

IBM DB2 10.5
for Linux, UNIX, and Windows

*Getting Started with Database
Application Development*
Updated October, 2014



IBM DB2 10.5
for Linux, UNIX, and Windows

*Getting Started with Database
Application Development*
Updated October, 2014



Note

Before using this information and the product it supports, read the general information under Appendix C, "Notices," on page 123.

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 <http://www.ibm.com/shop/publications/order>
- To find your local IBM representative, go to the IBM Directory of Worldwide Contacts at <http://www.ibm.com/planetwide/>

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

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

© Copyright IBM Corporation 2006, 2014.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

About this book	v	Chapter 9. Supported drivers for JDBC and SQLJ	31
Part 1. The DB2 database application development environment.	1	Chapter 10. IBM OLE DB Provider for DB2	33
Part 2. Support for elements of the database application development environment	3	Chapter 11. node-ibm_db driver for Node.js applications	35
Part 3. Supported operating systems for database application development	5	Chapter 12. Perl DBI	37
Chapter 1. Additional installation considerations (AIX)	7	Chapter 13. PHP application development for IBM Database servers.	39
Chapter 2. Additional installation considerations (HP-UX)	9	Chapter 14. Python, SQLAlchemy and Django Framework application development for IBM Database servers.	41
Chapter 3. Additional installation considerations (Linux)	11	Chapter 15. The IBM_DB Ruby driver and Rails adapter	43
Chapter 4. Additional installation considerations (Solaris)	15	Part 6. Programming languages and compilers for database application development	45
Chapter 5. Additional installation considerations (Windows)	17	Chapter 16. Supported .NET development software	47
Part 4. IBM Data Server Client support for database application development	19	Chapter 17. Support for database application development in C	49
Part 5. Supported database application programming interfaces	21	Chapter 18. Support for database application development in C++	51
Chapter 6. ADO.NET development for IBM Data Servers	23	Chapter 19. Support for database application development in COBOL	53
IBM Data Server Provider for .NET	23	Chapter 20. Support for database application development in Fortran	55
ODBC .NET Data Provider	24	Chapter 21. Java software support for DB2 database products	57
OLE DB .NET Data Provider	25	Chapter 22. Support for database application development in Perl	61
Chapter 7. Introduction to DB2 Call Level Interface and ODBC	27	Chapter 23. Support for database application development in PHP.	63
Chapter 8. Introduction to embedded SQL	29		

Chapter 24. Support for database application development in Ruby/Ruby on Rails	65
Chapter 25. Support for database application development in REXX	67
<hr/>	
Part 7. Supported transaction managers	69
<hr/>	
Part 8. Supported database application development tools	71
<hr/>	
Part 9. Configuring the database application development environment.	73
<hr/>	
Chapter 26. Configuring the operating system for database application development	75
Setting up the application development environment (Linux and UNIX)	75
Linux and UNIX environment variable settings	76
Setting up the application development environment (Windows)	77
Chapter 27. Configuring the IBM Data Server Client for database application development	79
Chapter 28. Configuring the development environment for application programming interfaces	81
Configuring your development environment to build and run CLI and ODBC applications	82

Installing the IBM Data Server Driver for JDBC and SQLJ	82
---	----

Chapter 29. Transaction managers	87
X/Open distributed transaction processing model	87
xa_open string formats	91
Updating host or System i database servers with an XA-compliant transaction manager	96
Resolving indoubt transactions manually	96
Indoubt transaction management APIs	98
Security considerations for XA transaction managers	99
XA function supported by DB2 for Linux, UNIX, and Windows	100
XA interface problem determination	102
Configuration considerations for XA transaction managers.	103
Compiling the unixODBC driver manager. . . .	109
Installing the unixODBC driver manager	111

Part 10. Appendixes	113
--------------------------------------	------------

Appendix A. DB2 database product and packaging information	115
---	------------

Appendix B. DB2 technical information	117
DB2 technical library in hardcopy or PDF format	118
Displaying SQL state help from the command line processor	120
Accessing DB2 documentation online for different DB2 versions	120
Terms and conditions.	121

Appendix C. Notices	123
--------------------------------------	------------

Index	127
------------------------	------------

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.

Part 1. The DB2 database application development environment

The DB2[®] database application development environment is composed of several software elements.

These software elements include:

- **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.

The Data Server Clients periodically invoke the SYSIBM.DB2CONN_LIC stored procedure for licensing and configuration purposes.

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

Application development for DB2 databases is supported for a wide range of operating systems and database clients, along with numerous programming interfaces, languages, compilers, and tools.

The following documentation topics define the support for database application development on DB2 products:

- Part 3, “Supported operating systems for database application development,” on page 5
- Part 4, “IBM Data Server Client support for database application development,” on page 19
- Part 5, “Supported database application programming interfaces,” on page 21
- Part 6, “Programming languages and compilers for database application development,” on page 45
- Part 7, “Supported transaction managers,” on page 69
- Part 8, “Supported database application development tools,” on page 71

Part 3. Supported operating systems for database application development

Application development for DB2 databases is supported on a range of operating systems and architectures.

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.

Chapter 1. Additional installation considerations (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.

Important: For the most up-to-date installation requirements for DB2 database products, you must start using the System requirements for IBM DB2 for Linux, UNIX, and Windows and System requirements for IBM DB2 Connect™ technotes. These technotes use IBM Software Product Compatibility Reports (SPCR). With the SPCR tool, you can locate and find complete lists of supported operating systems, system requirements, prerequisites, and optional supported software for DB2 database products. This DB2 Information Centre topic might be removed in a future release or fix pack.

Note: If the minimum operating system requirements were applied using an operating system upgrade rather than using a new operating system installation, you must install I/O completion ports (IOCP) separately. You can obtain the IOCP libraries from the AIX installation CD and configure IOCP on the DB2 server. For a client only installation, IOCP is not required. For more information, see “configuring IOCP on AIX” in *Troubleshooting and Tuning Database Performance*.

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 website.
- (Clients only) If you plan to use Kerberos Authentication, you require IBM Network Authentication Service client v1.4 or later.
- Use the **bosboot** command to switch to the 64-bit kernel.

To switch to a 64-bit kernel, you require root authority and 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
```

- 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.
- Live Partition Mobility (LPM) is supported in partitioned database environments with multi physical node configurations.
- For details regarding known AIX issues, see www.ibm.com/support/docview.wss?uid=swg21165448

Mounting considerations

For root installations of DB2 products on UNIX operating systems, do not mount your file system with the *noatime* option.

Chapter 2. Additional installation considerations (HP-UX)

Before you install DB2 database products on HP-UX operating systems, ensure that the system you choose meets the necessary operating system, hardware, software, and communications requirements. The **db2prereqcheck** command checks whether your system meets the prerequisites for the installation.

Important: For the most up-to-date installation requirements for DB2 database products, you must start using the System requirements for IBM DB2 for Linux, UNIX, and Windows and System requirements for IBM DB2 Connect technotes. These technotes use IBM Software Product Compatibility Reports (SPCR). With the SPCR tool, you can locate and find complete lists of supported operating systems, system requirements, prerequisites, and optional supported software for DB2 database products. This DB2 Information Centre topic might be removed in a future release or fix pack.

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, modify some of them before you install the 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

- `libpam.so.0` (32-bit) is required for DB2 database servers to run 32-bit non-SQL routines.
- (Clients only) If you plan to use Kerberos authentication, your DB2 applications must be 64-bit.
- (Client and server) To use Kerberos authentication, install the HP `krb5` Kerberos support via the Kerberos packages on the operating system.
- 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.
- For details regarding known HP-UX issues, see www.ibm.com/support/docview.wss?uid=swg21257602

Note: DB2 products installed on the HP-UX operating system support long host names. The length has been extended to 255 bytes, in any combination of characters or digits.

To enable long host name support, complete the following tasks:

1. Turn on the kernel tunable parameter `expanded_node_host_name`.
`Kctune expanded_node_host_name=1`
2. Compile applications requiring long host name support with the `-D_HPUX_API_LEVEL=20040821` option.

Mounting considerations

For root installations of DB2 products on UNIX operating systems, do not mount your file system with the `nosetuid` option.

Chapter 3. Additional installation considerations (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. The **db2prereqcheck** command checks whether your system meets the prerequisites for the installation.

Important: For the most up-to-date installation requirements for DB2 database products, you must start using the System requirements for IBM DB2 for Linux, UNIX, and Windows and System requirements for IBM DB2 Connect technotes. These technotes use IBM Software Product Compatibility Reports (SPCR). With the SPCR tool, you can locate and find complete lists of supported operating systems, system requirements, prerequisites, and optional supported software for DB2 database products. This DB2 Information Centre topic might be removed in a future release or fix pack.

In addition to system requirements, installing a DB2 product on supported Linux operating systems has the following additional considerations:

- If you are planning to install the IBM DB2 pureScale® Feature, different installation prerequisites are required. See the Installing the DB2 pureScale Feature topic.
- Inspur K-UX does not support 32-bit applications.
- For the most up-to-date requirements for DB2 database products, see <http://www.ibm.com/support/docview.wss?uid=swg27038033>.

Multithreaded architecture limitations

If you are installing a DB2 32-bit database product on a Linux operating system, consider upgrading to a 64-bit operating system and installing the DB2 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 when you have sufficient system resources to support this configuration.

Distribution Requirements

You should update your kernel configuration parameters before installing a DB2 database product. 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 about setting and activating these parameters using the **sysctl** command.

Package requirements

Table 1 lists the package requirements for SLES and RHEL distributions:

- `libpam.so.0` (32-bit) is required for DB2 database servers to run 32-bit non-SQL routines and JDBC type 2 driver applications in 32-bit JDK.
- `libaio.so.1` is required for DB2 database servers using asynchronous I/O.
- `libstdc++.so.6.0.8` is required for DB2 database servers and clients.
- `libstdc++33` and `libstdc++43` are required for SLES 11.
- The **pdksh** Korn Shell package is required for all DB2 database systems.

Table 1. Package requirements for SLES and RHEL. Package requirements for SLES and RHEL

Package name	Description
libaio	Contains the asynchronous library required for DB2 database servers.
compat-libstdc++	Contains <code>libstdc++.so.6</code> (not required for Linux on POWER® or SLES 11)
pdksh	Korn Shell

Table 2 and Table 3 on page 13 list the package requirements for SUSE Linux and Red Hat distributions for DB2 partitioned database servers.

- The **pdksh** Korn Shell package is required for all DB2 database systems.
- A remote shell utility is required for partitioned database systems. DB2 database systems support the following remote shell utilities:
 - **rsh**
 - **ssh**

By default, DB2 database systems use **rsh** when executing commands on remote DB2 nodes, for example, when starting a remote DB2 database partition. To use the DB2 database system default, the `rsh-server` package must be installed (see following table). More information about **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 database system setup. For general Linux information, see your Linux distribution documentation.

Table 2. Package requirements for SUSE Linux. Package requirements for SUSE Linux

Package name	Description
pdksh	Korn Shell.

Table 2. Package requirements for SUSE Linux (continued). Package requirements for SUSE Linux

Package name	Description
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 database systems 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 database systems to use ssh .
nfs-utils	Network File System support package. It allows access to local files from remote computers.

Table 3. Package requirements for Red Hat. Package requirements for Red Hat

Directory	Package name	Description
/System Environment/Shell	pdcksh	Korn Shell.
/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 database systems 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 database systems 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 database systems to use ssh .
/System Environment/Daemons	nfs-utils	Network File System support package. It allows access to local files from remote computers.

Software considerations

- (Client and Server) To use Kerberos authentication, install the Linux krb5 Kerberos support via the operating system Kerberos packages.
- 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, or
 - You want to use any DB2 graphical tools on Linux for x86 and Linux on AMD 64/EM64T.

- Micro Focus does not offer support for any of its COBOL compiler products on SLES 11.
- If you plan on using transparent Lightweight Directory Access Protocol (LDAP) on RHEL 5, you must install `nss_ldap-253-35.el5` or later.

Security-enhanced Linux considerations

On RHEL 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.

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 database product installs successfully on an RHEL 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.

Chapter 4. Additional installation considerations (Solaris)

Before you install a DB2 database product on Solaris, ensure that your system meets the operating system, hardware, and software requirements. The **db2prereqcheck** command checks whether your system meets the prerequisites for the installation.

Important: For the most up-to-date installation requirements for DB2 database products, you must start using the System requirements for IBM DB2 for Linux, UNIX, and Windows and System requirements for IBM DB2 Connect technotes. These technotes use IBM Software Product Compatibility Reports (SPCR). With the SPCR tool, you can locate and find complete lists of supported operating systems, system requirements, prerequisites, and optional supported software for DB2 database products. This DB2 Information Centre topic might be removed in a future release or fix pack.

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

- `libpam.so.0` (32-bit) is required for DB2 database servers to run 32-bit non-SQL routines.
- (Client and server) To use Kerberos authentication, install the `krb5` Kerberos support via the available Kerberos packages.
- 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
- For details regarding known Solaris issues, see www.ibm.com/support/docview.wss?uid=swg21257606

Security patches can be obtained from the <http://java.sun.com> website. From the SunSolve online website, click **Patches** 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://java.sun.com> website.

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

DB2 database products support the following Solaris concepts:

- Solaris Logical Domains (LDoms)
- Solaris Zones
- ZFS filesystems

For details about virtualization technology supported by DB2 products, see <https://www.ibm.com/developerworks/community/wikis/home?lang=en-us#!/>

wiki/Information%20Management/page/Virtualization%20Support.

Mounting considerations

For root installations of DB2 products on UNIX operating systems, do not mount your file system with the *noatime* option.

Chapter 5. Additional installation considerations (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. The **db2prereqcheck** command checks whether your system meets the prerequisites for the installation.

Important: For the most up-to-date installation requirements for DB2 database products, you must start using the System requirements for IBM DB2 for Linux, UNIX, and Windows and System requirements for IBM DB2 Connect technotes. These technotes use IBM Software Product Compatibility Reports (SPCR). With the SPCR tool, you can locate and find complete lists of supported operating systems, system requirements, prerequisites, and optional supported software for DB2 database products. This DB2 Information Centre topic might be removed in a future release or fix pack.

In addition to system requirements, installing a DB2 product on supported Windows operating systems has these additional considerations.

- Windows Installer 3.0 is required. It is installed by the installer if it is not detected.
- IBM Data Server Provider for .NET client applications and CLR server-side procedures require .NET 2.0 or later framework runtime. In an x64 environment, 32-bit IBM data server provider for .NET applications runs in the WOW64 emulation mode.
- If you plan to use LDAP (Lightweight Directory Access Protocol), 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 database products). Before installation of the Microsoft Active Directory, you must 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.

Note: DB2 database products support the hardware-enforced Data Execution Prevention (DEP) feature that is built into some Windows operating systems.

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

The Data Server Client supports application development for databases.

The Data Server Client includes the following features:

- **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 Line Processor (CLP) to prototype SQL statements or to perform ad hoc queries against the database.
- **A set of documented APIs** for other application development tools to implement precompiler support for DB2 directly within their products. For example, IBM COBOL on AIX uses this interface.

Part 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).
- 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 is 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.
- Use Oracle Call Interface (OCI) applications to access DB2 databases. The IBM Data Server Driver for DB2CI provides support for calls to a number of OCI APIs.

Chapter 6. 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 database server support for the ADO.NET interface. The provider delivers high-performing, secure access to IBM data servers.

The IBM Data Server Provider for .NET is a name that is used to describe the .NET providers that are packaged with the IBM data server clients products. There are two .NET providers included in the IBM data server clients or IBM Data Server Driver Package. The two .NET providers are also called the Common .NET Providers.

The DB2 .NET provider

You can use the DB2 .NET provider to access all supported DB2 database servers and Informix database servers. To connect to DB2 for z/OS® and IBM DB2 for IBM i servers, you require the DB2 Connect Server license.

The dynamic-link library file for the DB2 .NET provider is `IBM.Data.DB2.dll`.

For information about supported DB2 and Informix database servers, see the detailed system requirements for a specific product site (<http://pic.dhe.ibm.com/infocenter/prodguid/v1r0/clarity/softwareReqsForProduct.html>).

The Informix .NET provider

Important: The Informix .NET provider (`IBM.Data.Informix.dll`) is deprecated since DB2 Version 10.1 Fix Pack 1 and might be discontinued in a later release. Start using the DB2 .NET provider (`IBM.Data.DB2.dll`) to connect to Informix database servers.

You can use the Informix .NET provider to access supported Informix servers.

The dynamic-link library file for the Informix .NET provider is `IBM.Data.Informix.dll`.

For information about supported Informix database servers, see the detailed system requirements for a specific product site (<http://pic.dhe.ibm.com/infocenter/prodguid/v1r0/clarity/softwareReqsForProduct.html>).

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

You can also use the IBM Database Add-Ins for Visual Studio software to help quickly and easily develop .NET applications for IBM data servers with Microsoft Visual Studio. You can use the IBM Database Add-Ins for Visual Studio software 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 CLI Driver. Therefore, the connection string keywords supported by the ODBC .NET Data Provider are the same as those supported by the CLI driver. This provider is no longer tested. Users are recommended to use the IBM Data Server Provider for .NET.

Also, the ODBC .NET Data Provider has the same restrictions as the CLI driver. There are additional restrictions for the ODBC .NET Data Provider, which are identified in the topic: “ODBC .NET Data Provider restrictions” in *Developing ADO.NET and OLE DB Applications*.

In order to use the ODBC .NET Data Provider, you must have the .NET Framework Version 2.0, 3.0, or 3.5 installed. For DB2 Universal Database™ for AS/400® V5R4 and earlier, the following fix is required on the server: APAR II13348.

The supported connection keywords for the ODBC .NET Data Provider are listed in the table 1:

*Table 4. 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 code 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. This provider is no longer tested. Users are recommended to use the IBM Data Server Provider for .NET.

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 *Developing ADO.NET and OLE DB Applications*.

In order to use the OLE DB .NET Data Provider, you must have the .NET Framework Version 2.0, 3.0, or 3.5 installed.

For DB2 Universal Database for AS/400 R520, R530 and R540, the following fix is required on the server: APAR ii13348.

All the supported connection keywords for the OLE DB .NET Data Provider are shown in table 1:

Table 5. 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.

Example of creating an `OleDbConnection` to connect to the SAMPLE database is:

[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()
```

Chapter 7. Introduction to DB2 Call Level Interface and ODBC

DB2 Call Level Interface (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.

CLI is an alternative to embedded dynamic SQL, but unlike embedded SQL, it does not require host variables or a precompiler. Applications can be run against a variety of databases without having to be compiled against each of these databases. Applications use procedure calls at run time to connect to databases, issue SQL statements, and retrieve data and status information.

The CLI interface provides many features not available in embedded SQL. For example:

- CLI provides function calls that support a way of querying database catalogs that is consistent across the DB2 family. This reduces the need to write catalog queries that must be tailored to specific database servers.
- CLI provides the ability to scroll through a cursor:
 - Forward by one or more rows
 - Backward by one or more rows
 - Forward from the first row by one or more rows
 - Backward from the last row by one or more rows
 - From a previously stored location in the cursor.
- Stored procedures called from application programs that were written using CLI can return result sets to those programs.

CLI 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 CLI driver also acts as an ODBC driver when loaded by an ODBC driver manager. It conforms to ODBC 3.51.

CLI Background information

To understand CLI 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. Because 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 through a mechanism called an escape clause.

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

The CLI 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/downloads>. 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 about DB2 specific extensions and diagnostic information), in conjunction with the ODBC Programmer's Reference and SDK Guide available from Microsoft.

Applications written using CLI APIs link directly to the CLI library. CLI 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 CLI 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)

Chapter 8. Introduction to embedded SQL

Embedded SQL applications connect to databases and execute embedded SQL statements. The embedded SQL statements are contained in a package that must be bound to the target database server.

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

Building embedded SQL applications involves two prerequisite steps before 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.

Related information:

 Installing and configuring Optim Performance Manager Extended Insight

Chapter 9. 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 that 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 data server using a data-server-independent protocol. The data server 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 for Linux, UNIX, and Windows supports the following driver:

Driver name	Packaged as	Driver type
IBM Data Server Driver for JDBC and SQLJ	<ul style="list-style-type: none">db2jcc.jar and sqlj.zip for JDBC 3.0 supportdb2jcc4.jar and sqlj4.zip for support of some JDBC 4.0 or later 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 3.5x is JDBC 3.0-compliant. IBM Data Server Driver for JDBC and SQLJ version 4.x is compliant with JDBC 4.0 or later.

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

- Version 3.5x supports all of the methods that are described in the JDBC 3.0 specifications.
- Version 4.x supports all of the methods that are described in the JDBC 4.0 or later specifications.
- 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 data server from Java user-defined functions and stored procedures use IBM Data Server Driver for JDBC and SQLJ type 2 connectivity only. Applications that call user-defined functions or stored procedures can use IBM Data Server Driver for JDBC and SQLJ type 2 connectivity or IBM Data Server Driver for JDBC and SQLJ type 4 connectivity to connect to a data server. 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>).

Chapter 10. 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 z/OS, DB2 Server for VM and VSE, or DB2 Universal Database for AS/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 about 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 previously. You also need to install a full DB2 product, IBM Data Server Driver for ODBC and CLI, or IBM Data Server Driver Package.

Chapter 11. node-ibm_db driver for Node.js applications

You can use the node-ibm_db driver in your Node.js applications to access IBM database servers.

Node.js is a software platform that is built on JavaScript. Node.js provides a fast, scalable, lightweight application solution for data-intensive real-time applications.

The node-ibm_db driver is a Node.js binding for IBM database servers. The node-ibm_db driver contains both asynchronous and synchronous interfaces.

You can install the node-ibm_db driver with the following IBM data server client products:

- All supported versions, releases, and fix packs of the IBM Data Server Driver Package product
- All Version 10.5 Fix Pack 4 or later IBM data server products

Chapter 12. Perl DBI

IBM Data Server product supports the Perl database Interface (DBI) specification for data access through the DBD:DB2 driver.

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 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.

For details about the DB2 Perl DBI module, the DBD:DB2 driver, see <http://www.ibm.com/software/data/db2/perl>/<http://www.ibm.com/software/data/db2/perl/>.

Chapter 13. PHP application development for IBM Database servers

PHP: Hypertext Preprocessor (PHP) is an interpreted programming language that is widely used for developing web applications. PHP is a popular language for web development because it is easy to learn, focuses on practical solutions, and supports the most commonly required functionality in web applications.

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 website.

IBM provides the following PHP extensions for accessing IBM Database servers:

ibm_db2

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

A driver for the PHP Data Objects (PDO) extension that offers access to IBM Database servers through the standard object-oriented database interface that is introduced in PHP 5.1.

The most recent versions of the `ibm_db2` and `pdo_ibm` extensions are also available from the PHP Extension Community Library (PECL) at <http://pecl.php.net/>.

Chapter 14. Python, SQLAlchemy and Django Framework application development for IBM Database servers

Python is a general purpose, high-level scripting language that is well suited for rapid application development. Python emphasizes code readability and supports various programming paradigms, including procedural, object-oriented, aspect-oriented, meta, and functional programming. The Python language is managed by the Python Software Foundation.

The listed extensions are available for accessing IBM Database servers from a Python application:

ibm_db

This API is defined by IBM and provides the best support for advanced features. In addition to issuing SQL queries, calling stored procedures, and using pureXML[®], you can access metadata information.

ibm_db_dbi

This API implements Python Database API Specification v2.0. Because the `ibm_db_dbi` API conforms to the specification, it does not offer some of the advanced features that the `ibm_db` API supports. If you have an application with a driver that supports Python Database API Specification v2.0, you can easily switch to `ibm_db`. The `ibm_db` and `ibm_db_dbi` APIs are packaged together.

ibm_db_sa

This adapter supports SQLAlchemy, which offers a flexible way to access IBM Database servers. SQLAlchemy is a popular open source Python SQL toolkit and object-to-relational mapper (ORM).

ibm_db_django

This adapter provides access to IBM Database servers from Django. Django is a popular web framework used to build high-performing, elegant web applications quickly.

If you want to connect your Python applications to DB2 for i V5R4 and later servers, you must have PTF SI27256 applied to those servers.

Chapter 15. The IBM_DB Ruby driver and Rails adapter

Collectively known as the IBM_DB gem, the IBM_DB Ruby driver, and Rails adapter allows Ruby applications to access the IBM database servers.

Ruby applications that are connecting to DB2 for z/OS servers and DB2 for i servers requires the use of the DB2 Connect license.

The IBM_DB Ruby adapter allows any database-backed Rails application to interface with IBM data servers.

For more information about IBM Ruby projects and the RubyForge open source community, see <http://rubyforge.org/projects/rubyibm/>

For a list of installation requirements for DB2 database products, see http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.qb.server.doc/doc/r0025127.html

For a list of installation requirements for IBM Informix server, see http://www-01.ibm.com/support/knowledgecenter/SSGU8G_11.50.0/com.ibm.expr.doc/ids_in_004x.htm

For information about downloading an IBM Data Server Driver Package, see <http://www.ibm.com/software/data/support/data-server-clients/index.html>.

Part 6. Programming languages and compilers for database application development

Application development for DB2 databases is supported on a range of programming languages and compilers.

You can develop DB2 database applications in the following languages:

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

The APIs and programming languages for developing external routines are generally compatible with those for database application development. 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*.

Chapter 16. Supported .NET development software

IBM Data Server Provider for .NET supports the following .NET Framework and Visual Studio versions.

Supported development software for .NET Framework applications

In addition to an IBM data server client or driver package, you need one of the supported tools to develop .NET Framework applications:

- Visual Studio 2008
- Visual Studio 2010
- Visual Studio 2012

Supported deployment software for .NET Framework applications

In addition to an IBM data server client or driver package, you need one of the following packages to deploy .NET Framework applications. In most cases, a .NET Framework redistributable package is included with a Windows installation.

- .NET Framework Version 2.0 Redistributable Package
- .NET Framework Version 3.0 Redistributable Package
- .NET Framework Version 3.5 Redistributable Package
- .NET Framework Version 4.0 Redistributable Package
- .NET Framework Version 4.5 Redistributable Package

When you install a 64-bit IBM Data Server Package, both 32-bit and 64-bit providers are installed and configured.

If a .NET Framework is not installed, the IBM Data Server Client and driver installer will not install the IBM Data Server Provider for .NET. You must install the IBM Data Server Provider for .NET manually.

Chapter 17. Support for database application development in C

Application development for DB2 databases is supported on a range of C language compilers.

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 6. Supported C compilers

Operating System	Supported Compiler(s)
AIX	<ul style="list-style-type: none">• IBM XL C/C++ Enterprise Edition Version 10.1 for AIX• IBM XL C/C++ Enterprise Edition Version 11.0 for AIX• IBM XL C/C++ Enterprise Edition Version 12.0 for AIX
HP-UX on IPF (IA64)	<ul style="list-style-type: none">• For HP-UX 11i v3, use HP aC++ Compiler Version A.06.12
Linux on x86	<ul style="list-style-type: none">• GNU/Linux gcc versions 3.3 and 3.4 and higher• GNU/Linux gcc version 4.1 and higher• Intel C Compiler Version 10.1
Linux on x64	<ul style="list-style-type: none">• GNU/Linux gcc versions 3.3 and 3.4 and higher• GNU/Linux gcc version 4.1 and higher• Intel C Compiler Version 10.1
Linux on POWER (PowerPC®, Power Systems™ Servers, pSeries)	<ul style="list-style-type: none">• GNU/Linux gcc versions 3.3 and 3.4 and higher• GNU/Linux gcc version 4.1 and higher• IBM XL C/C++ Advanced Edition Version 9.0 for Linux• IBM XL C/C++ Advanced Edition Version 10.1 for Linux• IBM XL C/C++ Advanced Edition Version 11.0 for Linux
Linux on System z® (s/390x)	<ul style="list-style-type: none">• GNU/Linux gcc versions 3.3 and 3.4 and higher• GNU/Linux gcc version 4.1 and higher
Solaris on UltraSPARC	<ul style="list-style-type: none">• Sun Studio 10• Sun Studio 11• Sun Studio 12• Forte C++ Version 6.2• Sun ONE Studio 7, Compiler Collection
Solaris on x64	<ul style="list-style-type: none">• Sun Studio 10• Sun Studio 11• Sun Studio 12
Windows on x86	<ul style="list-style-type: none">• Intel Proton Compiler for Windows 32-bit applications, Version 9.0.021 or later• Microsoft Visual C++ .NET or later

Table 6. Supported C compilers (continued)

Operating System	Supported Compiler(s)
Windows on x64 (x64, AMD64/EM64T)	<ul style="list-style-type: none"> • 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++ 2008, 2010

Chapter 18. Support for database application development in C++

Application development for DB2 databases is supported on a range of C++ language compilers.

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 7. Supported C++ compilers

Operating System	Supported Compiler(s)
AIX	<ul style="list-style-type: none">• IBM XL C/C++ Enterprise Edition Version 10.1 for AIX• IBM XL C/C++ Enterprise Edition Version 11.0 for AIX• IBM XL C/C++ Enterprise Edition Version 12.0 for AIX
HP-UX on IPF (IA64)	<ul style="list-style-type: none">• For HP-UX 11i v3, use HP aC++ Compiler Version A.06.12
Linux on x86	<ul style="list-style-type: none">• GNU/Linux g++ version 4.1.1 and 4.1.0 on RHEL5 and SLES10• Intel C++ Compiler Version 9.1 and later <p>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.</p>
Linux on x64	<ul style="list-style-type: none">• GNU/Linux g++ versions 4.1.1 and 4.1.0 on RHEL5 and SLES10 <p>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.</p>
Linux on POWER (PowerPC, Power Systems Servers, pSeries)	<ul style="list-style-type: none">• GNU/Linux g++ version 4.1.1 and 4.1.0 on RHEL5 and SLES10• IBM XL C/C++ Enterprise Edition Version 9.0 for Linux on pSeries <p>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.</p>
Linux on zSeries (s/390x)	<ul style="list-style-type: none">• GNU/Linux g++ versions 3.4 and 3.3 on SLES9• GNU/Linux g++ version 4.1.1 and 4.1.0 on RHEL5 and SLES10 <p>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.</p>

Table 7. Supported C++ compilers (continued)

Operating System	Supported Compiler(s)
Solaris on UltraSPARC	<ul style="list-style-type: none"> • Sun Studio 10 • Sun Studio 11 • Sun Studio 12 • Forte C++ Version 6.2 • Sun ONE Studio 7, Compiler Collection
Solaris on x64	<ul style="list-style-type: none"> • Sun Studio 10 • Sun Studio 11 • Sun Studio 12
Windows on x86	<ul style="list-style-type: none"> • Intel Proton Compiler for Windows 32-bit applications, Version 9.0.021 or later • Microsoft Visual C++ .NET
Windows on x64	<ul style="list-style-type: none"> • Microsoft Windows Server 2003 SP1 SDK (Build 1830 or later), C/C++ compiler for x64 • Intel Proton Compiler for Windows x64, Version 9.0.024 or later • Microsoft Visual C++ 2008, 2010

Chapter 19. Support for database application development in COBOL

This reference table outlines the current level of compiler support provided by DB2 for application development with the COBOL programming language.

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 8. Supported COBOL compilers

Operating system	Supported compilers
AIX	<ul style="list-style-type: none">• IBM COBOL for AIX Version 3.1• IBM COBOL for AIX Version 4.1• Micro Focus Server Express®, Version 5.1• Micro Focus Server Express, Version 6.0
HP-UX on IPF (IA64)	<ul style="list-style-type: none">• Micro Focus Server Express, Version 5.1• Micro Focus Server Express, Version 6.0
Linux on x86	<ul style="list-style-type: none">• Micro Focus Server Express, Version 5.1• Micro Focus Server Express, Version 6.0
Linux on x64	<ul style="list-style-type: none">• Micro Focus Server Express, Version 5.1• Micro Focus Server Express, Version 6.0
Linux on zSeries (s/390x)	<ul style="list-style-type: none">• Micro Focus Server Express, Version 5.1• Micro Focus Server Express, Version 6.0
Solaris on UltraSPARC	<ul style="list-style-type: none">• Micro Focus Server Express, Version 5.1• Micro Focus Server Express, Version 6.0
Windows on x86	<ul style="list-style-type: none">• Micro Focus Net Express, Version 5.1• Micro Focus Server Express, Version 6.0• IBM Rational® Developer for zSeries v7.6.0
Windows on x64	<ul style="list-style-type: none">• Micro Focus Net Express, Version 5.1• Micro Focus Server Express, Version 6.0

Chapter 20. Support for database application development in Fortran

Application development for DB2 databases is supported on a range of Fortran language compilers.

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 9. Supported Fortran compilers

Operating System	Supported Compiler(s)
AIX	<ul style="list-style-type: none">• IBM XL Fortran for AIX Version 12.1
HP-UX on IPF (IA64)	<ul style="list-style-type: none">• HP-UX F90 B.11.23
Solaris on UltraSPARC	<ul style="list-style-type: none">• SPARCompiler Fortran Versions 4.2 and 5.0 (32-bit only)

Chapter 21. Java software support for DB2 database products

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

Important: For the most up-to-date installation requirements for DB2 database products, you must start using the System requirements for IBM DB2 for Linux, UNIX, and Windows and System requirements for IBM DB2 Connect technotes. These technotes use IBM Software Product Compatibility Reports (SPCR). With the SPCR tool, you can locate and find complete lists of supported operating systems, system requirements, prerequisites, and optional supported software for DB2 database products. This DB2 Information Centre topic might be removed in a future release or fix pack.

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 IBM Data Server Runtime Client or IBM Data Server Driver Package.

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

Operating System Platform	SDK for Java level
AIX	SDK 7
HP-UX for Itanium-based systems	SDK 7
Linux on x86	SDK 7
Linux on AMD64/EM64T	SDK 7
Linux on zSeries	SDK 7
Linux on POWER	SDK 7
Sun SPARC x64	SDK 7
Sun Solaris x64	SDK 7
Windows x86	SDK 7
Windows x64	SDK 7

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 later in this section entitled DB2 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.

Versions of SDK for Java, other than IBM SDK, are supported only for building and running stand-alone Java applications. For building and running new Java stored procedures and user-defined functions, only the IBM SDK for Java that is included with the DB2 for Linux, UNIX, and Windows product is supported. For running Java stored procedures and user-defined functions that were built by prior DB2 releases, refer to Table 1, column "Java Stored Procedures and User Defined Functions" for details.

Table 10. DB2 for Linux, UNIX, and Windows supported levels of SDKs for Java

	Java applications that use JDBC 3.0 or earlier	Java applications that use JDBC 4.0 or earlier and JDBC 3.0 or earlier ⁷	Java Stored Procedures and User Defined Functions	DB2 Graphical Tools
AIX	1.4.2 to 7	6 and 7	1.4.2 ⁶ to 7 ⁵	N/A
HP-UX for Itanium-based systems	1.4.2 to 7 ¹	6 and 7 ¹	1.4.2 ⁶ to 7	N/A
Linux on POWER	1.4.2 to 7 ^{3,4}	6 and 7 ^{3,4}	1.4.2 ⁶ to 7	N/A
Linux on x86	1.4.2 to 7 ^{2,3,4}	6 and 7 ^{2,3,4}	1.4.2 ⁶ to 7	5 to 7
Linux on AMD64 and Intel EM64T processors	1.4.2 to 7 ^{2,3,4}	6 and 7 ^{2,3,4}	1.4.2 ⁶ to 7	N/A
Linux on zSeries	1.4.2 to 7 ^{3,4}	6 and 7 ^{3,4}	1.4.2 ⁶ to 7	N/A
Sun SPARC 64	1.4.2 to 7 ²	6 and 7 ²	1.4.2 ⁶ to 7	N/A
Solaris x64	1.4.2 to 7 ²	6 and 7 ²	1.4.2 ⁶ to 7	N/A
Windows on x86	1.4.2 to 7 ²	6 and 7 ²	1.4.2 ⁶ to 7	5 to 7
Windows on x64, for AMD64 and Intel EM64T processors	1.4.2 to 7 ²	6 and 7 ²	1.4.2 ⁶ to 7	5 to 7

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 Oracle are supported for building and running stand-alone applications with the IBM Data Server Driver for JDBC and SQLJ. However, if you set the IBM Data Server Driver for JDBC and SQLJ property `securityMechanism` for a type of security that uses encryption, the SDK for Java must support the type of encryption that you use. For example, the SDK for Java that you use might support 256-bit AES (strong) encryption, but not 56-bit DES (weak) encryption. You can specify the encryption algorithm by setting the IBM Data Server Driver for JDBC and SQLJ property `encryptionAlgorithm`. To use 256-bit AES encryption, set `encryptionAlgorithm` to 2. When you use 256-bit AES encryption with the SDK for Java from Oracle, you might need to install the JCE Unlimited Strength Jurisdiction Policy File, which is available from Oracle.
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. If SDK for Java 6 SR2 or later is used, set `DB2LIBPATH=java_home/jre/lib/ppc64`.
6. Support for Java stored procedures and user-defined functions built by IBM SDK for Java 1.4.2 was deprecated in Version 9.7 and might be removed in a future release. IBM SDK for Java 1.4.2 has an End of Service date of September 2011. It is recommended to remove SDK for Java 1.4.2 dependency well before this date. Removing this dependency can be done by rebuilding Java stored procedures and user-defined functions with the SDK for Java included in DB2 Version 9.1, DB2 Version 9.5, DB2 Version 9.7 or DB2 V10.1 .
7. Java 6 is sufficient if you need to use JDBC 4.0 functions only. Java 7 is required if you need to use JDBC 4.1 functions.

Chapter 22. Support for database application development in Perl

Application development for DB2 databases is supported for a range of Perl language versions.

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 11. Supported versions of the Perl programming language

Operating System	Supported Version(s)
AIX	<ul style="list-style-type: none">• Perl 5.8 or later• DBI 1.41 or later
HP-UX on IPF (IA64)	<ul style="list-style-type: none">• Perl 5.8 or later• DBI 1.41 or later
Linux on x86	<ul style="list-style-type: none">• Perl 5.8 or later• DBI 1.41 or later
Linux on x64	<ul style="list-style-type: none">• Perl 5.8 or later• DBI 1.41 or later
Linux on POWER (PowerPC, Power Systems Servers, pSeries)	<ul style="list-style-type: none">• Perl 5.8 or later• DBI 1.41 or later
Linux on zSeries (s/390x)	<ul style="list-style-type: none">• Perl 5.8 or later• DBI 1.41 or later
Solaris on UltraSPARC	<ul style="list-style-type: none">• Perl 5.8 or later• DBI 1.41 or later
Solaris on x64	<ul style="list-style-type: none">• Perl 5.8 or later• DBI 1.41 or later
Windows on x86	<ul style="list-style-type: none">• Perl 5.8 or later• DBI 1.41 or later
Windows on x64	<ul style="list-style-type: none">• Perl 5.8 or later• DBI 1.41 or later

Chapter 23. Support for database application development in PHP

Application development for DB2 databases is supported for a range of PHP language versions.

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 following table, 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 12. Supported versions of the PHP scripting language

Operating System	Supported Version(s)
AIX	<ul style="list-style-type: none">• PHP 4.3.4 or later for IBM_DB2 driver• PHP 5.1 or later for PDO_IBM driver
HP-UX on IPF (IA64)	<ul style="list-style-type: none">• PHP 4.3.4 or later for IBM_DB2 driver• PHP 5.1 or later for PDO_IBM driver
Linux on x86 (32-bit DB2 instances and language only)	<ul style="list-style-type: none">• PHP 4.3.4 or later for IBM_DB2 driver• PHP 5.1 or later for PDO_IBM driver
Linux on AMD64/EM64T	<ul style="list-style-type: none">• PHP 4.3.4 or later for IBM_DB2 driver• PHP 5.1 or later for PDO_IBM driver
Linux on POWER (PowerPC, System i [®] , pSeries)	<ul style="list-style-type: none">• PHP 4.3.4 or later for IBM_DB2 driver• PHP 5.1 or later for PDO_IBM driver
Linux on zSeries (s/390x)	<ul style="list-style-type: none">• PHP 4.3.4 or later for IBM_DB2 driver• PHP 5.1 or later for PDO_IBM driver
Solaris on UltraSPARC	<ul style="list-style-type: none">• PHP 4.3.4 or later for IBM_DB2 driver• PHP 5.1 or later for PDO_IBM driver
Solaris on x64	<ul style="list-style-type: none">• PHP 4.3.4 or later for IBM_DB2 driver• PHP 5.1 or later for PDO_IBM driver
Windows on x86 (32-bit DB2 instances only)	<ul style="list-style-type: none">• PHP 4.3.4 or later for IBM_DB2 driver• PHP 5.1 or later for PDO_IBM driver
Windows on x64 (AMD64/EM64T)	<ul style="list-style-type: none">• PHP 4.3.4 or later for IBM_DB2 driver• PHP 5.1 or later for PDO_IBM driver

Chapter 24. Support for database application development in Ruby/Ruby on Rails

Application development for DB2 databases is supported for a range of Ruby and Ruby on Rails language versions.

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 13. Supported versions for Ruby/Ruby on Rails

Operating System	Supported Versions
AIX	<ul style="list-style-type: none">• Ruby 1.8.5 and later• Ruby on Rails 1.2.1 and later.
HP-UX on IPF (IA64)	<ul style="list-style-type: none">• Ruby 1.8.5 and later• Ruby on Rails 1.2.1 and later.
Linux on x86 (32-bit DB2 instances and language only)	<ul style="list-style-type: none">• Ruby 1.8.5 and later• Ruby on Rails 1.2.1 and later.
Linux on AMD64/EM64T (x64)	<ul style="list-style-type: none">• Ruby 1.8.5 and later• Ruby on Rails 1.2.1 and later.
Linux on POWER (PowerPC, System i, pSeries)	<ul style="list-style-type: none">• Ruby 1.8.5 and later• Ruby on Rails 1.2.1 and later.
Linux on zSeries (s/390x)	<ul style="list-style-type: none">• Ruby 1.8.5 and later• Ruby on Rails 1.2.1 and later.
Solaris on UltraSPARC	<ul style="list-style-type: none">• Ruby 1.8.5 and later• Ruby on Rails 1.2.1 and later.
Solaris on x64	<ul style="list-style-type: none">• Ruby 1.8.5 and later• Ruby on Rails 1.2.1 and later.
Windows on x86 (32-bit application development only)	<ul style="list-style-type: none">• Ruby 1.8.5 and later• Ruby on Rails 1.2.1 and later.
Windows on x64 (AMD64/EM64T, 32-bit application development only)	<ul style="list-style-type: none">• Ruby 1.8.5 and later• Ruby on Rails 1.2.1 and later.

Chapter 25. Support for database application development in REXX

Application development for DB2 databases is supported for a range of REXX programming language versions.

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 14. Supported versions of the REXX programming language

Operating System	Supported Version(s)
AIX ¹	<ul style="list-style-type: none">• REXXSAA 4.00
Linux on x86 (32-bit DB2 instances and compilers only) ¹	<ul style="list-style-type: none">• Object REXX Interpreter for Linux Version 2.1
Linux on zSeries (s/390x) ¹	<ul style="list-style-type: none">• Object REXX 2.2.0 for Linux/390

- ¹Only 32-bit versions of REXX are supported.

Part 7. Supported transaction managers

A DB2 database 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 DB2 databases:

- IBM TXSeries™ CICS®
- IBMWebSphere MQ
- BEA Tuxedo
- BEA WebLogic
- Microsoft Distributed Transaction Coordinator (DTC)

Part 8. Supported database application development tools

These integrated development environments and 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)
- InfoSphere® Data Architect (separately available IBM product)

Database application plug-ins for integrated development environments

- IBM Database Add-Ins for Visual Studio

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

- Command line processor (CLP)
- IBM Data Studio

Tools for database application monitoring and performance tuning

- Snapshot and Event Monitors
- Static Profiling for CLI, ODBC, and JDBC applications
- IBM Data Studio (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 Studio - 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

Developer Edition

The 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.

Part 9. Configuring the database application development environment

To configure the database application development environment, you must configure the operating system, Data Server Client, development environment, and transaction manager.

Before you begin

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

- A supported operating system
See Part 3, “Supported operating systems for database application development,” on page 5.
- The IBM Data Server Client installed
See “Methods for installing IBM data server clients” in *Installing IBM Data Server Clients*.
- API drivers and, if required, driver managers installed
See Part 5, “Supported database application programming interfaces,” on page 21.
- Compilers or interpreters required for the programming languages you will be using installed
See Part 6, “Programming languages and compilers for database application development,” on page 45.
- A transaction manager installed
See Part 7, “Supported transaction managers,” on page 69.
- Development tools installed
See Part 8, “Supported database application development tools,” on page 71.

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 steps:

1. Configure the operating system.
See Chapter 26, “Configuring the operating system for database application development,” on page 75.

2. Configure the Data Server Client.
See Chapter 27, “Configuring the IBM Data Server Client for database application development,” on page 79.
3. Configure the development environment.
See Chapter 28, “Configuring the development environment for application programming interfaces,” on page 81.
4. Configure the transaction manager.
See “Configuration considerations for XA transaction managers” on page 103.

Chapter 26. Configuring the operating system for database application development

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

Before you begin

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

- A supported operating system
See Part 3, “Supported operating systems for database application development,” on page 5.
- A IBM Data Server Client installed
See “Methods for installing IBM data server clients” in *Installing IBM Data Server Clients*.

Procedure

To configure environment variables for DB2 database application development:

Perform one of the following tasks:

- For UNIX operating systems, “Setting up the application development environment (Linux and UNIX).”
- For Windows operating systems, “Setting up the application development environment (Windows)” on page 77

Setting up the application development environment (Linux and UNIX)

You must set environment variables for database manager instances on Linux and UNIX operating systems. This can be accomplished by launching the scripts that are provided for each instance.

About this task

Each database manager instance has two scripts to set up the environment for that instance: `db2profile` and `db2cshrc`. Setting up your environment is not required when using type 4 connectivity in your JDBC applications.

Procedure

1. Run the correct script to set up your environment for the instance that you will use to develop your application.
 - For bash or Korn shell:
`. $HOME/sql1lib/db2profile`
 - For C shell:
`source $HOME/sql1lib/db2cshrc`
2. If you are developing applications using ODBC, DB2 CLI, JDBC, or SQLJ, perform the following tasks:
 - “Setting up the ODBC environment” in *Call Level Interface Guide and Reference Volume 1*.

- “Installing the IBM Data Server Driver for JDBC and SQLJ” on page 82.

Linux and UNIX environment variable settings

Depending on the operating system, values for the environment variables are set, either in the db2profile file (for the Bash or Korn shell) or the db2cshrc file (for the C shell).

Calls to db2profile or db2cshrc files are put in the instance owner's .bashrc (Bash shell) file and either in the .profile (Bash or Korn shell) file or in the .login (C shell) file.

AIX operating systems:

- **PATH**, includes several DB2 directories including sqllib/bin
- **LIBPATH**, includes the sqllib/lib directory

HP-UX operating systems:

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

Linux and Solaris operating systems:

- **PATH**, includes several DB2 directories including sqllib/bin
- **LD_LIBRARY_PATH**, includes the sqllib/lib directory

Empty sqllib/userprofile and sqllib/usercshrc files are created during instance creation. You can place your own instance environment settings into these files. If you do not want the new environment settings in the db2profile or db2cshrc script, you can override them using the corresponding user script, which is called at the end of the db2profile or db2cshrc script. During an instance upgrade, which you perform by using the **db2iupgrade** command, the user scripts are copied so that your environment modifications are still in use.

You can generate the db2profile file (for the Born and Korn shell) and the db2cshrc file (for the C shell) from a IBM Data Server Driver Package or IBM Data Server Driver for ODBC and CLI installation by using the **installDSDriver** command. Running the db2profile or db2cshrc shell script file sets the environment variables as shown in the following list:

- Generic PATH and library path settings.

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

 Setting library path on AIX operating systems:

```
export LIBPATH=INSTALL_PATH/lib:$LIBPATH
```

 Setting library path on HP-UX operating systems:

```
export SHLIB_PATH=INSTALL_PATH/lib:$SHLIB_PATH
```

 Setting the library path on other UNIX operating systems and Linux operating systems:

```
export LD_LIBRARY_PATH=INSTALL_PATH/lib:$LD_LIBRARY_PATH
```
- Environment variables to be used by open source drivers.

```
export IBM_DB_DIR=INSTALL_PATH
export IBM_DB_LIB=INSTALL_PATH/lib
export IBM_DB_INCLUDE=INSTALL_PATH/include
export DB2_HOME=INSTALL_PATH
export DB2LIB=INSTALL_PATH/lib
```
- Environment variables to be used by SQLJ, JDBC, and JCC drivers.

```
export CLASSPATH=INSTALL_PATH/java/db2jcc.jar:$CLASSPATH
export CLASSPATH=INSTALL_PATH/java/sqlj.zip:$CLASSPATH
```

- Environment variables to be used by CLPPlus utility.

```
export CLASSPATH=INSTALL_PATH/tools/clpplus.jar:$CLASSPATH
export CLASSPATH=INSTALL_PATH/tools/jline-0.9.93.jar:$CLASSPATH
export CLASSPATH=INSTALL_PATH/tools/antlr-3.2.jar:$CLASSPATH
```

Setting up the application development environment (Windows)

To set up an application development environment on Windows operating systems, you must follow these steps.

About this task

When you install the IBM Data Server 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 system 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. If **DB2INSTANCE** is not set, you do not have to make use of the **DB2INSTDEF** DB2 registry variable which defines the default instance name to use.

Procedure

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 quotation marks, 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 quotation marks.

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 “Upgrading 32-bit database applications to run on 64-bit instances” in *Upgrading to DB2 Version 10.5*.

- To run an application against a remote server, both the client and the server must be properly configured. For instance, the **DB2COMM** environment variable must be set at the server of the remote database.

Chapter 27. Configuring the IBM Data Server Client for database application development

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

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)” and “Installing IBM data server clients (Windows)” in *Installing IBM Data Server Clients*.
- Optional: Created sample databases with which to test you applications
See “The SAMPLE database” in *SQL Reference Volume 1*.
- Completed basic configuration steps for your Data Server Client
See “Client-to-server communications configuration overview” in *Installing IBM Data Server Clients*.

About this task

For information about configuring the Data Server Client for database application development, see Chapter 28, “Configuring the development environment for application programming interfaces,” on page 81.

Chapter 28. Configuring the development environment for application programming interfaces

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.

Before you begin

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

- An IBM data server client installed
- The API drivers installed
 - See Part 5, “Supported database application programming interfaces,” on page 21.
- 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*.

Procedure

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

1. For ADO .NET, see “ADO.NET development for IBM Data Servers” in *Developing ADO.NET and OLE DB Applications* for details.
2. For CLI and ODBC, perform any of the following tasks that apply to your environment:
 - “Setting up your application development environment to run CLI and ODBC applications” in *Call Level Interface Guide and Reference Volume 1*
 - “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, see “Perl DBI” in *Developing Perl, PHP, Python, and Ruby on Rails Applications* for details.
7. For PHP, perform one of the following tasks:
 - Set up the PHP environment on Linux or UNIX.
 - Set up the PHP environment on Windows.
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* that apply to your environment.
9. For Python, perform the task “Setting up the Python environment for IBM data servers” in *Developing Perl, PHP, Python, and Ruby on Rails Applications*.

Configuring your development environment to build and run CLI and ODBC applications

You can run 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 CLI or ODBC applications, you need the IBM Data Server Client.

Procedure

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

1. Ensure the 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 database is on.

On Windows operating systems, you can use the CLI/ODBC Settings GUI to catalog the DB2 database.

3. Optional: Explicitly bind the 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 Windows operating systems, you can use the CLI/ODBC Settings GUI to bind the CLI/ODBC bind files to the database.

4. Optional: Change the CLI /ODBC configuration keywords by editing the `db2cli.ini` file. For information about the location of the `db2cli.ini` file, see “*db2cli.ini initialization file*” in *Call Level Interface Guide and Reference Volume 1*.

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

Results

Once you have completed steps 1 to 4, proceed to setting up your Windows CLI environment, or setting up your Linux or UNIX ODBC environment if you are running ODBC applications on Linux or 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 begin

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 and the IBM Data Server Driver Package, the DB2 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 must install an SDK for Java, 6 or later.

If you want to use JDBC 4.1 functions, you must install an SDK for Java, 7 or later.

If you plan to run JDBC or SQLJ applications on your system, but not to prepare them, you need a Java runtime environment only.

Important: Support for the SDK for Java 1.4.2 is deprecated for Java routines, and might be discontinued in a future release.

- 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 i server, the System i operating system must support the Unicode UTF-8 encoding scheme. The following table lists the System i PTFs that you need for Unicode UTF-8 support:

Table 15. System i PTFs for Unicode UTF-8 support

System i version	PTF numbers
V5R3 or later	None (support is included)

- 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 must 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.

Restrictions

If you install the IBM Data Server Driver for JDBC and SQLJ on a Windows 64-bit operating system, you cannot use IBM Data Server Driver for JDBC and SQLJ type 2 connectivity to connect to a DB2 for Linux, UNIX, and Windows instance from a 32-bit Java application.

Procedure

To install the IBM Data Server Driver for JDBC and SQLJ:

1. During the DB2 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 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 `sql1ib\java` directory for Windows systems, or the `sql1ib/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 or 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 or later functions, and JDBC 3.0 or 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 or earlier functions.

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

You also must make this change in every DB2 command line window that you open.

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. Do not include db2jcc.jar with sqlj4.zip, or db2jcc4.jar with sqlj.zip.

- If IBM Data Server Driver for JDBC and SQLJ client license files exist, the installation process installs them 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 16. 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 i	All DB2 Connect products

Client license files are not required for connections to DB2 for Linux, UNIX, and Windows, Cloudscape, or IBM Informix databases from the IBM Data Server Driver for JDBC and SQLJ version 3.50 or later. Client license files are not required for direct connections to DB2 for z/OS if DB2 Connect Unlimited Edition for System z server license activation is performed.

Recommendation: If you connect to DB2 for z/OS data servers directly, rather than through a DB2 Connect gateway, and you use DB2 Connect Unlimited Edition for System z 9.7 Fix Pack 3 or later, you must activate the server license key for DB2 Connect Unlimited Edition for System z. You must

activate the license key on each DB2 for z/OS subsystem or data sharing group to which you will connect. See the DB2 Connect information for details on server license activation.

- 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

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 the *server* and *port number* 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 TCP/IP:

```
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 svcname TCP/IP-service-name
```

The port number used for applets and SQLJ programs must be the same as the TCP/IP **svcname** 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 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.

5. If you plan to call SQL procedures that are on DB2 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.

6. 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" in *Developing Java Applications*.

Chapter 29. 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 the DB2 product 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.

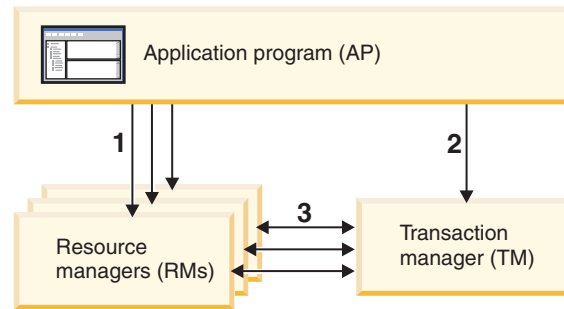
X/Open distributed transaction processing model

The X/Open Distributed Transaction Processing (DTP) model includes a number of interrelated components that control how distributed transactions are processed.

These components include the:

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

Figure 1 on page 88 illustrates this model, and shows the relationship among these components.



Legend

- 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 WebSphere application issues a request to commit a transaction, the WebSphere XA TM 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 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 in the following section.

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

AXLIB

Library that contains the **ax_reg** and **ax_unreg** functions for the TP monitor. This value is used by DB2 for Linux, UNIX, and Windows 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 for Linux, UNIX, and Windows 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 for Linux, UNIX, and Windows 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; 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.

LCS

Loosely Coupled Transaction Support. Valid values are T or F.

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 for Linux, UNIX, and Windows supports two methods of registering a global transaction. The first is Dynamic Registration, where DB2 for Linux, UNIX, and Windows calls the TP's **ax_reg** function to register the transaction (see AXLIB). The second method is Static Registration, where the TP calls the XA API **xa_start** to initiate a global transaction. Note that 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; 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.

TCTX

Specifies whether 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.

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 must 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 overrides 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.

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 for Linux, UNIX, and Windows 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 17. 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/cics/lib/libEncServer (for UNIX based systems) HOLD_CURSOR=T CHAIN_END=T SUSPEND_CURSOR=F TOC=T

Table 17. Valid Values for TPM and tp_mon_name (continued)

TPM Value	TP Monitor Product	Internal Settings
MQ	IBM MQSeries®	AXLIB=mqmax (for Windows) =/usr/mqm/lib/libmqmax_r.a (for AIX threaded applications) =/usr/mqm/lib/libmqmax.a (for AIX non-threaded applications) =/opt/mqm/lib/libmqmax.so (for Solaris) =/opt/mqm/lib/libmqmax_r.sl (for HP threaded applications) =/opt/mqm/lib/libmqmax.sl (for HP non-threaded applications) =/opt/mqm/lib/libmqmax_r.so (for Linux threaded applications) =/opt/mqm/lib/libmqmax.so (for Linux non-threaded applications) HOLD_CURSOR=F CHAIN_END=F SUSPEND_CURSOR=F TOC=P
CB	IBM Component Broker	AXLIB=somtrx1i (for Windows) =libsomtrx1 (for 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 Linux, UNIX, and Windows for MTS. MTS is automatically detected by the ODBC driver in DB2 for Linux, UNIX, and Windows.
JTA	Java Transaction API	It is not necessary to configure DB2 for Linux, UNIX, and Windows for Enterprise Java Servers (EJS) such as IBM WebSphere. The JDBC driver in DB2 for Linux, UNIX, and Windows automatically detects this environment. Therefore this TPM value is ignored.

xa_open string format for earlier versions

Earlier versions of DB2 for Linux, UNIX, and Windows 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 explicitly cataloged an alias name after you created the database. The `userid` 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 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 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 for Linux, UNIX, and Windows 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 for Linux, UNIX, and Windows 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.

Before you begin

- 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.

Procedure

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

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 attempts 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*.

About this task

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 were 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.

Procedure

To resolve indoubt transactions manually:

1. Connect to the database for which you require all transactions to be complete.
2. List 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. The **LIST INDOUBT TRANSACTIONS** command returns the current state of the indoubt transactions at the start of the interactive session, but the output is not refreshed on the client if the status changes on the server. If the status change prevents you from being

able to perform all of the actions that are listed as valid options, an error message is returned instructing you to quit the interactive session and start a new one.

- b. For host or System i database servers, you can 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, invoke the **DISPLAY THREAD TYPE(INDOUBT)** command. Use the **RECOVER** command to make a heuristic decision. To obtain indoubt information directly from DB2 for i, 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, 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, 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 to 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.

What to do next

To obtain indoubt transaction information from DB2 for Linux, UNIX, and 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

There might be times where it is useful to be able to query, commit, and roll back indoubt transactions without waiting for the Transaction Manager (TM) to perform a *re-sync* action. This situation might arise, for example, if the communication line is broken, and an indoubt transaction is tying up needed resources.

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 TM to perform the *re-sync* operation. 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 for Linux, UNIX, and Windows

DB2 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

The DB2 product 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 9.7 for AIX.
-------------	---

flags	For <i>db2xa_switch_std</i> TMREGISTER TMNOMIGRATE is set
--------------	---

Explicitly states that the DB2 product 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 the DB2 product 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 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 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`.

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. 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**, **tm_database**, **maxappls**, and **autorestart**.

- **tp_mon_name**

This database manager configuration parameter identifies the name of the TP monitor product being used (for example, TXSeries CICS).

- **tm_database**

Because DB2 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 for Linux, UNIX, and Windows client code will attempt to re-establish the connection to either the original database (which might 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, this database is called as 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 might 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 for Linux, UNIX, and Windows environments and the risks of data integrity problems in each.

High availability disaster recovery (HADR)

The DB2 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 four 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 four 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

Compared with the SYNC and NEARSYNC modes, the ASYNC mode results in shorter transaction response times but might cause greater transaction losses if the primary database fails. 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 might 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 SYNC.

- SUPERASYNC

This mode has the shortest transaction response time but has also the highest probability of transaction losses if the primary system fails. This mode is useful when you do not want transactions to be blocked or experience elongated response times due to network interruptions or congestion. The primary database considers log write successful only when logs have been written to its own log files. The primary database does not wait for acknowledgement of any kind from the standby database. The logs might 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 SYNC.

Partitioned database environments

The use of automatic client reroute (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, can 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).

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 PowerHA[®] SystemMirror for AIX, 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, can result in data integrity problems.

Transactions accessing partitioned databases

In a partitioned database environment, user data might be distributed across database partitions. An application accessing the database connects and sends requests to the database partition identified as 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.

Global variables and XA transactions

In an XA environment, session global variable objects will not change. Session global variables, within an XA transaction, are not available to other sessions joining the transaction.

Introduction

If a global variable is created within a session, then it cannot be used by other sessions until the unit of work has committed. However, the newly created variable can be used within the session that created it before the unit of work is committed. This behavior is consistent with other created objects such as tables.

The length attribute and data type of a global variable does not vary once it is created. Furthermore, the global variable cannot be altered at all (for example, its default value).

The setting of a global variable's value is non-transactional. Hence, an application cannot roll back the setting of the value of a global variable. Note the following:

- If the creation of a global variable is rolled back then the variable no longer exists.
- If the drop of a global variable is rolled back, the value of the global variable will be what it was before the drop.

Session global variables can have a default value defined for them. This value is automatically assigned to the variable as soon as it is first referenced in a session.

XA interaction

Session global variables are scoped to a per session level within an XA transaction and are not available to other sessions that join the transaction.

Global variables are non-transactional state objects. Hence, in an XA environment (for both tightly coupled transactions and loosely coupled transactions), global variable objects (both existence and value) will not change at XA_end (nor for other XA APIs, like XA_start, XA_rollback).

Note: In contrast, DB2 for Linux, UNIX and Windows has implemented declared global temporary tables as transactional state objects. Hence, when the following XA APIs are executed, declared global temporary tables are dropped.

- XA_rollback
- XA_end(SUCCESS)
- XA_start(NOFLAGS)
- XA_start(JOIN)

Configuring IBM WebSphere Application Server

About this task

IBM WebSphere Application Server is a Java-based application server. It can use the DB2 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

About this task

IBM WebSphere MQ is a message oriented business integration server. It can use the DB2 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/wmqv7/v7r0/index.jsp>.

Configuring IBM TXSeries CICS

About this task

For information about how to configure IBM TXSeries CICS to use DB2 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://www-01.ibm.com/software/http/cics/txseries/library/>. Host and System i database servers can participate in CICS-coordinated transactions.

Configuring BEA Tuxedo

What follows is a description of the process to configure BEA Tuxedo for use with DB2 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 for Linux, UNIX, and Windows or a 32-bit instance of DB2 for Linux, UNIX, and Windows.

About this task

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 *db2xacist*. The old switch data structure and API names can be used but only when working with a 32-bit instance of DB2 for Linux, UNIX, and Windows.

Procedure

To configure Tuxedo to use DB2 for Linux, UNIX, and Windows as a resource manager:

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 database 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 the DB2 database product 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 database-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 from DB2 for z/OS and OS/390®, DB2 for IBM i, or DB2 for VM&VSE, the DB2 Connect XA concentrator is required. Keep the default setting for HOLD_CURSOR (HOLD_CURSOR=F) as outlined in "Valid Values for TPM and tp_mon_name" table in the xa_open string formats topic.
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.

Compiling the unixODBC driver manager

You can compile the unixODBC Driver Manager on Linux or UNIX operating systems for use with CLI and ODBC applications.

Procedure

To compile the unixODBC Driver Manager:

1. Download the latest unixODBC source code from <http://www.unixodbc.org>.
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 with 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 are copied to the [prefix]/lib directory, and executable files are 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. 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 are dynamically loading the driver manager:

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

The **ar** commands in previous example produce libodbc.so.1, libodbcinst.so.1, and libodbccr.so.1 in the [prefix]/lib/so directory.

10. Build the application and ensure that 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 you are specifying the paths of the user and system INI files. Do not use relative paths or environment variables.

Note: If you are compiling 64-bit applications for the ODBC Driver, use the -DODBC64 option to enable the 64-bit definitions in the driver manager.

Installing the unixODBC driver manager

The unixODBC driver manager is an open source ODBC driver manager that can be used with the DB2 ODBC driver on all supported Linux and UNIX operating systems.

Procedure

To install the unixODBC driver manager:

1. Obtain the unixODBC driver manager.
 - You can download and compile the unixODBC driver manager source code. For information about compiling the source code, see “Compiling the unixODBC driver manager” on page 109.
 - You can download the compiled version of the unixODBC driver manager from the following URLs:
 - a. For an AIX RPM file, see <http://www.perzl.org/aix/index.php?n=Main.UnixODBC>
 - b. For a Linux RPM file, see <http://rpmfind.net/linux/rpm2html/search.php?query=unixODBC>
2. Install the unixODBC driver manager. You can issue the **apt-get install** command to install the unixODBC driver manager:
`apt-get install unixodbc unixodbc-dev`

Tip: You can also use different package manager command that is based on your operating system architecture and distribution to install a compiled version of the unixODBC driver manager. For example, on Red-Hat Linux operating system, you can use the **yum** command:

```
yum install unixODBC unixODBC-devel
```

3. Configure the unixODBC driver manager by adding the following lines in the `odbcinst.ini` configuration file:

```
[DB2]
Description = DB2 Driver
Driver = <instance_path>/lib/libdb2o.so
fileusage=1
dontdlclose=1
```

The `<instance_path>` is your DB2 instance path if you installed the full IBM Data Server Client, such as IBM Data Server Client, IBM Data Server Runtime Client or DB2 database server product. If you installed the IBM Data Server Driver for ODBC and CLI, the `<instance_path>` is your installation path. You can determine the location of the `odbcinst.ini` configuration file by issuing the following command:

```
odbcinst -j
```

For more information about configuring the unixODBC driver manager, see links in the related reference topics.

Part 10. Appendixes

Appendix A. DB2 database product and packaging information

DB2 database products are available in several different editions. There are also optional DB2 features. Some DB2 database products and features are only available on specific operating systems.

The following tables list the DB2 database products and features that are available on each operating system.

Table 18. Availability of DB2 database products and features on UNIX operating systems

DB2 database products and features	AIX	HP-UX on IA-64 hardware	Solaris on UltraSPARC hardware	Solaris on x86-64 ("x64") hardware
DB2 Advanced Enterprise Server Edition	Yes	Yes	Yes	Yes
DB2 Enterprise Server Edition	Yes	Yes	Yes	Yes
DB2 Workgroup Server Edition	Yes	Yes	Yes	Yes
DB2 Personal Edition	No	No	No	No
DB2 Express Server Edition	No	No	No	Yes
DB2 Express-C	No	No	No	Yes
Data Server Client and Data Server Runtime Client	Yes	Yes	Yes	Yes
DB2 Advanced Access Control Feature	Yes	Yes	Yes	Yes
IBM DB2 High Availability Feature for Express Edition	No	No	No	Yes
IBM Homogeneous Replication Feature for DB2 Enterprise Server Edition	Yes	Yes	Yes	Yes
IBM DB2 Performance Optimization Feature for Enterprise Server Edition	Yes	Yes	Yes	Yes
DB2 Storage Optimization Feature	Yes	Yes	Yes	Yes

Table 19. Availability of DB2 database products and features on Linux operating systems

DB2 database products and features	Linux on x86-32 hardware	Linux on x64 hardware	Linux on IBM Power Systems	Linux on System z
DB2 Advanced Enterprise Server Edition	No ¹	Yes	Yes	Yes
DB2 Enterprise Server Edition	No ¹	Yes	Yes	Yes
DB2 Workgroup Server Edition	Yes	Yes	Yes	No
DB2 Personal Edition	Yes	Yes	No	No
DB2 Express Server Edition	Yes	Yes	Yes	No
DB2 Express-C	Yes	Yes	Yes	No
Data Server Client and Data Server Runtime Client	Yes	Yes	Yes	Yes
DB2 Advanced Access Control Feature	No ¹	Yes	Yes	Yes
DB2 Geodetic Data Management Feature	No ¹	Yes	No	Yes
IBM DB2 High Availability Feature for Express Edition	Yes	Yes	Yes	No
IBM Homogeneous Replication Feature for DB2 Enterprise Server Edition	No ¹	Yes	Yes	Yes

Table 19. Availability of DB2 database products and features on Linux operating systems (continued)

DB2 database products and features	Linux on x86-32 hardware	Linux on x64 hardware	Linux on IBM Power Systems	Linux on System z
IBM DB2 Performance Optimization Feature for Enterprise Server Edition	No ¹	Yes	Yes	Yes
DB2 Storage Optimization Feature	No ¹	Yes	Yes	Yes
Note: 1. Only test and development are supported in DB2 Enterprise Server Edition for Linux on x86-32 hardware. However, you can use DB2 Enterprise Server Edition for Linux on x86-32 as a try and buy or by installing the Developer Edition license certificate onto a trial copy.				

Table 20. Availability of DB2 database products and features on Windows operating systems

DB2 database products and features	Windows 7 (32-bit editions)	Windows 7 (64-bit editions) on x86 64-bit ("x64") hardware	Windows Server 2003, Windows Server 2008 (32-bit editions)	Windows Server 2003, Windows Server 2008 (64-bit editions) on x64 hardware
DB2 Advanced Enterprise Server Edition	No ¹	No ¹	Yes	Yes
DB2 Enterprise Server Edition	No ¹	No ¹	Yes	Yes
DB2 Workgroup Server Edition	Yes	Yes	Yes	Yes
DB2 Personal Edition	Yes	Yes	Yes	Yes
DB2 Express Server Edition	Yes	Yes	Yes	Yes
DB2 Express-C	Yes	Yes	Yes	Yes
Data Server Client and Data Server Runtime Client	Yes	Yes	Yes	Yes
DB2 Advanced Access Control Feature	No ¹	No ¹	Yes	Yes
DB2 Geodetic Data Management Feature	No ¹	No ¹	Yes	Yes
IBM DB2 High Availability Feature for Express Edition	Yes	Yes	Yes	Yes
IBM Homogeneous Replication Feature for DB2 Enterprise Server Edition	No ¹	No ¹	Yes	Yes
IBM DB2 Performance Optimization Feature for Enterprise Server Edition	No ¹	No ¹	Yes	Yes
DB2 Storage Optimization Feature	No ¹	No ¹	Yes	Yes
Note: 1. Only test and development are supported in DB2 Advanced Enterprise Server Edition, DB2 Enterprise Server Edition, and Developer Edition on Windows .				

For information about DB2 database product and packaging information, see <http://www-1.ibm.com/support/docview.wss?rs=73&uid=swg21219983>.

For information about DB2 Express-C, see: www.ibm.com/software/data/db2/express

Appendix B. DB2 technical information

DB2 technical information is available in multiple formats that can be accessed in multiple ways.

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

- Online DB2 documentation in IBM Knowledge Center:
 - Topics (task, concept, and reference topics)
 - Sample programs
 - Tutorials
- Locally installed DB2 Information Center:
 - Topics (task, concept, and reference topics)
 - Sample programs
 - Tutorials
- DB2 books:
 - PDF files (downloadable)
 - PDF files (from the DB2 PDF DVD)
 - Printed books
- Command-line help:
 - Command help
 - Message help

Important: The documentation in IBM Knowledge Center and the DB2 Information Center is updated more frequently than either the PDF or the hardcopy books. To get the most current information, install the documentation updates as they become available, or refer to the DB2 documentation in IBM Knowledge Center.

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

Documentation feedback

The DB2 Information Development team values 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 Information Development team reads all of your feedback but cannot respond to you directly. Provide specific examples wherever possible to 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 the db2docs@ca.ibm.com email address to contact DB2 Customer Support. If you have a DB2 technical issue that you cannot resolve by using the documentation, contact your local IBM service center for assistance.

DB2 technical library in hardcopy or PDF format

You can download the DB2 technical library in PDF format or you can order in hardcopy from the IBM Publications Center.

English and translated DB2 Version 10.5 manuals in PDF format can be downloaded from DB2 database product documentation at www.ibm.com/support/docview.wss?rs=71&uid=swg27009474.

The following tables describe the DB2 library available from the IBM Publications Center at <http://www.ibm.com/e-business/linkweb/publications/servlet/pbi.wss>. Although the tables identify books that are available in print, the books might not be available in your country or region.

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

The DB2 documentation online in IBM Knowledge Center is updated more frequently than either the PDF or the hardcopy books.

Table 21. DB2 technical information

Name	Form number	Available in print	Availability date
<i>Administrative API Reference</i>	SC27-5506-00	Yes	28 July 2013
<i>Administrative Routines and Views</i>	SC27-5507-01	No	1 October 2014
<i>Call Level Interface Guide and Reference Volume 1</i>	SC27-5511-01	Yes	1 October 2014
<i>Call Level Interface Guide and Reference Volume 2</i>	SC27-5512-01	No	1 October 2014
<i>Command Reference</i>	SC27-5508-01	No	1 October 2014
<i>Database Administration Concepts and Configuration Reference</i>	SC27-4546-01	Yes	1 October 2014
<i>Data Movement Utilities Guide and Reference</i>	SC27-5528-01	Yes	1 October 2014
<i>Database Monitoring Guide and Reference</i>	SC27-4547-01	Yes	1 October 2014
<i>Data Recovery and High Availability Guide and Reference</i>	SC27-5529-01	No	1 October 2014
<i>Database Security Guide</i>	SC27-5530-01	No	1 October 2014
<i>DB2 Workload Management Guide and Reference</i>	SC27-5520-01	No	1 October 2014
<i>Developing ADO.NET and OLE DB Applications</i>	SC27-4549-01	Yes	1 October 2014
<i>Developing Embedded SQL Applications</i>	SC27-4550-00	Yes	28 July 2013

Table 21. DB2 technical information (continued)

Name	Form number	Available in print	Availability date
<i>Developing Java Applications</i>	SC27-5503-01	No	1 October 2014
<i>Developing Perl, PHP, Python, and Ruby on Rails Applications</i>	SC27-5504-01	No	1 October 2014
<i>Developing RDF Applications for IBM Data Servers</i>	SC27-5505-00	Yes	28 July 2013
<i>Developing User-defined Routines (SQL and External)</i>	SC27-5501-00	Yes	28 July 2013
<i>Getting Started with Database Application Development</i>	GI13-2084-01	Yes	1 October 2014
<i>Getting Started with DB2 Installation and Administration on Linux and Windows</i>	GI13-2085-01	Yes	1 October 2014
<i>Globalization Guide</i>	SC27-5531-00	No	28 July 2013
<i>Installing DB2 Servers</i>	GC27-5514-01	No	1 October 2014
<i>Installing IBM Data Server Clients</i>	GC27-5515-01	No	1 October 2014
<i>Message Reference Volume 1</i>	SC27-5523-00	No	28 July 2013
<i>Message Reference Volume 2</i>	SC27-5524-00	No	28 July 2013
<i>Net Search Extender Administration and User's Guide</i>	SC27-5526-01	No	1 October 2014
<i>Partitioning and Clustering Guide</i>	SC27-5532-01	No	1 October 2014
<i>pureXML Guide</i>	SC27-5521-00	No	28 July 2013
<i>Spatial Extender User's Guide and Reference</i>	SC27-5525-00	No	28 July 2013
<i>SQL Procedural Languages: Application Enablement and Support</i>	SC27-5502-00	No	28 July 2013
<i>SQL Reference Volume 1</i>	SC27-5509-01	No	1 October 2014
<i>SQL Reference Volume 2</i>	SC27-5510-01	No	1 October 2014
<i>Text Search Guide</i>	SC27-5527-01	Yes	1 October 2014
<i>Troubleshooting and Tuning Database Performance</i>	SC27-4548-01	Yes	1 October 2014
<i>Upgrading to DB2 Version 10.5</i>	SC27-5513-01	Yes	1 October 2014
<i>What's New for DB2 Version 10.5</i>	SC27-5519-01	Yes	1 October 2014
<i>XQuery Reference</i>	SC27-5522-01	No	1 October 2014

Table 22. DB2 Connect technical information

Name	Form number	Available in print	Availability date
Installing and Configuring DB2 Connect Servers	SC27-5517-00	Yes	28 July 2013
DB2 Connect User's Guide	SC27-5518-01	Yes	1 October 2014

Displaying SQL state help from the command line processor

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

Procedure

To start 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 DB2 documentation online for different DB2 versions

You can access online the documentation for all the versions of DB2 products in IBM Knowledge Center.

About this task

All the DB2 documentation by version is available in IBM Knowledge Center at <http://www.ibm.com/support/knowledgecenter/SSEPGG/welcome>. However, you can access a specific version by using the associated URL for that version.

Procedure

To access online the DB2 documentation for a specific DB2 version:

- To access the DB2 Version 10.5 documentation, follow this URL:
http://www.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.kc.doc/welcome.html.
- To access the DB2 Version 10.1 documentation, follow this URL:
http://www.ibm.com/support/knowledgecenter/SSEPGG_10.1.0/com.ibm.db2.luw.kc.doc/welcome.html.
- To access the DB2 Version 9.8 documentation, follow this URL:
http://www.ibm.com/support/knowledgecenter/SSEPGG_9.8.0/com.ibm.db2.luw.kc.doc/welcome.html.
- To access the DB2 Version 9.7 documentation, follow this URL:
http://www.ibm.com/support/knowledgecenter/SSEPGG_9.7.0/com.ibm.db2.luw.kc.doc/welcome.html.

- To access the DB2 Version 9.5 documentation, follow this URL:
http://www.ibm.com/support/knowledgecenter/SSEPGG_9.5.0/com.ibm.db2.luw.kc.doc/welcome.html.

Terms and conditions

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

Applicability: These terms and conditions are in addition to any terms of use for the IBM website.

Personal use: You may reproduce these publications for your personal, noncommercial 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.

Rights: 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 previous 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.

IBM Trademarks: IBM, the IBM logo, and [ibm.com](http://www.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 www.ibm.com/legal/copytrade.shtml

Appendix C. Notices

This information was developed for products and services offered in the U.S.A. Information about non-IBM products is based on information available at the time of first publication of this document and is subject to change.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information about 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 grant 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 character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, 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, changes, or both in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to websites not owned by IBM are provided for convenience only and do not in any manner serve as an endorsement of those

websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

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

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

IBM Canada Limited
U59/3600
3600 Steeles Avenue East
Markham, Ontario L3R 9Z7
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 contains 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. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample 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 or registered trademarks of Oracle, its affiliates, 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, Celeron, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
- 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 software 47

A

ActiveX Data Object (ADO) specification

IBM Data Server Provider for .NET 23

ActiveX data objects

DB2 Data Server Client support 19

ADO.NET applications

developing 23

AIX

installing

DB2 server products 7

IBM data server clients 7

APIs

heuristic 98

overview 21

application design

prototyping in Node.js 35

application development

application programming interfaces (APIs)

configuring 81

summary 21

configuring

environment (Linux) 75

environment (overview) 73

environment (UNIX) 75

environment (Windows) 77

environment for using application programming

interfaces (APIs) 81

IBM Data Server Client 79

operating systems 75

elements supported 3

embedded SQL overview 29

environment

configuring (Linux) 75

configuring (operating system) 75

configuring (overview) 73

configuring (UNIX) 75

configuring (Windows) 77

configuring for application programming interfaces

(APIs) 81

overview 1

IBM Data Server Client

configuring 79

details 19

IBM Data Server Provider for .NET 23

Linux

configuring 75

operating systems

configuring 75

Linux 75

supported 5

UNIX 75

Perl DBI 37

programming languages 45

tools 3, 71

transaction managers 69

application development (*continued*)

UNIX

configuring 75

applications

programming interfaces 21

automatic client reroute

resource manager setup 103

B

BEA Tuxedo

configuring 108

C

C language

versions supported 49

C/C++ language

versions supported 51

call level interface

see CLI 27

CLI

environment setup 82

IBM Data Server Client support 19

overview 27

clients

configuring 79

COBOL language

versions supported 53

command line processor (CLP)

IBM Data Server Client 19

connection keywords

ODBC .NET Data Provider 24

OLE DB .NET Data Provider 25

connections

failures

automatic client reroute 103

D

database applications

programming interfaces 21

DB2 documentation

available formats 117

DB2 documentation versions

IBM Knowledge Center 120

DB2 products

general information 115

packaging 115

Distributed Transaction Processing

application program (AP) 87

resource manager (RM) 87

transaction manager (TM) 87

X/Open Distributed Transaction Processing (DTP)

model 87

Distributed Transaction Processing (DTP)

configuring 103

databases

connections 103

updating servers (host and System i) 96

- Distributed Transaction Processing (DTP) *(continued)*
 - error handling 97
 - security 99
- documentation
 - PDF files 118
 - printed 118
 - terms and conditions of use 121
- DTP
 - see Distributed Transaction Processing 87
- dynamic SQL
 - Node.js support 35

E

- embedded SQL applications
 - IBM Data Server Client support 19
 - overview 29
- environment variables
 - UNIX
 - overview 76

F

- FORTTRAN language
 - supported versions 55

G

- global variables
 - XA transactions 106

H

- hardware
 - requirements
 - DB2 server products (AIX) 7
 - DB2 server products (HP-UX) 9
 - DB2 server products (Linux) 11
 - DB2 server products (Solaris Operating System) 15
 - DB2 server products (Windows) 17
 - IBM data server clients (AIX) 7
 - IBM data server clients (HP-UX) 9
 - IBM data server clients (Linux) 11
 - IBM data server clients (Solaris Operating System) 15
 - IBM data server clients (Windows) 17
- help
 - SQL statements 120
- heuristic decisions
 - resolving indoubt transactions 97
- heuristic operations
 - resolving indoubt transactions manually 97
- host databases
 - updating with XA transaction managers 96
- HP-UX
 - installing
 - DB2 servers 9
 - IBM data server clients 9

I

- IBM data server clients
 - application development 3
- IBM Data Server Driver for JDBC and SQLJ
 - installing 82

- IBM Data Server Driver for JDBC and SQLJ *(continued)*
 - levels for DB2 versions 57
- IBM Data Server Provider for .NET
 - documentation 23
 - overview 23
- IBM Data Studio
 - development tools 71
- IBM Database Add-Ins for Visual Studio
 - development tools 71
- IBM Knowledge Center
 - DB2 documentation versions 120
- IBM OLE DB Provider
 - consumers 33
 - installing 33
 - overview 33
 - providers 33
 - system requirements 33
 - versions 33
- IBM TXSeries CICS
 - configuring 107
- ibm_db API
 - details 41
- IBM_DB Ruby driver and Rails adapter
 - details 43
- ibm_db_dbi API
 - details 41
- ibm_db_sa adaptor
 - details 41
- ibm_db2 API
 - details 39
- indoubt transactions
 - recovering
 - X/Open Distributed Transaction Processing (DTP) model 87
 - resolving 97
- installation
 - IBM Data Server Driver for JDBC and SQLJ 82
 - requirements
 - AIX 7
 - HP-UX 9
 - Linux 11
 - Solaris Operating System 15
 - Windows 17

J

- Java
 - DB2 product support 57
 - IBM Data Server Client support 19
- JDBC
 - drivers
 - details 31, 57
 - IBM Data Server Client support 19
 - IBM Data Server Driver for JDBC and SQLJ installation 82

L

- Linux
 - application development environment 75
 - installing
 - DB2 servers 11
 - IBM data server clients 11
 - libraries
 - libaio.so.1 11
 - libstdc so.5 11

LIST INDOUBT TRANSACTIONS command
 resolving indoubt transactions manually 97

M

Microsoft Visual Studio .NET
 viewing IBM Data Server Provider for .NET
 documentation 23

N

node-ibm_db driver
 overview 35
notices 123

O

ODBC
 driver managers
 installing 111
 unixODBC 109
 overview 27
 unixODBC driver manager 109, 111
ODBC .NET Data Provider
 overview 24
OLE automation
 IBM Data Server Client support 19
OLE DB
 IBM Data Server Client support 19
 table functions 33
OLE DB .NET Data Provider
 overview 25
online DB2 documentation
 IBM Knowledge Center 120
operating systems
 application development 3, 5, 75
 configuring 75
 installation requirements
 AIX 7
 HP-UX 9
 Linux 11
 Solaris Operating System 15
 Windows 17

P

partitioned database environments
 transactions
 accessing databases 103
pdo_ibm
 details 39
Perl
 Database Interface (DBI) specification 37
 versions 61
PHP
 application development 39
 extensions for IBM data servers 39
 versions 63
precompilers
 IBM Data Server Client support 19
programming languages
 C 49
 C++ 51
 COBOL 53
 FORTRAN 55

programming languages (*continued*)

 Perl 61
 PHP 63
 REXX 67
 Ruby on Rails 65
 summary 45

Python
 application development 41
 extensions for IBM data servers 41

R

Rails adapter
 details 43
Rational Application Developer (RAD) 71
Rational Data Architect 71
Rational Software Architect 71
resolution
 indoubt transactions 97
resource managers
 overview 87
 setting up databases as 103
REXX language
 versions supported 67
RMs
 overview 87
 setting up databases as 103
Ruby
 supported versions 65
Ruby driver
 details 43
Ruby on Rails
 overview 71
 supported versions 65

S

SAMPLE database
 connecting to
 ODBC .NET Data Provider 24
 OLE DB .NET Data Provider 25
SDKs
 product levels 57
software requirements
 AIX 7
 HP-UX 9
 Linux 11
 Solaris Operating System 15
 Windows 17
Solaris operating systems
 DB2 servers 15
 IBM data server clients 15
SQL Access Group 27
SQL statements
 dynamic 29
 embedded 29
 help
 displaying 120
 static 29
SQLAlchemy
 adapter for IBM data servers 41
SQLJ
 drivers 31
 IBM Data Server Client support 19
static profiling
 development tools 71

- System i
 - database servers
 - updating with XA transaction managers 96
- system requirements
 - IBM OLE DB Provider for DB2 33

T

- terms and conditions
 - publications 121
- threads
 - IBM OLE DB Provider for DB2 33
- tp_mon_name configuration parameter 91
- TPM parameter 91
- transaction managers
 - BEA Tuxedo 108
 - distributed transaction processing 87
 - IBM TXSeries CICS 107
 - IBM WebSphere Application Server 107
 - IBM WebSphere MQ 107
 - overview 87
 - XA 99, 100, 102, 103
 - XA-compliant 69
- transaction processing monitors
 - BEA Tuxedo 108
 - IBM TXSeries CICS 107
 - XA transaction manager 103
- transactions
 - accessing partitioned databases 103
 - global 87
 - loosely coupled
 - X/Open distributed transaction processing (DTP) model 87
 - non-XA 87
 - tightly coupled 87
 - two-phase commit 87
- Tuxedo
 - configuring 108
- TXSeries CICS 107

U

- UNIX
 - application development
 - environment variable settings 76
 - application development environment 75
- unixODBC driver manager
 - installing 111
 - setting up 109

W

- WebSphere Application Server
 - overview 107
- WebSphere MQ
 - overview 107
- Windows
 - application development environment configuration 77
 - installing
 - DB2 servers (requirements) 17
 - IBM data server clients (requirements) 17

X

- X/Open Company 27

- X/Open distributed transaction processing (DTP) model
 - details 87
- X/Open SQL CLI 27
- XA
 - interface 87
 - specification 100
 - switch 100
 - transactions 106
- XA transaction managers
 - configuration 103
 - list 69
 - security 99
 - troubleshooting 102
 - updating host databases 96
 - updating System i databases 96

Z

- Zend Core 71



Printed in USA

GI13-2084-01



Spine information:

IBM DB2 10.5 for Linux, UNIX, and Windows

Getting Started with Database Application Development

