Operating system	**DB2 Version 9.5 and DB2 Version 9.5 Fix Pack 1**	**DB2 Version 9.5 Fix Pack 2 and Fix Pack 3**
AIX	1.4.2 to 5	1.4.2 to 6
HP-UX for Itanium-based systems	1.4.2 to 5[1]	1.4.2 to 6[1]
Linux on POWER	1.4.2 to 5[3]	1.4.2 to 6[3,4]
Linux on x86	1.4.2 to 5[2,3,5]	1.4.2 to 6[2,3,4]
Linux on AMD64 and Intel EM64T processors	1.4.2 to 5[2,3,5]	1.4.2 to 6[2,3,4]
Linux on zSeries	1.4.2 to 5[3]	1.4.2 to 6[3,4]
Solaris operating system	1.4.2 to 5 [2,5]	1.4.2 to 6[2]
Windows on x86	1.4.2 to 5[2,5]	1.4.2 to 6[2]
Windows on x64, for AMD64 and Intel EM64T processors	1.4.2 to 5[2,5]	1.4.2 to 6[2]

Table 12. DB2 Database for Linux, UNIX, and Windows support for SDKs for Java *(continued)*

	Supported levels of the SDK for Java	
Operating system	**DB2 Version 9.5 and DB2 Version 9.5 Fix Pack 1**	**DB2 Version 9.5 Fix Pack 2 and Fix Pack 3**

Note:

1. The same levels of the SDK for Java that are available from Hewlett-Packard are supported for building and running stand-alone client applications that run under the IBM Data Server Driver for JDBC and SQLJ.
2. The same levels of the SDK for Java that are available from Sun Microsystems are supported for building and running stand-alone client applications that run under the IBM Data Server Driver for JDBC and SQLJ.
3. A minimum level of SDK for Java 1.4.2 SR6 is required for SUSE Linux Enterprise Server (SLES) 10. A minimum level of SDK for Java 1.4.2 SR7 is required for Red Hat Enterprise Linux (RHEL) 5.
4. SDK for Java 6 support on Linux requires SDK for Java 6 SR3 or later.
5. The Sun Java Runtime Environment (JRE) 6 is supported, for running stand-alone Java applications.

The following table lists the versions of the IBM Data Server Driver for JDBC and SQLJ that are available with DB2 database products.

Table 13. Versions of IBM Data Server Driver for JDBC and SQLJ and DB2 Database for Linux, UNIX, and Windows fix pack levels

DB2 version and fix pack level	IBM Data Server Driver for JDBC and SQLJ version[1]
DB2 Version 9.1	3.1.*xx*
DB2 Version 9.1 Fix Pack 1	3.2.*xx*
DB2 Version 9.1 Fix Pack 2	3.3.*xx*
DB2 Version 9.1 Fix Pack 3	3.4.*xx*
DB2 Version 9.1 Fix Pack 4	3.6.*xx*
DB2 Version 9.1 Fix Pack 5	3.7.*xx*
DB2 Version 9.5	3.50.*xx*, 4.0.*xx*
DB2 Version 9.5 Fix Pack 1	3.51.*xx*, 4.1.*xx*
DB2 Version 9.5 Fix Pack 2	3.52.*xx*, 4.2.*xx*
DB2 Version 9.5 Fix Pack 3	3.53.*xx*, 4.3.*xx*

Note: All driver versions are of the form *n.m.xx*. *n.m* stays the same within a GA level or a fix pack level. *xx* changes when a new version of the IBM Data Server Driver for JDBC and SQLJ is introduced through an APAR fix.

Support for database application development in Perl

The following table lists versions of the Perl programming language that are supported for DB2 database application development.

Unless otherwise noted, the versions listed in the following table support both 32-bit and 64-bit application development.

Table 14. Supported versions of the Perl programming language

Operating System	Supported Version(s)
AIX	• Perl 5.8 or later, DBI 1.41 or later

Table 14. Supported versions of the Perl programming language (continued)

Operating System	Supported Version(s)
HP-UX on IPF (IA64)	• Perl 5.8 or later, DBI 1.41 or later
Linux on x86 (32-bit DB2 instances and compilers only)	• Perl 5.8 or later, DBI 1.41 or later
Linux on AMD64/EM64T (x86-64, x64)	• Perl 5.8 or later, DBI 1.41 or later
Linux on POWER (PowerPC, iSeries, pSeries)	• Perl 5.8 or later, DBI 1.41 or later
Linux on zSeries (s/390x)	• Perl 5.8 or later, DBI 1.41 or later
Solaris on UltraSPARC	• Perl 5.8 or later, DBI 1.41 or later
Windows on x86 (32-bit DB2instances only)	• Perl 5.8 or later, DBI 1.41 or later
Windows on X64 (x86-64, AMD64/EM64T)	• Perl 5.8 or later, DBI 1.41 or later

Support for database application development in PHP

The following table lists versions of the PHP scripting language that are supported for DB2 database application development.

Though the base support is listed in the table below, using the latest version of PHP 5 is recommended.

Unless otherwise noted, the version listed in the following table support both 32-bit and 64-bit application development.

Table 15. Supported versions of the PHP scripting language

Operating System	Supported Version(s)
AIX	• PHP 4.3.4 or later for IBM_DB2 driver • PHP 5.0 or later for PDO_IBM driver
HP-UX on IPF (IA64)	• PHP 4.3.4 or later for IBM_DB2 driver • PHP 5.0 or later for PDO_IBM driver
Linux on x86 (32-bit DB2 instances and language only)	• PHP 4.3.4 or later for IBM_DB2 driver • PHP 5.0 or later for PDO_IBM driver
Linux on AMD64/EM64T (x86-64, x64)	• PHP 4.3.4 or later for IBM_DB2 driver • PHP 5.0 or later for PDO_IBM driver
Linux on POWER (PowerPC, iSeries, pSeries)	• PHP 4.3.4 or later for IBM_DB2 driver • PHP 5.0 or later for PDO_IBM driver
Linux on zSeries (s/390x)	• PHP 4.3.4 or later for IBM_DB2 driver • PHP 5.0 or later for PDO_IBM driver
Solaris on UltraSPARC	• PHP 4.3.4 or later for IBM_DB2 driver • PHP 5.0 or later for PDO_IBM driver
Windows on x86 (32-bit DB2 instances only)	• PHP 4.3.4 or later for IBM_DB2 driver • PHP 5.0 or later for PDO_IBM driver

Table 15. Supported versions of the PHP scripting language (continued)

Operating System	Supported Version(s)
Windows on X64 (x86-64, AMD64/EM64T)	• PHP 4.3.4 or later for IBM_DB2 driver • PHP 5.0 or later for PDO_IBM driver

Support for database application development in Ruby/Ruby on Rails

The following table lists versions of Ruby/Ruby on Rails that are supported for DB2 database application development.

Unless otherwise noted, the version listed in the following table support both 32-bit and 64-bit application development.

Table 16. Supported versions for Ruby/Ruby on Rails

Operating System	Supported Version(s)
AIX	• Ruby 1.8.5 and later/Ruby on Rails 1.2.1 and later. Note: AIX 5.3 64 bit requires that the fix for APAR IZ01456 be applied.
HP-UX on IPF (IA64)	• Ruby 1.8.5 and later/Ruby on Rails 1.2.1 and later
Linux on x86 (32-bit DB2 instances and language only)	• Ruby 1.8.5 and later/Ruby on Rails 1.2.1 and later
Linux on AMD64/EM64T (x86-64, x64)	• Ruby 1.8.5 and later/Ruby on Rails 1.2.1 and later
Linux on POWER (PowerPC, iSeries, pSeries)	• Ruby 1.8.5 and later/Ruby on Rails 1.2.1 and later
Linux on zSeries (s/390x)	• Ruby 1.8.5 and later/Ruby on Rails 1.2.1 and later
Solaris on UltraSPARC	• Ruby 1.8.5 and later/Ruby on Rails 1.2.1 and later
Windows on x86 (32-bit DB2 instances only)	• Ruby 1.8.5 and later/Ruby on Rails 1.2.1 and later
Windows on X64 (x86-64, AMD64/EM64T)	• Ruby 1.8.5 and later/Ruby on Rails 1.2.1 and later

Support for database application development in REXX

The following table lists versions of the REXX programming language that are supported for DB2 database application development.

Only 32-bit versions of REXX are supported.

Table 17. Supported versions of the REXX programming language

Operating System	Supported Version(s)
AIX	• IBM AIX REXX/6000 AISPO® Product Number: 5764-057 • IBM Object REXX for AIX Version 1.1 • REXXSAA 4.00
Linux on x86 (32-bit DB2 instances and compilers only)	• Object REXX Interpreter for Linux Version 2.1

Table 17. Supported versions of the REXX programming language (continued)

Operating System	**Supported Version(s)**
Linux on zSeries (s/390x)	• Object REXX 2.2.0 for Linux/390

Chapter 7. Supported transaction managers

DB2database can serve as a resource manager for any XA-compliant transaction manager.

Here is a list of some common XA-compliant transaction managers that can be used to coordinate distributed transactions involving DB2databases:

- IBM TXSeries® CICS®
- IBM TxSeries Encina® Monitor
- IBMWebSphere MQ
- BEA Tuxedo
- BEA WebLogic
- Microsoft Distributed Transaction Coordinater (DTC)

Chapter 8. Supported database application development tools

The following integrated development environments and other development tools facilitate DB2 database application development:

IBM integrated database application development environments

- IBM Data Studio (complimentary IBM component)
- Rational® Application Developer (separately available IBM product)
- Rational Software Architect (separately available IBM product)
- Rational Data Architect (separately available IBM product)

Database application plug-ins for integrated development environments

- IBM Database Add-Ins for Visual Studio
- IBM Data Studio to extend Eclipse 3.2 environments

Tools for developing SQL, SQL/XML, and XQuery statements

- SQL Editor
- Command line processor (CLP)
- IBM Data Studio

Tools for database application monitoring and performance tuning

- Snapshot and Event Monitors
- Explain tools
- Static Profiling for CLI, ODBC, and JDBC applications
- IBM Data Studio (Visual Explain and SQL PL Profiling)

Tools for developing web applications

- IBM Data Studio - flexible web services data access (no programming required)
- Rational Application Developer - comprehensive Java development environment for web applications
- Zend Core for IBM - PHP development and deployment environment for web applications
- Ruby on Rails - rapid development of web applications using the Ruby on Rails framework and the new Rails Adapter to access data from IBM data servers
- Microsoft Visual Studio with IBM Database Add-Ins - development environment for .NET based web applications

IBM Database Enterprise Developer Edition

The IBM Database Enterprise Developer Edition product enables an application developer to design, build, and prototype applications for deployment on any of the IBM Information Management client or server platforms. Using the software that comes with this product, you can develop and test applications that run on one operating system and access databases on the same or on a different operating system. For example, you can create an application that runs on a Windows operating system but accesses a database on a UNIX operating system such as AIX.

For information on the software shipped with this product, see the IBM Web site: http://www.ibm.com/software/data/db2/9

Chapter 9. Configuring the database application development environment

Before you begin

To configure for DB2 database application development, you need (if necessary) :

- a supported operating system

  See: Supported operating systems for database application development.
- the IBM Data Server Client installed

  See: "Methods for installing IBM data server clients" in *Quick Beginnings for IBM Data Server Clients*.
- API driver(s) and, if required, driver manager(s) installed

  See: Supported database application programming interfaces.
- compiler(s) or interpreter(s) required for the programming language(s) you will be using installed

  See: Supported programming languages and compilers for database application development.
- a transaction manager installed

  See: Supported transaction managers.
- development tools installed

  See: Supported database application development tools.

About this task

The DB2 database application development environment is composed of several software elements:

- **operating system**
- **Data Server Client**
- **database application programming interface (API)**
- **programming language**
- **transaction manager**
- **development tools**

Each of these elements requires some configuration for DB2 database application development.

Procedure

To configure for DB2 database application development, do the following:

1. Configure the operating system.
2. Configure the Data Server Client.
3. Configure the development environment.
4. Configure the transaction manager.Configure the transaction manager. See "Configuration considerations for XA transaction managers" for details.

Configuring the operating system for database application development

Before you begin

To configure environment variables for DB2 database application development, you must have:

- a supported operating system

  See: Supported operating systems for database application development.
- a IBM Data Server Clientinstalled

  See: "Methods for installing IBM data server clients" in *Quick Beginnings for IBM Data Server Clients*.

About this task

Part of configuring for DB2 database application development is configuring operating system local and global environment variables.

Procedure

To configure environment variables for DB2 database application development:

Perform one of the following tasks:

- For UNIX operating systems, Set up the UNIX application development environment.
- For Windows operating systems, Set up the Windows application development environment

Setting up the UNIX application development environment

You need to set environment variables for your database instance. Each database manager instance has two files, `db2profile` and `db2cshrc`, which are scripts to set the environment variables for that instance. Note that these settings are not required when using T4 JDBC drivers.

1. Run the correct script for the shell you are using For bash or Korn shell:

   ```
   . $HOME/sqllib/db2profile
   ```

   For C shell:

   ```
   source $HOME/sqllib/db2cshrc
   ```

   Where $HOME is the home directory of the instance owner. If you include this command in your .profile or .login file, the command runs automatically when you log on.
2. If you will be using ODBC, DB2 CLI, or Java, do the steps in one of the following topics:
   - Setting up the UNIX ODBC environment .
   - Installing the IBM Data Server Driver for JDBC and SQLJ.

UNIX environment variable settings

Depending on the UNIX platform you are on, values for the following environment variables are set, either in `db2profile` (for bash or korn shell) or `db2cshrc` (for C shell), and a call to these files are put in the instance owner's `.profile` (bash or korn shell) or `.login` (C shell) file.

AIX:

- PATH, includes several DB2 directories including `sqllib/bin`
- LIBPATH, includes the directory `sqllib/lib` (see note below)

HP-UX:

- PATH, includes several DB2 directories including `sqllib/bin`
- SHLIB_PATH (32-bit and 64-bit) or LD_LIBRARY_PATH (64-bit), includes the directory `sqllib/lib` (see note below)

Linux and Solaris:

- PATH, includes several DB2 directories including `sqllib/bin`
- LD_LIBRARY_PATH, includes the directory `sqllib/lib` (see note below)

The blank files `sqllib/userprofile` and `sqllib/usercshrc` are created during instance creation to allow users to place their own instance environmental settings. If you do not want the new environment settings in the `db2profile` or `db2cshrc` scripts, you can override them using the corresponding ″user″ script, which is called at the end of the `db2profile` or `db2cshrc` script. During an instance migration (`db2imigr`), the user scripts are copied over so that your environment modifications will still be in use.

Setting up the Windows application development environment

When you install the DB2 Client on Windows operating systems, the install program updates the configuration registry with the environment variables INCLUDE, LIB, and PATH. The system-wide environment variable, DB2INSTANCE, is set by install to the default instance created, called DB2. DB2PATH is set inside a DB2 command window when the window is opened.

You can override these environment variables to set the values for the machine or the currently logged-on user. Exercise caution when changing these environment variables. Do not change the DB2PATH environment variable. DB2INSTANCE is defined as a system-level environment variable. You do not need to make use of the DB2INSTDEF DB2 registry variable which defines the default instance name to use if DB2INSTANCE is not set.

1. To override the environment variable settings, use the Windows Control Panel. When using the variable `%DB2PATH%` in a command, put the full path in quotes, as in:

   ```
   set LIB="%DB2PATH%;%LIB%
   ```

   The default installation value for this variable is `\Program Files\IBM\SQLLIB`, which contains a space, so put the full path in quotes.
2. In addition, you must take the following specific steps for running DB2 applications:
   - When building C or C++ programs, you must ensure that the `INCLUDE` environment variable contains `%DB2PATH%\INCLUDE` as the first directory.

     To do this, update the environment for your compiler:

     a. Launch the shortcut to the Visual Studio Command Prompt.

     b. In the Visual Studio Command window, run DB2CMD.exe to open the DB2 Command Window.

     c. In the DB2 command window, set your INCLUDE path as follows :

     ```
     set INCLUDE=%DB2PATH%\INCLUDE;%INCLUDE%
     ```
   - When building Micro Focus COBOL programs, set the `COBCPY` environment variable to point to `%DB2PATH%\INCLUDE\cobol_mf`.

- When building IBM COBOL programs, set the `SYSLIB` environment variable to point to `%DB2PATH%\INCLUDE\cobol_a`.
- Ensure the LIB environment variable points to `%DB2PATH%\lib` by using:

  ```
  set LIB="%DB2PATH%\lib";%LIB%
  ```

  To enable cross-developing 64-bit applications from a 32-bit environment, see Migrating 32-bit database applications to run on 64-bit instances
- To run an application against a remote server, both the client and the server have to be properly configured. For instance, the `DB2COMM` environment variable should be set at the server of the remote database.

Configuring the IBM Data Server Client for database application development

Before you begin

To configure the Data Server Client for database application development, you must have:

- installed a Data Server Client

  See: "Installing IBM data server clients (Linux and UNIX)" in *Quick Beginnings for IBM Data Server Clients* and "Installing IBM data server clients (Windows)" in *Quick Beginnings for IBM Data Server Clients*.
- optional: created sample databases with which to test you applications

  See: The SAMPLE database.
- completed basic configuration steps for your Data Server Client

  See: Client-to-server communications configuration overview.

Procedure

To configure the Data Server Client for database application development, do the following:

The Data Server Client supports database application development; part of configuring for application development is configuring the client.

Configuring the development environment for application programming interfaces

Before you begin

To configure the application development environment for the application programming interfaces (APIs) that you will use, you must have:

- the IBM Data Server Client installed

  See: "Methods for installing IBM data server clients" in *Quick Beginnings for IBM Data Server Clients*.
- the API driver(s) installed

  See: Chapter 5, "Supported database application programming interfaces," on page 17.
- for the ODBC API only: an ODBC driver manager installed

  See: "Setting up the Windows CLI environment" in *Call Level Interface Guide and Reference, Volume 1*.

Part of configuring for DB2 database application development is configuring the development environment for any application programming interfaces (APIs) that you will use in your applications.

Procedure

To configure the development environment for APIs, do the following:

1. For ADO .NET, see “ADO.NET development for IBM Data Servers” in *Developing ADO.NET and OLE DB Applications* for details.
2. For DB2 CLI and ODBC, perform any of the following tasks that apply to your environment:
   - “Setting up your application development environment to run DB2 CLI and ODBC applications” in *Call Level Interface Guide and Reference, Volume 1*.
   - Setting up the UNIX ODBC Driver Manager“Setting up the UNIX ODBC Driver Manager” in *Call Level Interface Guide and Reference, Volume 1*.
3. For Embedded SQL, perform the following task: “Setting up the embedded SQL development environment” in *Developing Embedded SQL Applications*.
4. For JDBC and SQLJ, perform the following task: “Installing the IBM Data Server Driver for JDBC and SQLJ” in *Developing Java Applications*.
5. For OLE DB, see “IBM OLE DB Provider for DB2” in *Getting Started with Database Application Development* for details.
6. For Perl, “Perl DBI” in *Developing Perl and PHP Applications* for details.
7. For PHP, perform one of the following tasks:
   - “Setting up the PHP environment on Linux or UNIX” in *Developing Perl and PHP Applications*.
   - “Setting up the PHP environment on Windows” in *Developing Perl and PHP Applications*.
8. For Ruby/Ruby on Rails, perform any of the tasks that are listed in “The IBM_DB Ruby driver and Rails adapter” in *Getting Started with Database Application Development* and that apply to your environment.

Setting up your application development environment to run DB2 CLI and ODBC applications

You can run DB2 CLI and ODBC applications against a DB2 database server using the IBM Data Server Client, the IBM Data Server Runtime Client, or the IBM Data Server Driver for ODBC and CLI. However, to compile DB2 CLI or ODBC applications, you need the IBM Data Server Client.

Before you set up your CLI environment, ensure you have set up the application development environment.

In order for a DB2 CLI application to successfully access a DB2 database:

1. Ensure the DB2 CLI/ODBC driver was installed during the DB2 client install.
2. For the IBM Data Server Client and Runtime Client only: If the database is being accessed from a remote client, catalog the database and hostname of the machine the databse is on.

   On the Windows platform, you can use the CLI/ODBC Settings GUI to catalog the DB2 database.
3. Optional: Explicitly bind the DB2 CLI/ODBC bind files to the database with the command:

```
db2 bind ~/sqllib/bnd/@db2cli.lst blocking all sqlerror continue \
         messages cli.msg grant public
```

On the Windows platform, you can use the CLI/ODBC Settings GUI to bind the DB2 CLI/ODBC bind files to the database.

4. Optional: Change the DB2 CLI/ODBC configuration keywords by editing the `db2cli.ini file`, located in the `sqllib` directory on Windows, and in the `sqllib/cfg` directory on UNIX platforms.

   On the Windows platform, you can use the CLI/ODBC Settings GUI to set the DB2 CLI/ODBC configuration keywords.

Once you have completed the above steps, proceed to setting up your Windows CLI environment, or setting up your UNIX ODBC environment if you are running ODBC applications on UNIX.

Installing the IBM Data Server Driver for JDBC and SQLJ

After you install the IBM Data Server Driver for JDBC and SQLJ, you can prepare and run JDBC or SQLJ applications.

Before you install the IBM Data Server Driver for JDBC and SQLJ, you need the following software.

- An SDK for Java, 1.4.2 or later.

  For all DB2 products except the IBM Data Server Runtime Client, the DB2 Database for Linux, UNIX, and Windows installation process automatically installs the SDK for Java, Version 5.

  If you want to use JDBC 4.0 functions, you need to install an SDK for Java, 6 or later.

  If you plan to run JDBC or SQLJ applications on your system, but not to prepare them, you need a Java run-time environment only.
- JVM native threads support

  Any JVMs that run Java applications that access DB2 databases must include native threads support. You can specify native threads as the default thread support for some JVMs by setting the THREADS_FLAG environment variable to ″native″. Refer to the documentation for your Java environment for instructions on making native threads the default on your system.
- Unicode support for System i servers

  If any SQLJ or JDBC programs will use IBM Data Server Driver for JDBC and SQLJ type 4 connectivity to connect to a DB2 for i5/OS server, the OS/400® operating system must support the Unicode UTF-8 encoding scheme. The following table lists the OS/400 PTFs that you need for Unicode UTF-8 support:

Table 18. OS/400 PTFs for Unicode UTF-8 support

OS/400 version	PTF numbers
V5R3 or later	None (support is included)
V5R2	SI06541, SI06796, SI07557, SI07564, SI07565, SI07566, and SI07567
V5R1	SI06308, SI06300, SI06301, SI06302, SI06305, SI06307, and SI05872

- Java support for HP-UX clients and servers

  HP-UX servers: The IBM Data Server Driver for JDBC and SQLJ does not support databases that are in the HP-UX default character set, Roman8. Therefore, when you create a database on an HP-UX server that you plan to access with the IBM Data Server Driver for JDBC and SQLJ, you need to create the database with a different character set.

HP-UX clients and servers: The Java environment on an HP-UX system requires special setup to run stored procedures under the IBM Data Server Driver for JDBC and SQLJ.

Follow these steps to install the IBM Data Server Driver for JDBC and SQLJ.

1. During the DB2 Database for Linux, UNIX, and Windows installation process, select `Java support` on UNIX or Linux, or `JDBC support` on Windows. These selections are defaults. If you have already installed DB2 Database for Linux, UNIX, and Windows without JDBC support, you can run the installation process in `Custom` mode to add JDBC support.

   Selection of `Java support` or `JDBC support` causes the installation process to perform the following actions:

   - Installs the IBM Data Server Driver for JDBC and SQLJ class files.

     The files are placed in the sqllib\java directory for Windows systems, or the sqllib/java directory for UNIX or Linux systems.

     The files names are:

     db2jcc.jar or db2jcc4.jar

     Include db2jcc.jar in the CLASSPATH if you plan to use the version of the IBM Data Server Driver for JDBC and SQLJ that includes only **JDBC 3.0 and earlier functions.**

     Include db2jcc4.jar in the CLASSPATH if you plan to use the version of the IBM Data Server Driver for JDBC and SQLJ that includes **JDBC 4.0 and later functions, as well as JDBC 3.0 and earlier functions.**

     sqlj.zip or sqlj4.zip

     Include sqlj.zip in the CLASSPATH if you plan to prepare SQLJ applications that include only **JDBC 3.0 and earlier functions.**

     Include sqlj4.zip in the CLASSPATH if you plan to prepare SQLJ applications that include **JDBC 4.0 and later functions, as well as JDBC 3.0 and earlier functions.**

   - Modifies the CLASSPATH to include the IBM Data Server Driver for JDBC and SQLJ class files.

     Important: This step is performed automatically only for the db2jcc.jar and sqlj.zip file. If you are using the db2jcc4.jar file or the sqlj4.zip file, you must modify the CLASSPATH manually. Change db2jcc.jar to db2jcc4.jar or sqlj.zip to sqlj4.zip in the CLASSPATH.

     Important: Include db2jcc.jar or db2jcc4.jar in the CLASSPATH. Do not include both files.

     Important: Include sqlj.zip or sqlj4.zip in the CLASSPATH. Do not include both files.

   - Installs IBM Data Server Driver for JDBC and SQLJ license files, and modifies the CLASSPATH to include them.

     The files are placed in the sqllib\java directory for Windows systems, or the sqllib/java directory for UNIX or Linux systems. The file names are:

Table 19. IBM Data Server Driver for JDBC and SQLJ license files

License file	Server to which license file permits a connection	Product that includes license file
db2jcc_license_cisuz.jar	DB2 for z/OS DB2 for i5/OS	All DB2 Connect products

License files are not required for connections to DB2 Database for Linux, UNIX, and Windows, Cloudscape®, or IBM Informix Dynamic Server (IDS) databases from the IBM Data Server Driver for JDBC and SQLJ version 3.50 or later.

- Installs IBM Data Server Driver for JDBC and SQLJ native libraries for support of IBM Data Server Driver for JDBC and SQLJ type 2 connectivity.

  The files are placed in the sqllib\bin directory for Windows systems, or the sqllib/lib directory for UNIX or Linux systems.

  The file names are:

  libdb2jcct2.so
  For AIX, HP-UX on IPF, Linux, and Solaris

  libdb2jcct2.sl
  For HP-UX on PA-RISC

  db2jcct2.dll
  For Windows

2. Customize the driver-wide configuration properties, if any of the defaults are inappropriate.
3. Configure TCP/IP.

   Servers must be configured for TCP/IP communication in the following cases:

   - JDBC or SQLJ applications that use IBM Data Server Driver for JDBC and SQLJ type 4 connectivity.
   - JDBC or SQLJ applications that use IBM Data Server Driver for JDBC and SQLJ type 2 connectivity, and specify *server* and *port* in the connection URL.

   Ensure that the TCP/IP listener is running. To activate the TCP/IP listener:

   a. Set the environment variable DB2COMM to TCPIP:

   ```
   db2set DB2COMM=TCPIP
   ```

   b. Update the database manager configuration file with the TCP/IP service name as specified in the services file:

   ```
   db2 update dbm cfg using SVCENAME TCP/IP-service-name
   ```

   The port number used for applets and SQLJ programs needs to be the same as the TCP/IP SVCENAME number used in the database manager configuration file.

   c. Execute the db2stop and db2start commands for the service name setting to take effect.
4. On DB2 Database for Linux, UNIX, and Windows servers on which you plan to run Java stored procedures or user-defined functions, ensure that the DB2_USE_DB2JCCT2_JROUTINE environment variable is not set, or is set to its default value of `YES`, `yes`, `ON`, `on`, `TRUE`, `true`, or 1 on those database servers. This setting indicates that Java stored procedures run under the IBM Data Server Driver for JDBC and SQLJ.

   If you need to run stored procedures under the DB2 JDBC Type 2 Driver for Linux, UNIX and Windows, set the DB2_USE_DB2JCCT2_JROUTINE environment variable to `OFF`.

5. On DB2 Database for Linux, UNIX, and Windows servers on which you plan to run Java stored procedures or user-defined functions, update the database manager configuration to include the path where the SDK for Java is located.

   You can do this by entering commands similar to these on the server command line:

   - *For database systems on UNIX or Linux:*

     ```
     db2 update dbm cfg using JDK_PATH /home/db2inst/jdk15
     ```

     /home/db2inst/jdk15 is the path where the SDK for Java is installed.
   - *For database systems on Windows:*

     ```
     db2 update dbm cfg using JDK_PATH c:\Program Files\jdk15
     ```

     c:\Program Files\jdk15 is the path where the SDK for Java is installed.

   To verify the correct value for the `JDK_PATH` field in the DB2 database manager configuration, enter the following command on the database server:

   ```
   db2 get dbm cfg
   ```

   You might want to redirect the output to a file for easier viewing. The `JDK_PATH` field appears near the beginning of the output.
6. If you plan to call SQL procedures that are on DB2 Database for Linux, UNIX, and Windows servers from Java programs, and the date and time format that is associated with the territory code of the database servers is **not** the USA format, take the following actions:
   a. Set the DB2_SQLROUTINE_PREPOPTS registry variable on the database servers to indicate that the default datetime format is ISO:

      ```
      db2set DB2_SQLROUTINE_PREPOPTS="DATETIME ISO"
      ```
   b. Redefine any existing SQL procedures that you plan to call from Java programs.

   These steps are necessary to ensure that the calling application receives date and time values correctly.
7. If you plan to access DB2 for z/OS database servers with your Java applications, follow the instructions in Special setup for accessing DB2 for z/OS servers from Java programs.To connect to DB2 on zOS using Type 4 JCC driver, you need the `db2jcc_license_cisuz.jar` license. Make sure you have a valid licensed DB2 Connect product and add the license file to your classpath. The file is usually located under `/SQLLIB/java/db2jcc_license_cisuz.jar`.

Transaction managers

Transaction managers are software components used to assign identifiers to specific database transactions, monitor transaction processing, and to validate or act when a transaction completes or fails. Transaction management guards against hardware and software errors that might leave a transaction partially completed, with a database left in an unknown, inconsistent state. The transaction manager ensures that all operations in any uncommitted (i.e., not completely processed) transactions are cancelled.

The DB2 Database for Linux, UNIX, and Windows transaction manager (TM), provided with DB2 and DB2 Connect, performs all of these functions. The database manager provides transaction manager functions used to coordinate the updating of several databases within a single unit of work. The database client automatically coordinates the unit of work, and uses a transaction manager database to register each transaction and track its completion status. You can use the DB2 transaction manager with DB2 databases.

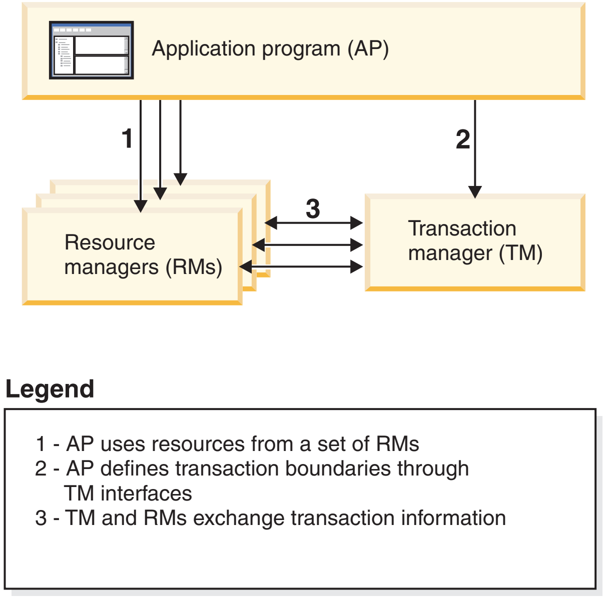
If you have resources other than DB2 databases that you want to participate in a two-phase commit transaction, you can use an XA-compliant transaction manager.

X/Open distributed transaction processing model

The X/Open Distributed Transaction Processing (DTP) model includes three interrelated components:

- Application program (AP)
- Transaction manager (TM)
- Resources managers (RM)

Figure 1 illustrates this model, and shows the relationship among these components.



1 - AP uses resources from a set of RMs
2 - AP defines transaction boundaries through TM interfaces
3 - TM and RMs exchange transaction information

Figure 1. X/Open distributed transaction processing (DTP) model

Application program (AP)

The application program (AP) defines transaction boundaries, and defines the application-specific actions that make up the transaction.

For example, a CICS application program might want to access resource managers (RMs), such as a database and a CICS Transient Data Queue, and use programming logic to manipulate the data. Each access request is passed to the appropriate resource managers through function calls specific to that RM. In the case of DB2 products, these could be function calls generated by the DB2 database precompiler for each SQL statement, or database calls coded directly by the programmer using the APIs.

A transaction manager (TM) product usually includes a transaction processing (TP) monitor to run the user application. The TP monitor provides APIs to allow an application to start and end a transaction, and to perform application scheduling and load balancing among the many users who want to run the application. The application program in a distributed transaction processing (DTP) environment is really a combination of the user application and the TP monitor.

To facilitate an efficient online transaction processing (OLTP) environment, the TP monitor pre-allocates a number of server processes at startup, and then schedules and reuses them among the many user transactions. This conserves system resources, by allowing more concurrent users to be supported with a smaller number of server processes and their corresponding RM processes. Reusing these processes also avoids the overhead of starting up a process in the TM and RMs for each user transaction or program. (A program invokes one or more transactions.) This also means that the server processes are the actual ″user processes″ to the TM and the RMs. This has implications for security administration and application programming.

The following types of transactions are possible from a TP monitor:

- Non-XA transactions

  These transactions involve RMs that are not defined to the TM, and are therefore not coordinated under the two-phase commit protocol of the TM. This might be necessary if the application needs to access an RM that does not support the XA interface. The TP monitor simply provides efficient scheduling of applications and load balancing. Since the TM does not explicitly ″open″ the RM for XA processing, the RM treats this application as any other application that runs in a non-DTP environment.
- Global transactions

  These transactions involve RMs that are defined to the TM, and are under the TM's two-phase commit control. A global transaction is a unit of work that could involve one or more RMs. A *transaction branch* is the part of work between a TM and an RM that supports the global transaction. A global transaction could have multiple transaction branches when multiple RMs are accessed through one or more application processes that are coordinated by the TM.

  Loosely coupled global transactions exist when each of a number of application processes accesses the RMs as if they are in a separate global transaction, but those applications are under the coordination of the TM. Each application process will have its own transaction branch within an RM. When a commit or rollback is requested by any one of the APs, TM, or RMs, the transaction branches are completed altogether. It is the application's responsibility to ensure that resource deadlock does not occur among the branches. (Note that the transaction coordination performed by the DB2 transaction manager for applications prepared with the SYNCPOINT(TWOPHASE) option is roughly equivalent to these loosely coupled global transactions.

  Tightly coupled global transactions exist when multiple application processes take turns to do work under the same transaction branch in an RM. To the RM, the two application processes are a single entity. The RM must ensure that resource deadlock does not occur within the transaction branch.

Transaction manager (TM)

The transaction manager (TM) assigns identifiers to transactions, monitors their progress, and takes responsibility for transaction completion and failure. The transaction branch identifiers (known as XIDs) are assigned by the TM to identify both the global transaction, and the specific branch within an RM. This is the correlation token between the log in a TM and the log in an RM. The XID is needed for two-phase commit, or rollback, to perform the *resynchronization* operation (also known as a *resync*) on system startup, or to let the administrator perform a *heuristic* operation (also known as *manual intervention*), if necessary.

After a TP monitor is started, it asks the TM to open all the RMs that a set of application servers have defined. The TM passes xa_open calls to the RMs, so that they can be initialized for DTP processing. As part of this startup procedure, the TM performs a resync to recover all *indoubt transactions*. An indoubt transaction is a global transaction that was left in an uncertain state. This occurs when the TM (or at least one RM) becomes unavailable after successfully completing the first phase (that is, the prepare phase) of the two-phase commit protocol. The RM will not know whether to commit or roll back its branch of the transaction until the TM can reconcile its own log with the RM logs when they become available again. To perform the resync operation, the TM issues a xa_recover call one or more times to each of the RMs to identify all the indoubt transactions. The TM compares the replies with the information in its own log to determine whether it should inform the RMs to xa_commit or xa_rollback those transactions. If an RM has already committed or rolled back its branch of an indoubt transaction through a heuristic operation by its administrator, the TM issues an xa_forget call to that RM to complete the resync operation.

When a user application requests a commit or a rollback, it must use the API provided by the TP monitor or TM, so that the TM can coordinate the commit and rollback among all the RMs involved. For example, when a CICS application issues the CICS SYNCPOINT request to commit a transaction, the CICS XA TM (implemented in the Encina Server) will in turn issue XA calls, such as xa_end, xa_prepare, xa_commit, or xa_rollback to request the RM to commit or roll back the transaction. The TM could choose to use one-phase instead of two-phase commit if only one RM is involved, or if an RM replies that its branch is read-only.

Resource managers (RM)

A resource manager (RM) provides access to shared resources, such as databases.

The DB2 system, as resource manager of a database, can participate in a *global transaction* that is being coordinated by an XA-compliant TM. As required by the XA interface, the database manager provides a *db2xa_switch* external C variable of type xa_switch_t to return the XA switch structure to the TM. This data structure contains the addresses of the various XA routines to be invoked by the TM, and the operating characteristics of the RM.

There are two methods by which the RM can register its participation in each global transaction: *static registration* and *dynamic registration*:

- Static registration requires the TM to issue (for every transaction) the xa_start, xa_end, and xa_prepare series of calls to all the RMs defined for the server application, regardless of whether a given RM is used by the transaction. This is inefficient if not every RM is involved in every transaction, and the degree of inefficiency is proportional to the number of defined RMs.
- Dynamic registration (used by DB2) is flexible and efficient. An RM registers with the TM using an ax_reg call only when the RM receives a request for its resource. Note that there is no performance disadvantage with this method, even when there is only one RM defined, or when every RM is used by every transaction, because the ax_reg and the xa_start calls have similar paths in the TM.

The XA interface provides two-way communication between a TM and an RM. It is a system-level interface between the two DTP software components, not an ordinary application program interface to which an application developer codes. However, application developers should be familiar with the programming restrictions that the DTP software components impose.

Although the XA interface is invariant, each XA-compliant TM may have product-specific ways of integrating an RM. For information about integrating your DB2 product as a resource manager with a specific transaction manager, see the appropriate TM product documentation.

xa_open string formats

xa_open string format for DB2 Database for Linux, UNIX, and Windows and DB2 Connect Version 8 FixPak 3 and later

This is the format for the xa_open string:

```
parm_id1 = <parm value>,parm_id2 = <parm value>, ...
```

It does not matter in what order these parameters are specified. Valid values for *parm_id* are described below.

Note: Unless explicitly stated, these parameters are not case sensitive and have no default value.

AXLIB
Library that contains the TP monitor's ax_reg and ax_unreg functions. This value is used by DB2 to obtain the addresses of the required ax_reg and ax_unreg functions. It can be used to override assumed values based on the TPM parameter, or it can be used by TP monitors that do not appear on the list for TPM. On AIX, if the library is an archive library, the archive member should be specified in addition to the library name. For example: `AXLIB=/usr/mqm/lib/libmqmax_r.a(libmqmax_r.o)`. This parameter is optional.

CHAIN_END
xa_end chaining flag. Valid values are `T`, `F`, or no value. XA_END chaining is an optimization that can be used by DB2 to reduce network flows. If the TP monitor environment is such that it can be guaranteed that xa_prepare will be invoked within the same thread or process immediately following the call to xa_end, and if CHAIN_END is on, the xa_end flag will be chained with the xa_prepare command, thus eliminating one network flow. A value of `T` means that CHAIN_END is on; a value of `F` means that CHAIN_END is off; no specified value means that CHAIN_END is on. This parameter can be used to override the setting derived from a specified TPM value. If this parameter is not specified, the default value of `F` is used.

CREG
xa_start chaining flag. Valid values are `T`, or `F`, or no value.xa_start chaining is an optimization that is used by DB2 to reduce network flows. The parameter is only valid if the TP monitor is using static registration (see SREG). The TP monitor environment is such that it can guarantee that an SQL statement will be invoked immediately after the call to the XA API xa_start. If CREG is set to `T`, the SQL statement is chained to the xa_start request, thus eliminating one network flow. This parameter can be used to override the setting derived from a specified TPM value. If this parameter is not specified, the default value of `F` is used.

CT
Connect Timeout. Valid values are `0` - `32767`. CT specifies the amount of time,

in seconds, that an application will wait when attempting to establish a connection with the server. If a connection is not established in the amount of time specified, an error will be returned. Specifying a value of 0 means that the application will attempt to wait until a connection is established regardless of how long it takes. However, it is possible that the connection attempt will be terminated by the default TCP/IP timeout setting. If this parameter is not specified, the default value of 0 is used.

DB

Database alias. Database alias used by the application to access the database. This parameter must be specified.

HOLD_CURSOR

Specifies whether cursors are held across transaction commits. Valid values are T, F, or no value. TP monitors typically reuse threads or processes for multiple applications. To ensure that a newly loaded application does not inherit cursors opened by a previous application, cursors are closed after a commit. If HOLD_CURSORS is on, cursors with hold attributes are not closed, and will persist across transaction commit boundaries. When using this option, the global transaction must be committed or rolled back from the same thread of control. If HOLD_CURSOR is off, the opening of any cursors with hold attributes will be rejected. A value of T means that HOLD_CURSOR is on; a value of F means that HOLD_CURSOR is off; no specified value means that HOLD_CURSOR is on. This parameter can be used to override the setting derived from a specified TPM value. If this parameter is not specified, the default value of F is used.

PWD

Password. A password that is associated with the user ID. Required if a user ID is specified. This parameter is case sensitive.

SINGLE_PROCESS

Specifies whether direct XA support is enabled for clients accessing DB2 for z/OS. Valid values are T, F, or no value. For instance-based clients (IBM data server clients), XA support must be explicitly enabled either through this setting or the enableDirectXA setting in the db2dsdriver configuration file. For instance-less clients (IBM data server drivers), XA support is enabled by default for Microsoft Distributed Transaction Coordinator or Microsoft Component Services (COM+). For all other supported transaction managers, XA support for instance-less clients must be explicitly enabled through the SINGLE_PROCESS keyword.

SREG

Static Registration. Valid values are T, or F, or no value.DB2 supports two methods of registering a global transaction. The first is Dynamic Registeration, where DB2 calls the TP's ax_reg function to register the transaction (see AXLIB). The second method is Static Registeration, where the TP calls the XA API xa_start to initiate a global transaction. Please note both dynamic and static registration are mutually exclusive. If this parameter is not specified, the default value of F is used.

SUSPEND_CURSOR

Specifies whether cursors are to be kept when a transaction thread of control is suspended. Valid values are T, F, or no value. TP monitors that suspend a transaction branch can reuse the suspended thread or process for other transactions. If SUSPEND_CURSOR is off, all cursors except cursors with hold attributes are closed. On resumption of the suspended transaction, the application must obtain the cursors again. If SUSPEND_CURSOR is on, any open cursors are not closed, and are available to the suspended transaction on resumption. A value of T means that SUSPEND_CURSOR is on; a value of F means that SUSPEND_CURSOR is off; no specified value means that SUSPEND_CURSOR is on. This parameter can be used to override the setting derived from a specified TPM value. If this parameter is not specified, the default value of F is used.

TOC

The entity (“Thread of Control”) to which all DB2 XA Connections are bound. Valid values are T, or P, or not set. TOC is the entity where all DB2 XA Connections are bound. All DB2 XA Connections formed within an entity must be unique. That is, they cannot have two connections to the same database within the entity. The TOC has two parameters: T (OS Thread) and P (OS Process). When set to a value of T, all DB2 XA Connections formed under a particular OS Thread are unique to that thread only. Multiple threads cannot share DB2 XA Connections. Each OS thread has to form its own set of DB2 XA Connections. When set to a value of P, all DB2 XA Connections are unique to the OS Process and all XA Connections can be shared between OS threads. If this parameter is not specified, the default value of T is used.

TPM

Transaction processing monitor name. Name of the TP monitor being used. For supported values, see the next table. This parameter can be specified to allow multiple TP monitors to use a single DB2 instance. The specified value will override the value specified in the *tp_mon_name* database manager configuration parameter. This parameter is optional.

UID

User ID. Specifies the user ID that has authority to connect to the database. Required if a password is specified. This parameter is case sensitive.

UREGNM

User Registry Name. When an identity mapping service is being used, this parameter gives the name of the registry to which the user name given in the UID parameter belongs.

TCTX

Specifies whether or not the transaction should use a trusted connection. Valid values are TRUE or FALSE. If this parameter is set to TRUE it tells the transaction manager to try to open a trusted connection.

TPM and tp_mon_name values

The xa_open string TPM parameter and the *tp_mon_name* database manager configuration parameter are used to indicate to DB2 which TP monitor is being used. The *tp_mon_name* value applies to the entire DB2 instance. The TPM parameter applies only to the specific XA resource manager. The TPM value overrides the *tp_mon_name* parameter. Valid values for the TPM and *tp_mon_name* parameters are as follows:

Table 20. Valid Values for TPM and tp_mon_name

TPM Value	TP Monitor Product	Internal Settings

CICS	IBM TxSeries CICS	AXLIB=libEncServer (for Windows)
		=/usr/lpp/encina/lib/libEncServer
		(for UNIX based systems)
		HOLD_CURSOR=T
		CHAIN_END=T
		SUSPEND_CURSOR=F
		TOC=T

Table 20. Valid Values for TPM and tp_mon_name (continued)

TPM Value	TP Monitor Product	Internal Settings

ENCINA	IBM TxSeries Encina Monitor	AXLIB=libEncServer (for Windows)
		=/usr/lpp/encina/lib/libEncServer
		(for UNIX based systems)
		HOLD_CURSOR=F
		CHAIN_END=T
		SUSPEND_CURSOR=F
		TOC=T

MQ	IBM MQSeries®	AXLIB=mqmax (for Windows) =/usr/mqm/lib/libmqmax_r.a (for AIX threaded applications) =/usr/mqm/lib/libmqmax.a (for AIX non-threaded applications) =/opt/mqm/lib/libmqmax.so (for Solaris) =/opt/mqm/lib/libmqmax_r.sl (for HP threaded applications) =/opt/mqm/lib/libmqmax.sl (for HP non-threaded applications) =/opt/mqm/lib/libmqmax_r.so (for Linux threaded applications) =/opt/mqm/lib/libmqmax.so (for Linux non-threaded applications) HOLD_CURSOR=F CHAIN_END=F SUSPEND_CURSOR=F TOC=P

CB	IBM Component Broker	AXLIB=somtrx1i (for Windows)
		=libsomtrx1
		(for UNIX based systems)
		HOLD_CURSOR=F
		CHAIN_END=T
		SUSPEND_CURSOR=F
		TOC=T

SF	IBM San Francisco	AXLIB=ibmsfDB2
		HOLD_CURSOR=F
		CHAIN_END=T
		SUSPEND_CURSOR=F
		TOC=T

TUXEDO	BEA Tuxedo	AXLIB=libtux
		HOLD_CURSOR=F
		CHAIN_END=F
		SUSPEND_CURSOR=F
		TOC=T

MTS	Microsoft Transaction Server	It is not necessary to configure DB2 for MTS. MTS is automatically detected by DB2's ODBC driver.
JTA	Java Transaction API	It is not necessary to configure DB2 for Enterprise Java Servers (EJS) such as IBM WebSphere. DB2's JDBC driver automatically detects this environment. Therefore this TPM value is ignored.

xa_open string format for earlier versions

Earlier versions of DB2 used the xa_open string format described here. This format is still supported for compatibility reasons. Applications should be migrated to the new format when possible.

Each database is defined as a separate resource manager (RM) to the transaction manager (TM), and the database must be identified with an xa_open string that has the following syntax:

```
"database_alias<,userid,password>"
```

The *database_alias* is required to specify the alias name of the database. The alias name is the same as the database name unless you have explicitly cataloged an alias name after you created the database. The user name and password are optional and, depending on the authentication method, are used to provide authentication information to the database.

Examples

1. You are using IBM TxSeries CICS on Windows. The TxSeries documentation indicates that you need to configure *tp_mon_name* with a value of `libEncServer:C`. This is still an acceptable format; however, with DB2 Database for Linux, UNIX, and Windows or DB2 Connect Version 8 FixPak 3 and later, you have the option of:
   - Specifying a *tp_mon_name* of `CICS` (recommended for this scenario):

     ```
     db2 update dbm cfg using tp_mon_name CICS
     ```

     For each database defined to CICS in the Region—> Resources—> Product—> XAD—> Resource manager initialization string, specify:

     ```
     db=dbalias,uid=userid,pwd=password
     ```
   - For each database defined to CICS in the Region—> Resources—> Product—> XAD—> Resource manager initialization string, specify:

     ```
     db=dbalias,uid=userid,pwd=password,tpm=cics
     ```
2. You are using IBM MQSeries on Windows. The MQSeries documentation indicates that you need to configure *tp_mon_name* with a value of `mqmax`. This is still an acceptable format; however, with DB2 Database for Linux, UNIX, and Windows or DB2 Connect Version 8 FixPak 3 and later, you have the option of:
   - Specifying a *tp_mon_name* of `MQ` (recommended for this scenario):

     ```
     db2 update dbm cfg using tp_mon_name MQ
     ```

     For each database defined to CICS in the Region—> Resources—> Product—> XAD—> Resource manager initialization string, specify:

     ```
     uid=userid,db=dbalias,pwd=password
     ```
   - For each database defined to CICS in the Region—> Resources—> Product—> XAD—> Resource manager initialization string, specify:

     ```
     uid=userid,db=dbalias,pwd=password,tpm=mq
     ```
3. You are using both IBM TxSeries CICS and IBM MQSeries on Windows. A single DB2 instance is being used. In this scenario, you would configure as follows:
   a. For each database defined to CICS in the Region—> Resources—> Product—> XAD—> Resource manager initialization string, specify:

      ```
      pwd=password,uid=userid,tpm=cics,db=dbalias
      ```

   b. For each database defined as a resource in the queue manager properties, specify an XaOpenString as:

      ```
      db=dbalias,uid=userid,pwd=password,tpm=mq
      ```

4. You are developing your own XA-compliant transaction manager (XA TM) on Windows, and you want to tell DB2 that library ″myaxlib″ has the required functions ax_reg and ax_unreg. Library ″myaxlib″ is in a directory specified in the PATH statement. You have the option of:
   - Specifying a *tp_mon_name* of `myaxlib`:

     ```
     db2 update dbm cfg using tp_mon_name myaxlib
     ```

     and, for each database defined to the XA TM, specifying an xa_open string:

     ```
     db=dbalias,uid=userid,pwd=password
     ```

   - For each database defined to the XA TM, specifying an xa_open string:

     ```
     db=dbalias,uid=userid,pwd=password,axlib=myaxlib
     ```

5. You are developing your own XA-compliant transaction manager (XA TM) on Windows, and you want to tell DB2 that library ″myaxlib″ has the required functions ax_reg and ax_unreg. Library ″myaxlib″ is in a directory specified in the PATH statement. You also want to enable XA END chaining. You have the option of:
   - For each database defined to the XA TM, specifying an xa_open string:

     ```
     db=dbalias,uid=userid,pwd=password,axlib=myaxlib,chain_end=T
     ```

   - For each database defined to the XA TM, specifying an xa_open string:

     ```
     db=dbalias,uid=userid,pwd=password,axlib=myaxlib,chain_end
     ```

Updating host or System i database servers with an XA-compliant transaction manager

Host and System i database servers may be updatable depending upon the architecture of the XA Transaction Manager.

- To support commit sequences from different processes, the DB2 Connect connection concentrator must be enabled.
- DB2 Connect with the DB2 sync point manager (SPM) must be configured.

To enable the DB2 Connect connection concentrator, set the database manager configuration parameter *max_connections* to a value greater than *max_coordagents*.

Note: The DB2 Connect connection concentrator requires a DB2 Universal Database (DB2 UDB) Version 7.1 client or later to support XA commit sequences from different processes.

Resolving indoubt transactions manually

An XA-compliant transaction manager (Transaction Processing Monitor) uses a two-phase commit process similar to that used by the DB2 transaction manager. The principal difference between the two environments is that the TP monitor provides the function of logging and controlling the transaction, instead of the DB2 transaction manager and the transaction manager database.

Errors similar to those that occur for the DB2 transaction manager can occur when using an XA-compliant transaction manager. Similar to the DB2 transaction manager, an XA-compliant transaction manager will attempt to resynchronize indoubt transactions.

If you cannot wait for the transaction manager to automatically resolve indoubt transactions, you can manually resolve them. This manual process is sometimes referred to as “making a heuristic decision”.

The LIST INDOUBT TRANSACTIONS command (using the WITH PROMPTING option), or the related set of APIs (*db2XaListIndTrans*, *sqlxphcm*, *sqlxhfrg*, *sqlxphrl*), allows you to query, commit, and roll back indoubt transactions. In addition, it also allows you to “forget” transactions that have been heuristically committed or rolled back, by removing the log records and releasing the log space.

Manually resolve indoubt transactions by using these commands (or related APIs) with *extreme caution*, and only as a last resort. The best strategy is to wait for the transaction manager to drive the resynchronization process. You could experience data integrity problems if you manually commit or roll back a transaction in one of the participating databases, and the opposite action is taken against another participating database. Recovering from data integrity problems requires you to understand the application logic, to identify the data that was changed or rolled back, and then to perform a point-in-time recovery of the database, or manually undo or reapply the changes.If you cannot wait for the transaction manager to initiate the resynchronization process, and you must release the resources tied up by an indoubt transaction, heuristic operations are necessary. This situation could occur if the transaction manager will not be available for an extended period of time to perform the resynchronization, and the indoubt transaction is tying up resources that are urgently needed. An indoubt transaction ties up the resources that were associated with this transaction before the transaction manager or resource managers became unavailable. For the database manager, these resources include locks on tables and indexes, log space, and storage taken up by the transaction. Each indoubt transaction also decreases (by one) the maximum number of concurrent transactions that can be handled by the database. Moreover, an offline backup cannot be taken unless all indoubt transactions have been resolved.The heuristic forget function is required in the following situations:

- When a heuristically committed or rolled back transaction causes a log full condition, indicated in output from the LIST INDOUBT TRANSACTIONS command
- When an offline backup is to be taken

The heuristic forget function releases the log space occupied by an indoubt transaction. The implication is that if a transaction manager eventually performs a resynchronization operation for this indoubt transaction, it could potentially make the wrong decision to commit or roll back other resource managers, because there is no log record for the transaction in this resource manager. In general a “missing” log record implies that the resource manager has rolled back the transaction.

1. Connect to the database for which you require all transactions to be complete.
2. Display the indoubt transactions:
   a. For DB2 database servers, use the LIST INDOUBT TRANSACTIONS WITH PROMPTING command. The *xid* represents the global transaction ID, and is identical to the *xid* used by the transaction manager and by other resource managers participating in the transaction.
   b. For host or System i database servers, you may use one of the following: You can obtain indoubt information directly from the host or System i server. To obtain indoubt information directly from DB2 for z/OS and OS/390, invoke the DISPLAY THREAD TYPE(INDOUBT) command. Use the RECOVER command to make a heuristic decision. To obtain indoubt information directly from DB2 for i5/OS, invoke the wrkcmtdfn command.

You can obtain indoubt information from the DB2 Connect server used to access the host or System i database server. You can obtain indoubt information from the DB2 Connect server used to access the host or System i database server.

3. For each indoubt transaction that has been listed or displayed, use the information shown about the application and the operating environment to determine the other participating resource managers.
4. Determine the actions to take with each indoubt transaction: If the transaction manager is available, and the indoubt transaction in a resource manager was caused by the resource manager not being available in the second commit phase, or for an earlier resynchronization process, you should do the following:
   a. Check the transaction manager's log to determine what action has been taken against the other resource managers.
   b. Take the same action against the database; that is, use the LIST INDOUBT TRANSACTIONS WITH PROMPTING command, to either heuristically commit or heuristically roll back the transaction.

   If the transaction manager is *not* available, use the status of the transaction in the other participating resource managers to determine what action you should take:
   - If at least one of the other resource managers has committed the transaction, heuristically commit the transaction in all the resource managers.
   - If at least one of the other resource managers has rolled back the transaction, heuristically roll back the transaction.
   - If the transaction is in the "prepared" (indoubt) state in all of the participating resource managers, heuristically roll back the transaction.
   - If one or more of the other resource managers is not available, heuristically roll back the transaction.

To obtain indoubt transaction information from DB2 on UNIX or Windows, connect to the database and issue the LIST INDOUBT TRANSACTIONS WITH PROMPTING command, or call the *db2XaListIndTrans* API from a client application.

Indoubt transaction management APIs

Databases can be used in a distributed transaction processing (DTP) environment.

A set of APIs is provided for tool writers to perform heuristic functions on indoubt transactions when the resource owner (such as the database administrator) cannot wait for the Transaction Manager (TM) to perform the *re-sync* action. This condition may occur if, for example, the communication line is broken, and an indoubt transaction is tying up needed resources. For the database manager, these resources include locks on tables and indexes, log space, and storage used by the transaction. Each indoubt transaction also decreases, by one, the maximum number of concurrent transactions that could be processed by the database manager.

The heuristic APIs have the capability to query, commit, and roll back indoubt transactions, and to cancel transactions that have been heuristically committed or rolled back, by removing the log records and releasing log pages.

Attention: The heuristic APIs should be used with caution and only as a last resort. The TM should drive the re-sync events. If the TM has an operator command to start the re-sync action, it should be used. If the user cannot wait for a TM-initiated re-sync, heuristic actions are necessary.

Although there is no set way to perform these actions, the following guidelines may be helpful:

- Use the db2XaListIndTrans function to display the indoubt transactions. They have a status = ’P’ (prepared), and are not connected. The *gtrid* portion of an *xid* is the global transaction ID that is identical to that in other resource managers (RM) that participate in the global transaction.
- Use knowledge of the application and the operating environment to identify the other participating RMs.
- If the transaction manager is CICS, and the only RM is a CICS resource, perform a heuristic rollback.
- If the transaction manager is not CICS, use it to determine the status of the transaction that has the same *gtrid* as does the indoubt transaction.
- If at least one RM has committed or rolled back, perform a heuristic commit or a rollback.
- If they are all in the prepared state, perform a heuristic rollback.
- If at least one RM is not available, perform a heuristic rollback.

If the transaction manager is available, and the indoubt transaction is due to the RM not being available in the second phase, or in an earlier re-sync, the DBA should determine from the TM’s log what action has been taken against the other RMs, and then do the same. The *gtrid* is the matching key between the TM and the RMs.

Do not execute sqlxhfrg unless a heuristically committed or rolled back transaction happens to cause a log full condition. The forget function releases the log space occupied by this indoubt transaction. If a transaction manager eventually performs a re-sync action for this indoubt transaction, the TM could make the wrong decision to commit or to roll back other RMs, because no record was found in this RM. In general, a missing record implies that the RM has rolled back.

Security considerations for XA transaction managers

The TP monitor pre-allocates a set of server processes and runs the transactions from different users under the IDs of the server processes. To the database, each server process appears as a big application that has many units of work, all being run under the same ID associated with the server process.

For example, in an AIX environment using CICS, when a TXSeries CICS region is started, it is associated with the AIX user name under which it is defined. All the CICS Application Server processes are also being run under this TXSeries CICS ″master″ ID, which is usually defined as ″cics″. CICS users can invoke CICS transactions under their DCE login ID, and while in CICS, they can also change their ID using the CESN signon transaction. In either case, the end user’s ID is not available to the RM. Consequently, a CICS Application Process might be running transactions on behalf of many users, but they appear to the RM as a single program with many units of work from the same ″cics″ ID. Optionally, you can specify a user ID and password on the xa_open string, and that user ID will be used, instead of the ″cics″ ID, to connect to the database.

There is not much impact on static SQL statements, because the binder’s privileges, not the end user’s privileges, are used to access the database. This does mean, however, that the EXECUTE privilege of the database packages must be granted to the server ID, and not to the end user ID.

For dynamic statements, which have their access authentication done at run time, access privileges to the database objects must be granted to the server ID and not to the actual user of those objects. Instead of relying on the database to control the access of specific users, you must rely on the TP monitor system to determine which users can run which programs. The server ID must be granted all privileges that its SQL users require.

To determine who has accessed a database table or view, you can perform the following steps:

1. From the SYSCAT.PACKAGEDEP catalog view, obtain a list of all packages that depend on the table or view.
2. Determine the names of the server programs (for example, CICS programs) that correspond to these packages through the naming convention used in your installation.
3. Determine the client programs (for example, CICS transaction IDs) that could invoke these programs, and then use the TP monitor's log (for example, the CICS log) to determine who has run these transactions or programs, and when.

XA function supported by DB2 Database for Linux, UNIX, and Windows

DB2 Database for Linux, UNIX, and Windows supports the XA91 specification defined in *X/Open CAE Specification Distributed Transaction Processing: The XA Specification*, with the following exceptions:

- Asynchronous services

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

  The XA interface allows two ways to register an RM: static registration and dynamic registration. DB2 supports both dynamic and static registration. DB2 provides two switches to control the type of registration used.
  - *db2xa_switch_std* for dynamic registration
  - *db2xa_switch_static_std* for static registration
- Association migration

  DB2 V9.1 does not support transaction migration between threads of control.

XA switch usage and location

As required by the XA interface, the database manager provides a *db2xa_switch_std* and a *db2xa_switch_static_std* external C variable of type xa_switch_t to return the XA switch structure to the TM. Other than the addresses of various XA functions, the following fields are returned:

Field **Value**

name The product name of the database manager. For example, IBM DB2 Version 9.1 for AIX.

flags For *db2xa_switch_std* TMREGISTER | TMNOMIGRATE is set

Explicitly states that DB2 V9.1 uses dynamic registration, and that the TM should not use association migration. Implicitly states that asynchronous operation is not supported.

For *db2xa_switch_static_std* TMNOMIGRATE is set

Explicitly states that DB2 V9.1 uses static registration, and that the TM should not use association migration. Implicitly states that asynchronous operation is not supported.

version
Must be zero.

Using the DB2 Database for Linux, UNIX, and Windows XA switch

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

Linux and UNIX

The switch for DB2 Database for Linux, UNIX, and Windows can be obtained through either of the following two ways:

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

  ```
  #define db2xa_switch_std (*db2xa_switch_std)
  #define db2xa_switch_static_std (*db2xa_switch_std)
  ```

  prior to using *db2xa_switch_std* or *db2xa_switch_static_std*.
- By calling db2xacic_std or db2xacicst_std

  DB2 provides these APIs, which return the address of the *db2xa_switch_std* or *db2xa_switch_static_std* structure. This function is prototyped as:

  ```
  struct xa_switch_t * SQL_API_FN  db2xacic_std( )
  struct xa_switch_t * SQL_API_FN  db2xacicst_std( )
  ```

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

Windows

The pointer to the *xa_switch* structure, *db2xa_switch_std*, or *db2xa_switch_static_std* is exported as DLL data. This implies that a Windows application using this structure must reference it in one of three ways:

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

  ```
  #define db2xa_switch_std (*db2xa_switch_std)
  #define db2xa_switch_static_std (*db2xa_switch_std)
  ```

  prior to using *db2xa_switch_std* or *db2xa_switch_static_std*.
- If using the Microsoft Visual C++ compiler, *db2xa_switch_std* or *db2xa_switch_static_std* can be defined as:

  ```
  extern __declspec(dllimport) struct xa_switch_t db2xa_switch_std
  extern __declspec(dllimport) struct xa_switch_t db2xa_switch_static_std
  ```
- By calling db2xacic_std or db2xacicst_std

  DB2 provides this API, which returns the address of the *db2xa_switch_std* or *db2xa_switch_static_std* structure. This function is prototyped as:

  ```
  struct xa_switch_t * SQL_API_FN  db2xacic_std( )
  struct xa_switch_t * SQL_API_FN  db2xacicst_std( )
  ```

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

Note: When linking with db2api.lib, the functions will resolve to different dlls on Windows 32-bit and Windows 64-bit platforms. The runtime dlls on a 64-bit platform will have the same base name as the 32-bit version with the addition of the ″64″ suffix. For example, db2app.dll on a Windows 32-bit operating system is equivalent to db2app64.dll on a Windows 64-bit operating system.

Example C Code

The following code illustrates the different ways in which the *db2xa_switch_std* or *db2xa_switch_static_std* can be accessed via a C program on any DB2 V9.1 platform. Be sure to link your application with the appropriate library.

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

   struct xa_switch_t * SQL_API_FN  db2xacic_std( );

   #ifdef DECLSPEC_DEFN
   extern __declspec(dllimport) struct xa_switch_t db2xa_switch_std;
   #else
   #define db2xa_switch_std (*db2xa_switch_std)
   extern struct xa_switch_t db2xa_switch_std;
   #endif

main( )
   {
      struct xa_switch_t *foo;
      printf ( "switch_std.name );
      foo = db2xacic_std();
      printf ( "name );
      return ;
   }
```

XA interface problem determination

When an error is detected during an XA request from the TM, the application program may not be able to get the error code from the TM. If your program abends, or gets a cryptic return code from the TP monitor or the TM, you should check the First Failure Service Log, which reports XA error information when diagnostic level 3 or greater is in effect.

You should also consult the console message, TM error file, or other product-specific information about the external transaction processing software that you are using.

The database manager writes all XA-specific errors to the First Failure Service Log with SQLCODE -998 (transaction or heuristic errors) and the appropriate reason codes. Following are some of the more common errors:

- Invalid syntax in the xa_open string.
- Failure to connect to the database specified in the open string as a result of one of the following:
  – The database has not been cataloged.
  – The database has not been started.
  – The server application’s user name or password is not authorized to connect to the database.
- Communications error.

Configuration considerations for XA transaction managers

You should consider the following configuration parameters when you are setting up your TP monitor environment:

- *tp_mon_name*

  This database manager configuration parameter identifies the name of the TP monitor product being used ("CICS", or "ENCINA", for example).
- *tm_database*

  Because DB2 Database for Linux, UNIX, and Windows does *not* coordinate transactions in the XA environment, this database manager configuration parameter is not used for XA-coordinated transactions.
- *maxappls*

  This database configuration parameter specifies the maximum number of active applications allowed. The value of this parameter must be equal to or greater than the sum of the connected applications, plus the number of these applications that may be concurrently in the process of completing a two-phase commit or rollback. This sum should then be increased by the anticipated number of indoubt transactions that might exist at any one time.

  For a TP monitor environment (for example, TXSeries CICS), you may need to increase the value of the *maxappls* parameter. This would help to ensure that all TP monitor processes can be accommodated.
- *autorestart*

  This database configuration parameter specifies whether the RESTART DATABASE routine will be invoked automatically when needed. The default value is `YES` (that is, enabled).

  A database containing indoubt transactions requires a restart database operation to start up. If *autorestart* is not enabled when the last connection to the database is dropped, the next connection will fail and require an explicit RESTART DATABASE invocation. This condition will exist until the indoubt transactions have been removed, either by the transaction manager's resync operation, or through a heuristic operation initiated by the administrator. When the RESTART DATABASE command is issued, a message is returned if there are any indoubt transactions in the database. The administrator can then use the LIST INDOUBT TRANSACTIONS command and other command line processor commands to find get information about those indoubt transactions.

Resource manager setup: Each database is defined as a separate resource manager (RM) to the transaction manager (TM), and the database must be identified with an xa_open string.

When setting up a database as a resource manager, you do not need the xa_close string. If provided, this string will be ignored by the database manager.

Database connection considerations

Automatic client reroute (ACR)

Whenever a server crashes, each client that is connected to that server gets a communication error which terminates the connection and concludes in an application error. In application environments where availability is important, the user will either have a redundant setup or will fail the server over to a standby node. In either case, the DB2 Database for Linux, UNIX, and Windows client code will attempt to re-establish the connection to either the original database (which may be running on a failover node where the IP address fails over as well), or to a new database on a different server. The application is then notified using an SQLCODE to indicate that the connection has been rerouted and that the specific transaction being run has been rolled back. At that point, the application can choose to rerun that transaction or continue on.

Data consistency between the failed primary database and the "failed to" standby database when using ACR is very dependent upon the state of the database logs in the database to which the connection has been rerouted. For the purposes of this discussion, we will call this database the "standby database" and the server on which this standby database resides the "standby server". If the standby database is an exact copy of the failed primary database at the point in time of the failure then the data at the standby database will be consistent and there will be no data integrity issues. However, if the standby database is not an exact copy of the failed primary database then there may be data integrity issues resulting from inconsistent transaction outcomes for transactions which have been prepared by the XA Transaction Manager but yet to be committed. These are known as indoubt transactions. The Database Administrator and application developers who are using the ACR function must be aware of the risk of data integrity problems when using this capability.

The following sections describe the various DB2 Database for Linux, UNIX, and Windows environments and the risks of data integrity problems in each.

High availability disaster recovery (HADR)

DB2's High Availability Disaster Recovery feature (HADR) can be used to control the level of log duplication between the primary and standby databases when the application regains connectivity after a primary database failure. The database configuration parameter which controls the level of log duplication is called *hadr_syncmode*. There are three possible values for this parameter:

- SYNC

  This mode provides the greatest protection against transaction loss at the cost of longest transaction response time among the three modes. As the name of this mode suggests, SYNC is used to synchronize the writing of the transaction log in the primary database and in the standby database. Synchronization is accomplished when the primary database has written its own log files and it has received acknowledgement from the standby database that the logs have also been written on the standby database.

  If an XA Transaction Manager is being used to coordinate transactions involving DB2 resources, then it is strongly recommended that SYNC mode be used. SYNC mode will guarantee data integrity as well as transaction resynchronization integrity when a client is rerouted to the standby database since it is an exact replica of the primary database.

- NEARSYNC

  This mode provides slightly less protection against transaction loss, in exchange for a shorter transaction response time when compared with SYNC mode. The primary database considers log write successful only when logs have been written to its own log files and it has received acknowledgement from the standby database that the logs have also been written to main memory on the standby database. If the standby database crashes before it can copy the logs from memory to disk, the logs are lost on the standby database in the short term.

  Given the possibility that database logs are lost, and the situation where the standby database is not an exact replica of the primary database, it is possible that data integrity will be compromised. The compromise occurs if the given transaction was indoubt and then the primary database crashes. Assume the transaction outcome is COMMIT. When the XA TM issues the subsequent XA_COMMIT request, it will fail since the primary database has crashed. Since the XA_COMMIT request has failed, the XA TM will need to recover this transaction on this database by issuing an XA_RECOVER request. The standby database will respond by returning the list of all its transactions which are INDOUBT. If the standby database were to crash and restart before the "in memory," database logs were written to disk, and before the XA_RECOVER request was issued by the XA TM, the standby database would have lost the log information about the transaction and could not return it in response to the XA_RECOVER request. The XA TM would then assume the database committed this transaction. But, what has really occurred is the data manipulation will have been lost and the appearance that the transaction was rolled back. This results in a data integrity issue since all other resources involved in this transaction were COMMITTED by the XA TM.

Using NEARSYNC is a good compromise between data integrity and transaction response time since the likelihood of both the primary and standby databases crashing should be low. However, a database administrator still needs to understand that there is a possibility of data integrity problems.

- ASYNC

  This mode has the greatest chance of transaction loss in the event of primary failure, in exchange for the shortest transaction response time among the three modes. The primary database considers log write successful only when logs have been written to its own log files and the logs have been delivered to the TCP layer on the primary database's host machine. The primary database does not wait for acknowledgement of any kind from the standby database. The logs may be still on their way to the standby database when the primary database considers relevant transactions committed.

  If the same scenario as described in NEARSYNC occurs, the likelihood of loss of transaction information is higher than with NEARSYNC. Therefore, the likelihood of data integrity issues is higher than with NEARSYNC and, obviously, with SYNC.

DB2 ESE Partitioned Database Environments

The use of ACR in partitioned database environments can also lead to data integrity issues. If the standby database is defined to be a different database partition of the same database, then recovery of indoubt transactions in scenarios as described in the High Availability Disaster Recovery NEARSYNC section above, may result in data integrity problems. This occurs because the database partitions do not share database transaction logs. Therefore the standby database (database partition B) will have no knowledge of indoubt transactions that exist at the primary database (database partition A).

DB2 ESE Non Partitioned Database Environments

The use of ACR in non-partitioned database environments can also lead to data integrity issues. Assuming disk failover technology, such as IBM AIX High Availability Cluster Multiprocessor (HACMP™), Microsoft Cluster Service (MSCS), or HP's Service Guard, is not in use then the standby database will not have the database transaction logs that existed on the primary database when it failed. Therefore, the recovery of indoubt transactions in scenarios as described in the High Availability Disaster Recovery NEARSYNC section above, can result in data integrity problems.

Transactions accessing partitioned databases

In a partitioned database environment, user data may be distributed across database partitions. An application accessing the database connects and sends requests to one of the database partitions (the coordinator node). Different applications can connect to different database partitions, and the same application can choose different database partitions for different connections.

For transactions against a database in a partitioned database environment, all access must be through the *same* database partition. That is, the same database partition must be used from the start of the transaction until (and including) the time that the transaction is committed.

Any transaction against the partitioned database must be committed before disconnecting.

Configuring IBM WebSphere Application Server:

IBM WebSphere Application Server is a Java-based application server. It can use the DB2 Database for Linux, UNIX, and Windows XA support via the Java Transaction API (JTA) provided by the DB2 JDBC driver. Refer to IBM WebSphere documentation regarding how to use the Java Transaction API with WebSphere Application Server. WebSphere Application Server documentation can be viewed online at http://www.ibm.com/software/webservers/appserv/was/library/.

Configuring IBM WebSphere MQ:

IBM WebSphere MQ is a message oriented business integration server. It can use the DB2 Database for Linux, UNIX, and Windows XA support to integrate database updates within WebSphere MQ units of work. Refer to IBM WebSphere MQ documentation regarding how to use the transactional support in WebSphere MQ. WebSphere MQ documentation can be viewed online at http://publib.boulder.ibm.com/infocenter/wmqv6/v6r0/index.jsp.

Configuring IBM TXSeries CICS:

For information about how to configure IBM TXSeries CICS to use DB2 Database for Linux, UNIX, and Windows as a resource manager, refer to your *IBM TXSeries CICS Administration Guide*. TXSeries documentation can be viewed online at http://publibfp.boulder.ibm.com/epubs/html/txv5002/en_US/html/index.htm. Host and System i database servers can participate in CICS-coordinated transactions.

Configuring IBM TXSeries Encina:

Following are the various APIs and configuration parameters required for the integration of Encina Monitor and DB2 Database for Linux, UNIX, and Windows servers, or DB2 for z/OS and OS/390, DB2 for i5/OS, or DB2 for VSE & VM when accessed through DB2 Connect.

TXSeries documentation can be viewed online athttp://publibfp.boulder.ibm.com/epubs/html/txv5002/en_US/html/index.htm. Host and System i database servers can participate in Encina-coordinated transactions.

1. To configure DB2 Database for Linux, UNIX, and Windows:
   a. Each database name must be defined in the DB2 database directory. If the database is a remote database, a node directory entry must also be defined. You can perform the configuration using the Configuration Assistant, or the DB2 command line processor (CLP). For example:

      ```
      DB2 CATALOG DATABASE inventdb AS inventdb AT NODE host1 AUTH SERVER
      DB2 CATALOG TCPIP NODE host1 REMOTE hostname1 SERVER svcname1
      ```

   b. The DB2 client can optimize its internal processing for Encina if it knows that it is dealing with Encina. You can specify this by setting the *tp_mon_name* database manager configuration parameter to `ENCINA`. The default behavior is no special optimization. If *tp_mon_name* is set, the application must ensure that the thread that performs the unit of work also immediately commits the work after ending it. No other unit of work may be started. If this is *not* your environment, ensure that the *tp_mon_name* value is `NONE` (or, through the CLP, that the value is set to `NULL`). The parameter can be updated through the Control Center or the CLP. The CLP command is:

      ```
      db2 update dbm cfg using tp_mon_name ENCINA
      ```

2. Configuring Encina for Each Resource Manager

   a. To configure Encina for each resource manager (RM), an administrator must define the Open String, Close String, and Thread of Control Agreement for each DB2 database as a resource manager before the resource manager can be registered for transactions in an application. The configuration can be performed using the Enconcole full screen interface, or the Encina command line interface. For example:

      ```
      monadmin create rm inventdb -open "db=inventdb,uid=user1,pwd=password1"
      ```

      There is one resource manager configuration for each DB2 database, and each resource manager configuration must have an rm name (″logical RM name″). To simplify the situation, you should make it identical to the database name. The xa_open string contains information that is required to establish a connection to the database. The content of the string is RM-specific. The xa_open string of DB2 contains the alias name of the database to be opened, and optionally, a user ID and password to be associated with the connection. The database name defined here must also be cataloged into the regular database directory required for all database access. The xa_close string is not used by DB2. The Thread of Control Agreement determines if an application agent thread can handle more than one transaction at a time. If you are accessing DB2 for z/OS and OS/390, DB2 for i5/OS, or DB2 for VSE & VM, you must use the DB2 Syncpoint Manager.

3. Referencing a DB2 Database for Linux, UNIX, and Windows database from an Encina application

   a. Use the Encina Scheduling Policy API to specify how many application agents can be run from a single TP monitor application process. For example:

      ```
      rc = mon_SetSchedulingPolicy (MON_EXCLUSIVE)
      ```

   b. Use the Encina RM Registration API to provide the XA switch and the logical RM name to be used by Encina when referencing the RM in an application process. For example:

      ```
      rc = mon_RegisterRmi ( &db2xa_switch,   /* xa switch */
                             "inventdb",      /* logical RM name */
                             &rmiId );        /* internal RM ID */
      ```

      The XA switch contains the addresses of the XA routines in the RM that the TM can call, and it also specifies the functionality that is provided by the RM. The XA switch of DB2 V9.1 is `db2xa_switch`, and it resides in the DB2 client library (`db2app.dll` on Windows operating systems and `libdb2` on UNIX based systems). The logical RM name is the one used by Encina, and is not the actual database name that is used by the SQL application that runs under Encina. The actual database name is specified in the xa_open string in the Encina RM Registration API. The logical RM name is set to be the same as the database name in this example. The third parameter returns an internal identifier or handle that is used by the TM to reference this connection.

Configuring BEA Tuxedo:

What follows is a description of the process to configure BEA Tuxedo for use with DB2 Database for Linux, UNIX, and Windows. There are some differences that are noted based on whether Tuxedo is working with a 64-bit instance of DB2 Database for Linux, UNIX, and Windows or a 32-bit instance of DB2 Database for Linux, UNIX, and Windows.

Note: There are new names for the XA switch data structures: *db2xa_switch_std* and *db2xa_switch_static_std*. There are also new names for the APIs: *db2xacic* and *db2xacicst*. The old switch data structure and API names can be used but only when working with a 32-bit instance of DB2 Database for Linux, UNIX, and Windows.

To configure Tuxedo to use DB2 Database for Linux, UNIX, and Windows as a resource manager, perform the following steps:

1. Install Tuxedo as specified in the documentation for that product. Ensure that you perform all basic Tuxedo configuration, including the log files and environment variables. You also require a compiler and the Data Server Client. Install these if necessary.
2. At the Tuxedo server ID, set the DB2INSTANCE environment variable to reference the instance that contains the databases that you want Tuxedo to use. Set the PATH variable to include the DB2 program directories. Confirm that the Tuxedo server ID can connect to the DB2 databases.
3. Update the *tp_mon_name* database manager configuration parameter with the value `TUXEDO`.
4. Add a definition for DB2 V9.1 to the Tuxedo resource manager definition file. In the examples that follow, `UDB_XA` is the locally-defined Tuxedo resource manager name for DB2 V9.1, and *db2xa_switch_std* is the DB2-defined name for a structure of type `xa_switch_t`:
   - For AIX. In the file ${TUXDIR}/udataobj/RM, add the definition:

     ```
     # DB2 UDB
     UDB_XA:db2xa_switch_std:-L${DB2DIR} /lib -ldb2
     ```

     where {TUXDIR} is the directory where you installed Tuxedo, and {DB2DIR} is the DB2 instance directory.
   - For Windows. In the file %TUXDIR%\udataobj\rm, add the definition:

     ```
     # DB2 UDB
     UDB_XA;db2xa_switch_std;%DB2DIR%\lib\db2api.lib
     ```

     where %TUXDIR% is the directory where you installed Tuxedo, and %DB2DIR% is the DB2 instance directory.
5. Build the Tuxedo transaction monitor server program for DB2:
   - For AIX:

     ```
     ${TUXDIR}/bin/buildtms -r UDB_XA -o ${TUXDIR}/bin/TMS_UDB
     ```

     where {TUXDIR} is the directory where you installed Tuxedo.
   - For Windows:

```
XA -o UDB
```

6. Build the application servers. In the examples that follow, the `-r` option specifies the resource manager name, the `-f` option (used one or more times) specifies the files that contain the application services, the `-s` option specifies the application service names for this server, and the `-o` option specifies the output server file name:
   - For AIX:

     ```
     ${TUXDIR}/bin/buildserver -r UDB_XA -f svcfile.o -s SVC1,SVC2
        -o UDBserver
     ```

     where {TUXDIR} is the directory where you installed Tuxedo.
   - For Windows:

     ```
     XA -f svcfile.o -s SVC1,SVC2
        -o UDBserver
     ```

     where %TUXDIR% is the directory where you installed Tuxedo.
7. Set up the Tuxedo configuration file to reference the DB2 server. In the *GROUPS section of the UDBCONFIG file, add an entry similar to:

   ```
   UDB_GRP   LMID=simp GRPNO=3
     TMSNAME=TMS_UDB TMSCOUNT=2
     OPENINFO="UDB_XA:db=sample,uid=db2_user,pwd=db2_user_pwd"
   ```

   where the TMSNAME parameter specifies the transaction monitor server program that you built previously, and the OPENINFO parameter specifies the resource manager name. This is followed by the database name, and the DB2 database user ID and password, which are used for authentication. The application servers that you built previously are referenced in the *SERVERS section of the Tuxedo configuration file.
8. If the application is accessing data residing on DB2 for z/OS and OS/390, DB2 for i5/OS, or DB2 for VM&VSE, the DB2 Connect XA concentrator will be required.
9. Start Tuxedo:

   ```
   tmboot -y
   ```

   After the command completes, Tuxedo messages should indicate that the servers are started. In addition, if you issue the DB2 command LIST APPLICATIONS ALL, you should see two connections (in this situation) specified by the TMSCOUNT parameter in the UDB_GRP in the Tuxedo configuration file, UDBCONFIG

Setting up the unixODBC driver manager

To run ODBC applications on Linux or UNIX operating systems, you must configure the unixODBC Driver Manager.

To set up the unixODBC Driver Manager for use with DB2 CLI and ODBC applications:

1. Download the latest unixODBC source code from http://www.unixodbc.com.
2. Untar the source files. For example:

   ```
   gzip -d unixODBC-2.2.11.tar.gz
   tar xf unixODBC-2.2.11.tar
   ```

3. For AIX only: Configure the C compiler to be thread-enabled:

   ```
   export CC=xlc_r
   export CCC=xlC_r
   ```

4. To compile a 64-bit version of the driver manager using the xlc_r compilers, set the environment variables **OBJECT_MODE** and **CFLAGS**:

   ```
   export OBJECT_MODE=64
   export CFLAGS=-q64 -DBUILD_REAL_64_BIT_MODE
   ```

5. Install the driver manager in either your home directory or the default /usr/local prefix:

   - (Home directory) Issue the following command in the directory where you untarred the source files:

     ```
     ./configure --prefix=$HOME -DBUILD_REAL_64_BIT_MODE --enable-gui=no
                                        --enable-drivers=no
     ```

   - (/usr/local as root) Issue the following command:

     ```
     ./configure --enable-gui=no --enable-drivers=no
     ```

6. Optional: Examine all configuration options by issuing the following command:

   ```
   ./configure --help
   ```

7. Build and install the driver manager:

   ```
   make
   make install
   ```

   Libraries will be copied to the [prefix]/lib directory, and executables will be copied to the [prefix]/bin directory.

8. For AIX only: Extract the shared library from the ODBC driver for DB2 to yield shr.o on 32-bit operating systems and shr_64.o on 64-bit operating systems. To avoid confusion, rename the files db2.o and db2_64.o, respectively. These steps are necessary on AIX because the unixODBC Driver Manager loads the driver dynamically.

   - On 32-bit operating systems, issue the following commands:

     ```
     cd INSTHOME/sqllib/lib
     ar -x libdb2.a
     mv shr.o db2.o
     ```

     where *INSTHOME* is the home directory of the instance owner.

   - On 64-bit operating systems, issue the following commands:

     ```
     cd INSTHOME/sqllib/lib
     ar -x -X 64 libdb2.a
     mv shr_64.o db2_64.o
     ```

     where *INSTHOME* is the home directory of the instance owner.

   Ensure that your INI file references the correct library.

9. (Optional) For AIX only: Extract libodbc.a, libodbcinst.a, and libodbccr.a if you will be dynamically loading the driver manager:

   ```
   ar -x libodbc.a
   ar -x libodbcinst.a
   ar -x libodbccr.a
   ```

   This produces libodbc.so.1, libodbcinst.so.1, and libodbccr.so.1 respectively in the [prefix]/lib/so directory.

10. Build the application and ensure it is linked to the unixODBC Driver Manager by including the `-L[prefix]/lib -lodbc` option in the compile and link command.

11. Specify the paths for at least the user INI file (odbc.ini) or the system INI file (odbcinst.ini), and set the **ODBCHOME** environment variable to the directory where the system INI file was created.

Important: Provide absolute paths when specifying the paths of the user and system INI files. Do not use relative paths or environment variables.

Appendix A. Overview of the DB2 technical information

DB2 technical information is available through the following tools and methods:

- *DB2 Information Center*
  - Topics (Task, concept and reference topics)
  - Help for DB2 tools
  - Sample programs
  - Tutorials
- DB2 books
  - PDF files (downloadable)
  - PDF files (from the DB2 PDF DVD)
  - printed books
- Command line help
  - Command help
  - Message help

Note: The *DB2 Information Center* topics are updated more frequently than either the PDF or the hard-copy books. To get the most current information, install the documentation updates as they become available, or refer to the *DB2 Information Center* at ibm.com®.

You can access additional DB2 technical information such as technotes, white papers, and IBM Redbooks® publications online at ibm.com. Access the DB2 Information Management software library site at http://www.ibm.com/software/data/sw-library/.

Documentation feedback

We value your feedback on the DB2 documentation. If you have suggestions for how to improve the DB2 documentation, send an email to db2docs@ca.ibm.com. The DB2 documentation team reads all of your feedback, but cannot respond to you directly. Provide specific examples wherever possible so that we can better understand your concerns. If you are providing feedback on a specific topic or help file, include the topic title and URL.

Do not use this email address to contact DB2 Customer Support. If you have a DB2 technical issue that the documentation does not resolve, contact your local IBM service center for assistance.

If you would like to help IBM make the IBM Information Management products easier to use, take the Consumability Survey: http://www.ibm.com/software/data/info/consumability-survey/.

DB2 technical library in hardcopy or PDF format

The following tables describe the DB2 library available from the IBM Publications Center at www.ibm.com/shop/publications/order. English DB2 Version 9.5 manuals in PDF format and translated versions can be downloaded from www.ibm.com/support/docview.wss?rs=71&uid=swg2700947.

Although the tables identify books available in print, the books might not be available in your country or region.

The form number increases each time a manual is updated. Ensure that you are reading the most recent version of the manuals, as listed below.

Note: The *DB2 Information Center* is updated more frequently than either the PDF or the hard-copy books.

Table 21. DB2 technical information

Name	Form Number	Available in print	Last updated
Administrative API Reference	SC23-5842-02	Yes	April, 2009
Administrative Routines and Views	SC23-5843-02	No	April, 2009
Call Level Interface Guide and Reference, Volume 1	SC23-5844-02	Yes	April, 2009
Call Level Interface Guide and Reference, Volume 2	SC23-5845-02	Yes	April, 2009
Command Reference	SC23-5846-02	Yes	April, 2009
Data Movement Utilities Guide and Reference	SC23-5847-02	Yes	April, 2009
Data Recovery and High Availability Guide and Reference	SC23-5848-02	Yes	April, 2009
Data Servers, Databases, and Database Objects Guide	SC23-5849-02	Yes	April, 2009
Database Security Guide	SC23-5850-02	Yes	April, 2009
Developing ADO.NET and OLE DB Applications	SC23-5851-02	Yes	April, 2009
Developing Embedded SQL Applications	SC23-5852-02	Yes	April, 2009
Developing Java Applications	SC23-5853-02	Yes	April, 2009
Developing Perl and PHP Applications	SC23-5854-02	No	April, 2009
Developing User-defined Routines (SQL and External)	SC23-5855-02	Yes	April, 2009
Getting Started with Database Application Development	GC23-5856-02	Yes	April, 2009
Getting Started with DB2 installation and administration on Linux and Windows	GC23-5857-02	Yes	April, 2009
Internationalization Guide	SC23-5858-02	Yes	April, 2009

Table 21. DB2 technical information (continued)

Name	Form Number	Available in print	Last updated
Message Reference, Volume 1	GI11-7855-01	No	April, 2009
Message Reference, Volume 2	GI11-7856-01	No	April, 2009
Migration Guide	GC23-5859-02	Yes	April, 2009
Net Search Extender Administration and User's Guide	SC23-8509-02	Yes	April, 2009
Partitioning and Clustering Guide	SC23-5860-02	Yes	April, 2009
Query Patroller Administration and User's Guide	SC23-8507-01	Yes	April, 2009
Quick Beginnings for IBM Data Server Clients	GC23-5863-02	No	April, 2009
Quick Beginnings for DB2 Servers	GC23-5864-02	Yes	April, 2009
Spatial Extender and Geodetic Data Management Feature User's Guide and Reference	SC23-8508-02	Yes	April, 2009
SQL Reference, Volume 1	SC23-5861-02	Yes	April, 2009
SQL Reference, Volume 2	SC23-5862-02	Yes	April, 2009
System Monitor Guide and Reference	SC23-5865-02	Yes	April, 2009
Text Search Guide	SC23-5866-01	Yes	April, 2009
Troubleshooting Guide	GI11-7857-02	No	April, 2009
Tuning Database Performance	SC23-5867-02	Yes	April, 2009
Visual Explain Tutorial	SC23-5868-00	No	
What's New	SC23-5869-02	Yes	April, 2009
Workload Manager Guide and Reference	SC23-5870-02	Yes	April, 2009
pureXML Guide	SC23-5871-02	Yes	April, 2009
XQuery Reference	SC23-5872-02	No	April, 2009

Table 22. DB2 Connect-specific technical information

Name	Form Number	Available in print	Last updated
Quick Beginnings for DB2 Connect Personal Edition	GC23-5839-02	Yes	April, 2009
Quick Beginnings for DB2 Connect Servers	GC23-5840-02	Yes	April, 2009
DB2 Connect User's Guide	SC23-5841-02	Yes	April, 2009

Table 23. Information Integration technical information

Name	Form Number	Available in print	Last updated
Information Integration: Administration Guide for Federated Systems	SC19-1020-01	Yes	March, 2008
Information Integration: ASNCLP Program Reference for Replication and Event Publishing	SC19-1018-02	Yes	March, 2008
Information Integration: Configuration Guide for Federated Data Sources	SC19-1034-01	No	
Information Integration: SQL Replication Guide and Reference	SC19-1030-01	Yes	March, 2008
Information Integration: Introduction to Replication and Event Publishing	SC19-1028-01	Yes	March, 2008

Ordering printed DB2 books

If you require printed DB2 books, you can buy them online in many but not all countries or regions. You can always order printed DB2 books from your local IBM representative. Keep in mind that some softcopy books on the *DB2 PDF Documentation* DVD are unavailable in print. For example, neither volume of the *DB2 Message Reference* is available as a printed book.

Printed versions of many of the DB2 books available on the DB2 PDF Documentation DVD can be ordered for a fee from IBM. Depending on where you are placing your order from, you may be able to order books online, from the IBM Publications Center. If online ordering is not available in your country or region, you can always order printed DB2 books from your local IBM representative. Note that not all books on the DB2 PDF Documentation DVD are available in print.

Note: The most up-to-date and complete DB2 documentation is maintained in the DB2 Information Center at http://publib.boulder.ibm.com/infocenter/db2luw/v9r5.

To order printed DB2 books:

- To find out whether you can order printed DB2 books online in your country or region, check the IBM Publications Center at http://www.ibm.com/shop/publications/order. You must select a country, region, or language to access publication ordering information and then follow the ordering instructions for your location.
- To order printed DB2 books from your local IBM representative:
  1. Locate the contact information for your local representative from one of the following Web sites:
     - The IBM directory of world wide contacts at www.ibm.com/planetwide
     - The IBM Publications Web site at http://www.ibm.com/shop/publications/order. You will need to select your country, region, or

language to the access appropriate publications home page for your location. From this page, follow the ″About this site″ link.

2. When you call, specify that you want to order a DB2 publication.
3. Provide your representative with the titles and form numbers of the books that you want to order. For titles and form numbers, see "DB2 technical library in hardcopy or PDF format" on page 79.

Displaying SQL state help from the command line processor

DB2 returns an SQLSTATE value for conditions that could be the result of an SQL statement. SQLSTATE help explains the meanings of SQL states and SQL state class codes.

To invoke SQL state help, open the command line processor and enter:

```
? sqlstate or ? class code
```

where *sqlstate* represents a valid five-digit SQL state and *class code* represents the first two digits of the SQL state.
For example, `? 08003` displays help for the 08003 SQL state, and `? 08` displays help for the 08 class code.

Accessing different versions of the DB2 Information Center

For DB2 Version 9.5 topics, the DB2 Information Center URL is http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/

For DB2 Version 9 topics, the DB2 Information Center URL is http://publib.boulder.ibm.com/infocenter/db2luw/v9/

For DB2 Version 8 topics, go to the Version 8 Information Center URL at: http://publib.boulder.ibm.com/infocenter/db2luw/v8/

Displaying topics in your preferred language in the DB2 Information Center

The DB2 Information Center attempts to display topics in the language specified in your browser preferences. If a topic has not been translated into your preferred language, the DB2 Information Center displays the topic in English.

- To display topics in your preferred language in the Internet Explorer browser:
  1. In Internet Explorer, click the **Tools —> Internet Options —> Languages...** button. The Language Preferences window opens.
  2. Ensure your preferred language is specified as the first entry in the list of languages.
     - To add a new language to the list, click the **Add...** button.

       Note: Adding a language does not guarantee that the computer has the fonts required to display the topics in the preferred language.
     - To move a language to the top of the list, select the language and click the **Move Up** button until the language is first in the list of languages.
  3. Clear the browser cache and then refresh the page to display the DB2 Information Center in your preferred language.
- To display topics in your preferred language in a Firefox or Mozilla browser:

1. Select the button in the **Languages** section of the **Tools —> Options —> Advanced** dialog. The Languages panel is displayed in the Preferences window.
2. Ensure your preferred language is specified as the first entry in the list of languages.
   – To add a new language to the list, click the **Add...** button to select a language from the Add Languages window.
   – To move a language to the top of the list, select the language and click the **Move Up** button until the language is first in the list of languages.
3. Clear the browser cache and then refresh the page to display the DB2 Information Center in your preferred language.

On some browser and operating system combinations, you might have to also change the regional settings of your operating system to the locale and language of your choice.

Updating the DB2 Information Center installed on your computer or intranet server

If you have installed the DB2 Information Center locally, you can obtain and install documentation updates from IBM.

Updating your locally-installed *DB2 Information Center* requires that you:

1. Stop the *DB2 Information Center* on your computer, and restart the Information Center in stand-alone mode. Running the Information Center in stand-alone mode prevents other users on your network from accessing the Information Center, and allows you to apply updates. Non-Administrative and Non-Root *DB2 Information Centers* always run in stand-alone mode. .
2. Use the update feature to see what updates are available. If there are updates that you would like to install, you can use the update feature to obtain and install them.

   Note: If your environment requires installing the *DB2 Information Center* updates on a machine that is not connected to the internet, you have to mirror the update site to a local file system using a machine that is connected to the internet and has the *DB2 Information Center* installed. If many users on your network will be installing the documentation updates, you can reduce the time required for individuals to perform the updates by also mirroring the update site locally and creating a proxy for the update site.
   If update packages are available, use the update feature to get the packages. However, the update feature is only available in stand-alone mode.
3. Stop the stand-alone Information Center, and restart the *DB2 Information Center* on your computer.

Note: On Windows Vista, the commands listed below must be run as an administrator. To launch a command prompt or graphical tool with full administrator privileges, right-click on the shortcut and then select **Run as administrator**.

To update the *DB2 Information Center* installed on your computer or intranet server:

1. Stop the *DB2 Information Center*.
   - On Windows, click **Start** → **Control Panel** → **Administrative Tools** → **Services**. Then right-click on **DB2 Information Center** service and select **Stop**.

- On Linux, enter the following command:

  ```
  /etc/init.d/db2icdv95 stop
  ```

2. Start the Information Center in stand-alone mode.
   - On Windows:
     a. Open a command window.
     b. Navigate to the path where the Information Center is installed. By default, the *DB2 Information Center* is installed in the *Program_files*\IBM\DB2 Information Center\Version 9.5 directory, where *Program_files* represents the location of the Program Files directory.
     c. Navigate from the installation directory to the doc\bin directory.
     d. Run the help_start.bat file:

        ```
        help_start.bat
        ```

   - On Linux:
     a. Navigate to the path where the Information Center is installed. By default, the *DB2 Information Center* is installed in the /opt/ibm/db2ic/V9.5 directory.
     b. Navigate from the installation directory to the doc/bin directory.
     c. Run the help_start script:

        ```
        help_start
        ```

   The systems default Web browser launches to display the stand-alone Information Center.
3. Click the **Update** button ( ). On the right hand panel of the Information Center, click **Find Updates**. A list of updates for existing documentation displays.
4. To initiate the installation process, check the selections you want to install, then click **Install Updates**.
5. After the installation process has completed, click **Finish**.
6. Stop the stand-alone Information Center:
   - On Windows, navigate to the installation directory's doc\bin directory, and run the help_end.bat file:

     ```
     help_end.bat
     ```

     Note: The help_end batch file contains the commands required to safely terminate the processes that were started with the help_start batch file. Do not use `Ctrl-C` or any other method to terminate help_start.bat.
   - On Linux, navigate to the installation directory's doc/bin directory, and run the help_end script:

     ```
     help_end
     ```

     Note: The help_end script contains the commands required to safely terminate the processes that were started with the help_start script. Do not use any other method to terminate the help_start script.
7. Restart the *DB2 Information Center*.
   - On Windows, click **Start** → **Control Panel** → **Administrative Tools** → **Services**. Then right-click on **DB2 Information Center** service and select **Start**.
   - On Linux, enter the following command:

     ```
     /etc/init.d/db2icdv95 start
     ```

The updated *DB2 Information Center* displays the new and updated topics.

DB2 tutorials

The DB2 tutorials help you learn about various aspects of DB2 products. Lessons provide step-by-step instructions.

Before you begin

You can view the XHTML version of the tutorial from the Information Center at http://publib.boulder.ibm.com/infocenter/db2help/.

Some lessons use sample data or code. See the tutorial for a description of any prerequisites for its specific tasks.

DB2 tutorials

To view the tutorial, click on the title.

"pureXML™" in *pureXML Guide*
: Set up a DB2 database to store XML data and to perform basic operations with the native XML data store.

"Visual Explain" in *Visual Explain Tutorial*
: Analyze, optimize, and tune SQL statements for better performance using Visual Explain.

DB2 troubleshooting information

A wide variety of troubleshooting and problem determination information is available to assist you in using DB2 database products.

DB2 documentation
: Troubleshooting information can be found in the DB2 Troubleshooting Guide or the Database fundamentals section of the DB2 Information Center. There you will find information on how to isolate and identify problems using DB2 diagnostic tools and utilities, solutions to some of the most common problems, and other advice on how to solve problems you might encounter with your DB2 database products.

DB2 Technical Support Web site
: Refer to the DB2 Technical Support Web site if you are experiencing problems and want help finding possible causes and solutions. The Technical Support site has links to the latest DB2 publications, TechNotes, Authorized Program Analysis Reports (APARs or bug fixes), fix packs, and other resources. You can search through this knowledge base to find possible solutions to your problems.

  Access the DB2 Technical Support Web site at http://www.ibm.com/software/data/db2/support/db2_9/

Terms and Conditions

Permissions for the use of these publications is granted subject to the following terms and conditions.

Personal use: You may reproduce these Publications for your personal, non commercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these Publications, or any portion thereof, without the express consent of IBM.

Commercial use: You may reproduce, distribute and display these Publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these Publications, or reproduce, distribute or display these Publications or any portion thereof outside your enterprise, without the express consent of IBM.

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the Publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the Publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED ″AS-IS″ AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

Appendix B. Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

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

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

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country/region or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country/region where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions; therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

This document may provide links or references to non-IBM Web sites and resources. IBM makes no representations, warranties, or other commitments whatsoever about any non-IBM Web sites or third-party resources that may be referenced, accessible from, or linked from this document. A link to a non-IBM Web site does not mean that IBM endorses the content or use of such Web site or its owner. In addition, IBM is not a party to or responsible for any transactions you may enter into with third parties, even if you learn of such parties (or use a link to such parties) from an IBM site. Accordingly, you acknowledge and agree that IBM is not responsible for the availability of such external sites or resources, and is not responsible or liable for any content, services, products, or other materials on or available from those sites or resources. Any software provided by third parties is subject to the terms and conditions of the license that accompanies that software.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

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

IBM Canada Limited
Office of the Lab Director
8200 Warden Avenue
Markham, Ontario
L6G 1C7
CANADA

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

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems, and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements, or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

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

COPYRIGHT LICENSE:

This information may contain sample application programs, in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Each copy or any portion of these sample programs or any derivative work must include a copyright notice as follows:

© (*your company name*) (*year*). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _*enter the year or years*_. All rights reserved.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at Copyright and trademark information at www.ibm.com/legal/copytrade.shtml.

The following terms are trademarks or registered trademarks of other companies

- Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
- Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
- UNIX is a registered trademark of The Open Group in the United States and other countries.
- Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries. Intel trademark information
- Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Index

Special characters

.NET
- application development
  - supported software 29

A

ActiveX Data Object (ADO) specification
- IBM Data Server Provider for .NET 17

ActiveX data objects
- DB2 AD Client support 15

ADO (ActiveX Data Object) specification
- IBM Data Server Provider for .NET 17

ADO.NET applications
- developing
  - overview 17

AIX
- installation requirements 6

APIs
- heuristic 64
- supported 17

application development
- application programming interfaces (API) support 3
- application programming interfaces (APIs) 17, 29
  - configuring 48
- configuring 45
  - application programming interfaces (APIs) 48
  - DB2 Client 48
  - operating systems 46
  - Windows 47
- DB2 Client
  - configuring 48
- development tools support 3
- environment 1
  - configuring 45
- IBM data server client support 3
- IBM Data Server Client support 15
- IBM Data Server Provider for .NET 17
- operating systems 5
  - configuring 46
  - support 3
  - UNIX 46
- programming languages support 3
- supported
  - application programming interfaces (APIs) 17
  - programming languages 29
  - transaction managers 41
- supported development tools 43
- tools 43
- transaction manager support 3
- UNIX
  - configuring 46
- Windows
  - configuring 47

application programming interfaces (API)
- support for application development 3

application programs
- embedded SQL, overview 21
- Perl DBI 25

applications
- supported development tools 43
- supported programming interfaces 17

automatic client reroute 69

B

BEA Tuxedo
- configuring 74

books
- printed
  - ordering 82

C

C language
- supported versions 30

C/C++ language
- supported versions 31

call level interface (CLI)
- DB2 AD Client support 15

CLI (call level interface)
- environmental setup 49
- introduction 20

client reroute
- automatic 69

clients
- configuration 48

COBOL language
- supported versions 33

command line processor (CLP)
- DB2 AD Client support 15
- supported development tools 43

connection failures
- automatic client reroute 69

connection keywords
- ODBC .NET Data Provider 18
- OLE DB .NET Data Provider 19

core level functions 20

D

Data Studio
- supported development tools 43

database applications
- programming interfaces 17

DB2 Information Center
- languages 83
- updating 84
- versions 83
- viewing in different languages 83

development tools
- support for application development 3

Distributed Transaction Processing (DTP)
- application program (AP) 54
- configuring 68
- databases
  - connections 69
  - updates (host and System i) 62
- error handling 62

Distributed Transaction Processing (DTP) *(continued)*
- resource manager (RM) 54
- security 65
- transaction manager (TM) 54
- X/Open Distributed Transaction Processing (DTP) model 54

documentation
- overview 79
- PDF 79
- printed 79
- terms and conditions of use 86

DTP (Distributed Transaction Processing)
- See Distributed Transaction Processing (DTP) 54

E

Eclipse 3.2 environments
- extending using Data Studio 43

embedded SQL applications
- IBM Data Server Client support 15
- overview 21

environment variables
- UNIX
  - overview 46

F

FORTRAN language
- supported versions 34

H

hardware
- requirements
  - AIX 6
  - HP-UX 7
  - Linux 8
  - Solaris Operating Environment 11
  - Windows 12

help
- configuring language 83
- SQL statements 83

heuristic decisions
- resolving indoubt transactions 62

heuristic operations
- resolving indoubt transactions 62

host databases
- updating with XA transaction managers 62

HP-UX
- installing
  - DB2 servers 7
  - IBM data server clients 7

I

IBM data server clients
- support for application development 3

IBM Data Server Driver for JDBC and SQLJ
- installing 50

IBM Data Server Driver for JDBC and SQLJ level
- by DB2 version level 34

IBM Data Server Provider for .NET 17
- viewing documentation
  - Microsoft Visual Studio .NET 17

IBM Database Add-Ins for Visual Studio
- supported development tools 43

IBM OLE DB Provider
- consumer 24
- for DB2
  - installing 24
- provider 24

IBM TXSeries CICS
- configuring 72

IBM TXSeries Encina
- configuring 72

IBM_DB Rails adapter 26

IBM_DB Ruby driver 26

indoubt transactions
- recovering 54
- resolving 62

installing
- IBM Data Server Driver for JDBC and SQLJ 50
- requirements
  - AIX 6
  - HP-UX 7
  - Linux 8
  - Solaris Operating Environment 11
  - Windows 12

J

Java
- DB2 product support 34
- IBM Data Server Client support 15

Java Database Connectivity (JDBC)
- drivers
  - versions 34
- IBM Data Server Client support 15
- IBM Data Server Driver for JDBC and SQLJ installation 50

JDBC and SQLJ
- supported drivers 22

L

Linux
- installation requirements 8

Linux library
- libaio.so.1 8
- libstdc++so.5 8

LIST INDOUBT TRANSACTIONS command 62

M

Microsoft Visual Studio .NET
- viewing IBM Data Server Provider for .NET documentation 17

N

Network File System (NFS) installation
- on AIX 6
- on HP-UX 7
- on Linux 8
- on Solaris Operating Environment 11

notices 89

O

Object Linking and Embedding (OLE)
  DB2 AD Client support 15
ODBC .NET Data Provider
  overview 18
ODBC (open database connectivity)
  and DB2 CLI 20
  core level functions 20
  driver managers
    unixODBC 75
OLE DB
  table functions 24
OLE DB .NET Data Provider
  overview 19
operating system requirements
  AIX 6
  HP-UX 7
  Linux 8
  Solaris Operating Environment 11
  Windows 12
operating systems
  configuring 46
  support for application development 3
  supported 5
ordering DB2 books 82

P

partitioned database environments
  transactions
    accessing databases 69
Perl
  Database Interface (DBI) specification 25
  versions supported 36
PHP
  overview 25
  versions 37
precompilers
  IBM Data Server Client support 15
problem determination
  information available 86
  tutorials 86
programming languages
  C 30
  C++ 31
  COBOL 33
  FORTRAN 34
  Perl 36
  PHP 37
  REXX 38
  Ruby on Rails 38
  summary 29

R

Rails adapter
  description 26
Rational Application Developer (RAD)
  supported development tools 43
Rational Data Architect
  supported development tools 43
Rational Software Architect
  supported development tools 43
resolving indoubt transactions 62
resource managers (RM)
  described 54
resource managers (RM) *(continued)*
  setting up a database as 69
REXX language
  supported versions 38
Ruby driver 26
Ruby on Rails
  supported development tools 43
Ruby/Ruby on Rails language
  supported versions 38

S

SAMPLE database
  connecting
    ODBC .NET Data Provider 18
    OLE DB .NET Data Provider 19
SDK
  Java
    product levels 34
settings
  CLI environment
    run-time support 49
software requirements
  AIX 6
  HP-UX 7
  Linux 8
  Solaris Operating Environment 11
  Windows 12
Solaris Operating Environment
  installation requirements 11
SQL Access Group 20
SQL Editor
  supported development tools 43
SQL statements
  displaying help 83
SQLJ
  DB2 AD Client support 15
static profiling
  supported development tools 43
System i
  databases
    updating with XA transaction managers 62
system requirements
  IBM OLE DB Provider for DB2 24

T

terms and conditions
  use of publications 86
threads
  IBM OLE DB Provider for DB2 24
TPM values 57
TPMONNAME values 57
transaction manager
  support for application development 3
transaction managers 53
  BEA Tuxedo 74
  distributed transaction processing 54
  IBM TXSeries CICS 72
  IBM TXSeries Encina 72
  IBM WebSphere Application Server 72
  IBM WebSphere MQ 72
  problem determination 68
  supported 41
  XA architecture 66

transaction processing monitors
  BEA Tuxedo 74
  configuration considerations 68
  IBM TXSeries CICS 72
  IBM TXSeries Encina 72
  security considerations 65
  supported 41
transactions
  accessing partitioned databases 69
  global 54
  loosely coupled 54
  non-XA 54
  tightly coupled 54
  two-phase commit 54
troubleshooting
  online information 86
  tutorials 86
tutorials
  problem determination 86
  troubleshooting 86
  Visual Explain 86
Tuxedo
  configuring 74
TXSeries CICS 72
TXSeries Encina 72

U

UNIX
  application development
    environment variable settings 46
  configuring
    application development 46
unixODBC driver manager
  setting up 75
updates
  DB2 Information Center 84

V

versions
  IBM OLE DB Provider for DB2 24
Visual Explain
  tutorial 86

W

WebSphere Application Server
  overview 72
WebSphere MQ
  overview 72
Windows operating systems
  application development environment configuration 47
  installing
    DB2 servers (requirements) 12
    IBM data server clients (requirements) 12

X

X/Open Company 20
X/Open distributed transaction processing (DTP) model 54
X/Open SQL CLI 20
XA
  interface
    distributed transaction processing model 54

XA *(continued)*
  specification 66
  switch 66
XA transaction managers
  configuration considerations 68
  overview 41
  security considerations 65
  troubleshooting 68
  updating host and System i databases 62

Z

Zend Core
  supported development tools 43

Printed in USA

GC23-5856-02

Spine information:

DB2 Version 9.5 for Linux, UNIX, and Windows

Getting Started with Database Application Development