

IBM DB2 Connect 10.5

DB2 Connect User's Guide

Updated October, 2014



IBM DB2 Connect 10.5

DB2 Connect User's Guide

Updated October, 2014



Note

Before using this information and the product it supports, read the general information under Appendix B, "Notices," on page 165.

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 1993, 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 1. DB2 Connect overview 1

Key Concepts	1
Client and server connection options	1
Functionality in DB2 features in DB2 Connect product editions	2
Host databases	4
DB2 Connect and SQL statements	4
DB2 Connect administration utilities	5
InfoSphere Federation Server and DB2 Connect.	5
DB2 Connect scenarios	6
DB2 Connect client access to host databases	6
DB2 Connect server products as connectivity servers	7
DB2 Connect and transaction processing monitors	8

Chapter 2. Installing DB2 Connect server 13

Supported DB2 Connect interface languages	13
Displaying the DB2 Setup wizard in your national language (Linux and UNIX)	13
Language identifiers for running the DB2 Setup wizard in another language	13
Changing the DB2 Connect product interface language (Windows)	15
Changing the DB2 Connect interface language (Linux and UNIX)	15
Conversion of character data	16
Prerequisites for installing DB2 Connect server product.	17
Installation requirements for DB2 Connect server products (AIX)	17
Installation requirements for DB2 Connect server products (HP-UX)	19
Installation requirements for DB2 Connect server products (Linux).	20
Installation requirements for DB2 Connect products (Solaris)	21
Installation requirements for DB2 Connect server products (Windows)	22
DB2 Connect disk and memory requirements	23
Java software support for DB2 Connect	24
Preparing to install DB2 Connect for Linux on zSeries	26
Kernel parameters (Linux and UNIX).	27
Modifying kernel parameters for DB2 Connect (HP-UX)	27
Recommended kernel configuration parameters for DB2 Connect (HP-UX)	28
Modifying kernel parameters for DB2 Connect (Linux)	28
Modifying kernel parameters for DB2 Connect (Solaris)	29

DB2 Connect server products: installation and configuration overview	30
AIX	31
HP-UX	34
Linux	36
Solaris	39
Windows	42
Maintaining license keys	48
Registering a DB2 Connect license key using the db2licm command	48
Setting the DB2 Connect license policy using the db2licm command	49
Post-installation tasks	49
Adding your user ID to the DB2ADMNS and DB2USERS user groups (Windows)	49
Applying fix packs to DB2 Connect	50
Uninstalling	53
Uninstalling DB2 Connect (Windows)	53
Uninstalling DB2 Connect (Linux and UNIX)	54

Chapter 3. Upgrading to the latest version of DB2 Connect 55

Upgrade essentials for DB2 Connect	56
Pre-upgrade tasks for DB2 Connect servers.	57
Upgrading DB2 Connect servers	58
Post-upgrade tasks for DB2 Connect servers	60

Chapter 4. Configuring 63

Preparing IBM DB2 for IBM i for connections from DB2 Connect	63
Preparing DB2 for z/OS for connections from DB2 Connect	64
Host databases	65
Configuring TCP/IP for DB2 for z/OS	65
Configuring DB2 for z/OS	68
Preparing DB2 for VSE & VM for connections from DB2 Connect	68
Sysplex support	68
DB2 Connect server Sysplex support	69
Configuring connections to IBM mainframe database servers.	71
Registering a DB2 Connect license key using the db2licm command	71

Chapter 5. Administering 73

Binding applications and utilities (DB2 Connect server)	73
Moving data with DB2 Connect	76
Automatic client reroute description and setup (DB2 Connect server)	78
Administering DB2 Connect systems	79
Overview	79
Distributed Relational Database Architecture	85
Updating database directories	88
DB2 Connect and SQL statements	98

Multisite Updates	98
SQLCODE mapping	101
Chapter 6. Monitoring DB2 Connect server	107
Monitoring connections for remote clients	107
Monitoring performance using the Windows Performance Monitor	107
Using the GET SNAPSHOT commands.	108
DCS application status	110
Chapter 7. Developing database applications	115
Running your own applications	115
Application compatibility on DB2 for z/OS	115
Chapter 8. Security	119
Trusted connections through DB2 Connect.	119
Creating and terminating a trusted connection through CLI.	120
Switching users on a trusted connection through CLI.	121
DB2 Connect authentication considerations	124
Kerberos support	125
Authentication types supported with DB2 Connect server	126
Chapter 9. Tuning	127
DB2 Connect performance considerations	127
Application design	130
Connection management	133
Connection pooling	133
Connection concentrator.	135
Connection pooling and connection concentrator	139
Connection concentrator required with WebSphere MQ Transaction Manager and DB2 for z/OS	140
DB2 Connect server tuning.	140

Host database tuning.	142
Network tuning considerations	142
System resources contention	144
DB2 Connect performance troubleshooting	144
Tuning DB2 for z/OS.	144
Increasing DB2 Connect data transfer rates	145
Extra query block	145
RFC-1323 Window scaling	146
High availability and load balancing for host database connectivity.	147
Host data conversion.	148
Data types for character data	148
Network hardware	149
CLI/ODBC application performance tuning	150

Chapter 10. Troubleshooting	151
Troubleshooting DB2 Connect servers	151
Gathering relevant information	151
Initial connection is not successful	151
Problems encountered after an initial connection	152
Diagnostic tools	153

Chapter 11. Messages.	155
Common DB2 Connect problems.	155

Appendix A. DB2 technical information	159
DB2 technical library in hardcopy or PDF format	160
Displaying SQL state help from the command line processor.	162
Accessing DB2 documentation online for different DB2 versions	162
Terms and conditions.	163

Appendix B. Notices	165
--------------------------------------	------------

Index	169
------------------------	------------

About this book

The *DB2 Connect User's Guide* provides all the information you need to learn about and use the DB2 Connect™ product. DB2 Connect concepts are presented with typical scenario showing the relationships between DB2 Connect and the other parts of the network environment. Considerations involving database directories, security between systems, multisite updates, moving data, and monitoring DB2 Connect are discussed. How DB2 Connect supports high availability in your network environment is presented. Ensuring good performance by DB2 Connect and across the network is introduced as are some topics concerned with troubleshooting possible problems.

Who should use this book?

System administrators, database administrators, and system communications specialists would all be interested in part or all of this book.

Chapter 1. DB2 Connect overview

DB2 Connect provides connectivity to mainframe and midrange databases from Linux, UNIX, and Windows operating systems. You can connect to DB2® databases on the z/OS®, IBM® i, VSE, and VM operating systems and on IBM Power Systems™ hardware.

You can also connect to databases that you did not create by using IBM products if they are Distributed Relational Database Architecture™ (DRDA®) compliant.

DB2 Connect is the industry-leading solution integrating System z®, System i® and other enterprise data with client/server, web, mobile, and service-oriented architecture applications. DB2 Connect delivers significant feature enhancements to improve programmer productivity, provide a more robust infrastructure, and enable the deployment of DB2 technology. DB2 Connect has several product offerings:

- DB2 Connect Enterprise Edition
- DB2 Connect Application Server Edition
- DB2 Connect Unlimited Edition for System z
- DB2 Connect Unlimited Edition for System i
- IBM DB2 Connect Application Server Advanced Edition
- IBM DB2 Connect Unlimited Advanced Edition for System z

For detailed information about DB2 Connect product offerings, see:
<http://www.ibm.com/software/data/db2/db2connect/>.

It is strongly recommended that you use DB2 Connect client, notably the IBM data server drivers and clients, instead of the DB2 Connect server. IBM data server drivers and clients provide the same connection and application development functionality as the DB2 Connect server. However, you can reduce complexity, improve performance, and deploy application solutions with smaller footprints for your business users. DB2 Connect license files are required. For more information about DB2 Connect client, refer to Client and server connection options.

Key Concepts

Client and server connection options

DB2 Connect server provides a single point of connectivity to numerous workstations supporting a variety of applications. However, it adds additional processing time to applications accessing DB2 for z/OS data and increases the elapsed time of those applications.

Starting with DB2 Connect Version 8 and later, DB2 Connect clients use the DRDA protocol natively to connect directly to DB2 for z/OS and DB2 for IBM i.

Advantages of using DB2 Connect server

DB2 Connect server is advantageous in the following situations:

- For two-phase commits, if you are using transaction managers that use a dual transport model

- For Homogeneous Federation

Advantages of using DB2 Connect client

You can replace DB2 Connect server with DB2 Connect client, choosing from among the various IBM data server drivers, the IBM Data Server Runtime Client, or the IBM Data Server Client. DB2 Connect client and drivers offer functionality that is equivalent or superior to that of DB2 Connect server and includes the following other advantages:

- **Enhanced Performance.** You can achieve better performance due to less network traffic and code paths. DB2 Connect clients simplify network topology, since a direct connection is established between the application server and DB2 z/OS. This will also eliminate network hop and DB2 Connect gateway routing. Reduced resource consumption means hardware or software resources are not required for DB2 Connect server machines.
- **Reduced footprint.** By replacing DB2 Connect server with DB2 Connect client, you can reduce complexity and deploy application solutions with smaller footprints and achieve overall benefits.
- **Improved availability.** Application access, using IBM data server drivers or clients, to DB2 for z/OS data equals to or is superior to three-tier configuration due to elimination of a point of failure.
- **Improved monitoring.** A direct connection makes it easier to monitor application server or web application server traffic and behavior.
- **Improved problem determination.** If an application experiences a performance problem, the presence of the DB2 Connect server complicates the efforts to identify the source of the problem.
- **Latest code levels.** You can obtain the latest code levels to exploit new server features and APIs. Data support for some features such as new data types is easier to obtain.

If you replace DB2 Connect server with DB2 Connect client, DB2 Connect license files are required. In a DB2 Connect server configuration, DB2 Connect entitlement is stored at the DB2 Connect server, not at individual clients. If you change to direct client connectivity, you must store the DB2 Connect entitlement at each client.

Functionality in DB2 features in DB2 Connect product editions

Some functionality is available in only certain DB2 Connect product editions. In some cases, the functionality is associated with a particular DB2 feature.

The table indicates which functionality is included in a DB2 Connect product edition. If the functionality is not applicable to the DB2 Connect products, the value "Not applicable" is specified.

Table 1. Functionality in DB2 Connect product editions

Functionality	DB2 Connect server editions
Adaptive Compression	No
Advanced Copy Service	Yes
Compression: backup	No
Compression: Data	No
Compression: Index	No
Compression: Temp table	No

Table 1. Functionality in DB2 Connect product editions (continued)

Functionality	DB2 Connect server editions
Compression: XML	No
Connection concentrator	Yes
Continuous Data Ingest	No
Database partitioning	No
DB2 Governor	Yes
Heterogeneous Federation	No
High availability disaster recovery	Yes
Homogeneous Federation	Yes
Homogeneous Q Replication	No
IBM Data Studio	Yes
IBM InfoSphere® Optim™ Configuration Manager for z/OS ⁴	No
IBM InfoSphere Optim Performance Manager Extended Edition ¹	No
IBM InfoSphere Optim pureQuery® Runtime	Yes ²
Label-based access control (LBAC)	No
Materialized query tables (MQT)	Yes
Multidimensional clustering (MDC) tables	Yes
Multi-Temperature Storage	No
Online reorganization	No
DB2 pureScale®	No
pureXML® storage	No
Query parallelism	Yes
Replication tools	Yes ³
Scan Sharing	No
Spatial Extender	Yes
Time Travel Query	Yes
Table partitioning	No
Tivoli® System Automation	Yes
Workload management	Yes
Note: <ol style="list-style-type: none"> 1. IBM InfoSphere Optim Performance Manager Extended Edition is a follow-on to Performance Expert. IBM InfoSphere Optim Performance Manager Extended Edition helps optimize the performance and availability of mission-critical databases and applications. 2. Only DB2 Connect Unlimited Edition for System z and DB2 Connect Application Server Advanced Edition include IBM InfoSphere Optim pureQuery Runtime. 3. Replication tools except the Replication Center are available on all supported operating systems. The Replication Center is available only on Linux and Windows operating systems. 4. IBM InfoSphere Optim Configuration Manager for z/OS is packaged with DB2 Connect Unlimited Advanced Edition for System z. 	

Host databases

A host database is a relational database system from which a link request originates.

The term *database* is used throughout this document to describe a relational database management system (RDBMS). Other systems with which DB2 Connect communicates might use the term database to describe a slightly different concept. The DB2 Connect term database can also refer to:

System z

DB2 for z/OS. A DB2 for z/OS subsystem identified by its LOCATION NAME. Use the z/OS **-display ddf** command to get the DB2 server location name, domain name, IP address and port.

A DB2 for z/OS location is the unique name of a database server. An application uses the location name to access a DB2 for z/OS subsystem or a DB2 for z/OS data sharing group. A data sharing group enables applications on different DB2 subsystems to read from and write to the same data concurrently. The application uses a DB2 data sharing group network address to access a DB2 data sharing location. The accessed DB2 subsystem is transparent to the application.

Since DB2 for z/OS supports multiple databases at the same DB2 location, the location name is analogous to a Linux, UNIX, and Windows database alias name. A database alias can be used to override the location or location alias name when accessing a location. A location alias is another name for a location. It is used to control which subsystems in a data sharing group are accessed by an application.

LOCATION NAME is also defined in the Boot Strap Data Set (BSDS) as well as the DSNL004I message (LOCATION=location), which is written when the Distributed Data Facility (DDF) is started. LOCATION NAME supports up to 8 alias location names, allowing applications the ability to use different dbalias names to access a Version 8 z/OS server.

IBM Power Systems Servers

IBM DB2 for IBM i, an integral part of the IBM i operating system. Only one database can exist on an IBM Power Systems server unless the system is configured to use independent auxiliary storage pools.

DB2 Connect and SQL statements

DB2 Connect forwards SQL statements submitted by application programs to IBM mainframe database servers.

DB2 Connect can forward almost any valid SQL statement, as well as the supported DB2 APIs (application programming interfaces):

- JDBC
- SQLJ
- ADO.NET
- OLE DB
- ODBC
- Perl
- PHP
- pureQuery
- Python

- Ruby
- CLI
- Embedded SQL

Embedded SQL support

Two types of embedded SQL processing exist: static SQL and dynamic SQL. Static SQL minimizes the time required to execute an SQL statement by processing in advance. Dynamic SQL is processed when the SQL statement is submitted to the IBM mainframe database server. Dynamic SQL is more flexible, but potentially slower. The decision to use static or dynamic SQL is made by the application programmer. Both types are supported by DB2 Connect.

Different IBM mainframe database servers implement SQL differently. DB2 Connect fully supports the common IBM SQL, as well as the DB2 for z/OS, DB2 Server for VM and VSE (formerly SQL/DS), and IBM DB2 for IBM i implementations of SQL. IBM SQL is strongly recommended for maintaining database independence.

DB2 Connect administration utilities

You can use administration utilities to manage your DB2 Connect servers.

You can use the following utilities to administer DB2 Connect servers:

- The Command Line Processor (CLP) or CLPPlus. You can use the CLP or CLPPlus to issue SQL statements against an IBM mainframe database server database. The SQL statements are issued on the database that you specify.

Note: CLPPlus for administration is available in the IBM data server driver package and does not require DB2 Connect server modules to be installed.

- Replication tools to set up and administer all replication programs for Q replication and SQL replication. These tools are the Replication Center, the ASNCLP command-line program, and the Replication Alert Monitor tool. The Replication Center is available only on Linux and Windows operating systems.
- Import and export utilities. You can use these utilities to load, import, and to export data to and from a file on a workstation or an IBM mainframe database server database. You can then use these files to import data into databases, spreadsheets, and other applications running on your workstation.
- The Event Viewer and the Performance Monitor. If you are running a DB2 Connect server product, you can use these tools. Using the Event Viewer, you can view exception events that DB2 Connect logs. Using the Performance Monitor, you can monitor and manage the performance of DB2 Connect servers either locally or remotely.
- The database system monitor utility. You can use this utility to monitor system connections. This function is available only when DB2 Connect is acting as server. You can also use this utility to determine the source of an error. You can correlate client applications with the corresponding jobs running on the IBM mainframe database server.

InfoSphere Federation Server and DB2 Connect

InfoSphere Federation Server is a separate product offering that provides access to and integration of data across multivendor data sources, while DB2 Connect enables you to leverage the large volumes of data located in existing host and midrange servers.

InfoSphere Federation Server helps integrate information by allowing a collection of data sources to be viewed and manipulated as if they were a single source. It makes data source access completely transparent to the calling application. InfoSphere Federation Server works in conjunction with DB2 Connect server products. InfoSphere Federation Server provides native read and write access to the DB2 family of products, Informix®, Oracle, Sybase, Teradata, and Microsoft SQL Server databases. InfoSphere Federation Server also provides read access to nonrelational and life sciences data sources such as Documentum, IBM Lotus® Extended Search, table-structured files, and XML. You can use it to formulate queries on data in a federated system.

DB2 Connect scenarios

DB2 Connect can provide a variety of solutions to your IBM mainframe database access needs.

This topic outlines several scenarios that might apply to your particular needs or environment.

DB2 Connect client access to host databases

DB2 Connect's basic feature is providing a direct connection to a host database from desktop applications running on your workstations. IBM Data Server Driver Package with DB2 Connect license is the simplest way to provide this solution.

Each workstation that has a client package and DB2 Connect license installed can establish a direct TCP/IP connection to DB2 for z/OS, IBM DB2 for IBM i, and DB2 for Linux, UNIX, and Windows servers. In addition, applications can connect to and update multiple DB2 family databases in the same transaction with the complete data integrity provided by the two-phase commit protocol.

Figure 1 on page 7 shows a direct connection to an IBM mainframe database server from a workstation with DB2 Connect installed.

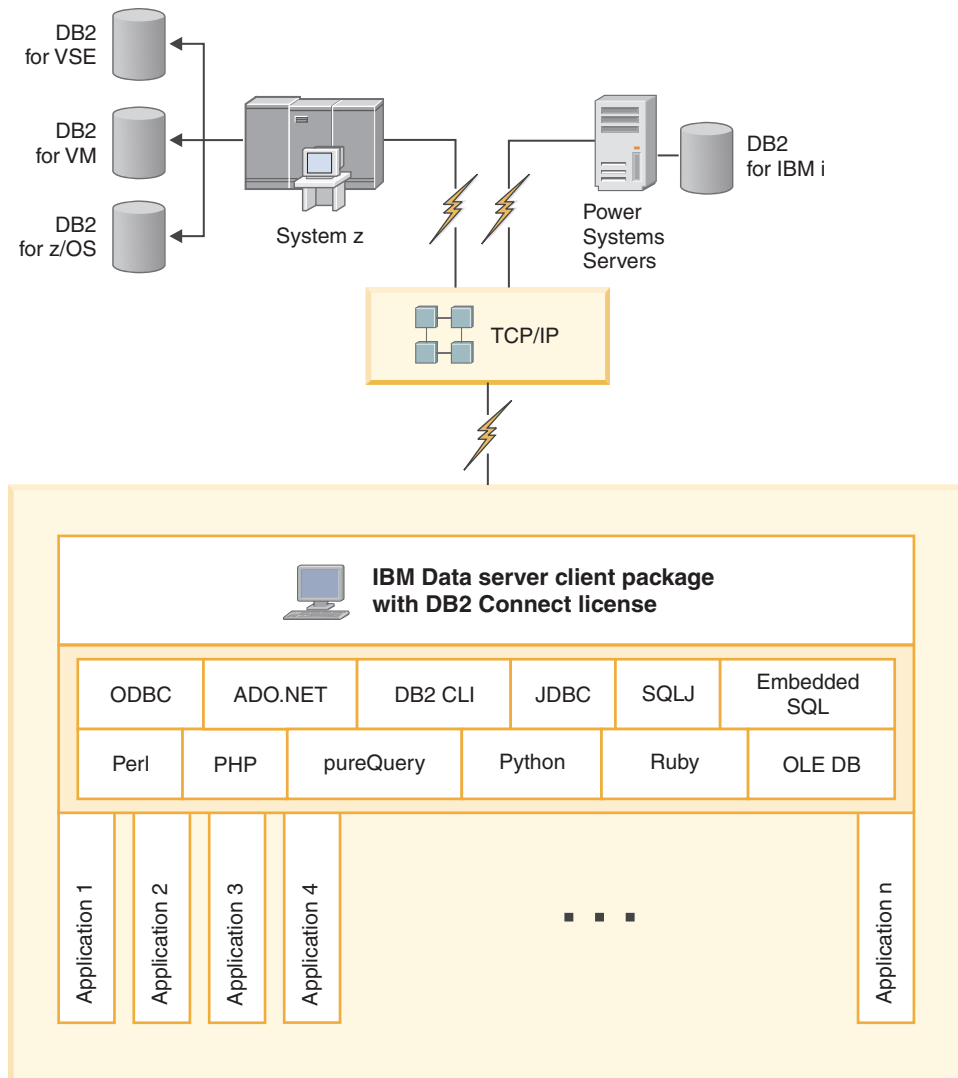


Figure 1. Direct Connection Between DB2 Connect and an IBM mainframe database server

Note:

1. All IBM data server drivers provide the ability to perform workload balancing and seamless automatic client reroute features without requiring DB2 Connect modules to be installed or configured.

DB2 Connect server products as connectivity servers

A DB2 Connect Server is used to provide a single point of connectivity to numerous workstations supporting a variety of applications.

Figure 2 on page 8 illustrates IBM's solution for environments in which you want a DB2 client to make an indirect connection to an IBM mainframe database server through a DB2 Connect server product, such as DB2 Connect Enterprise Edition.

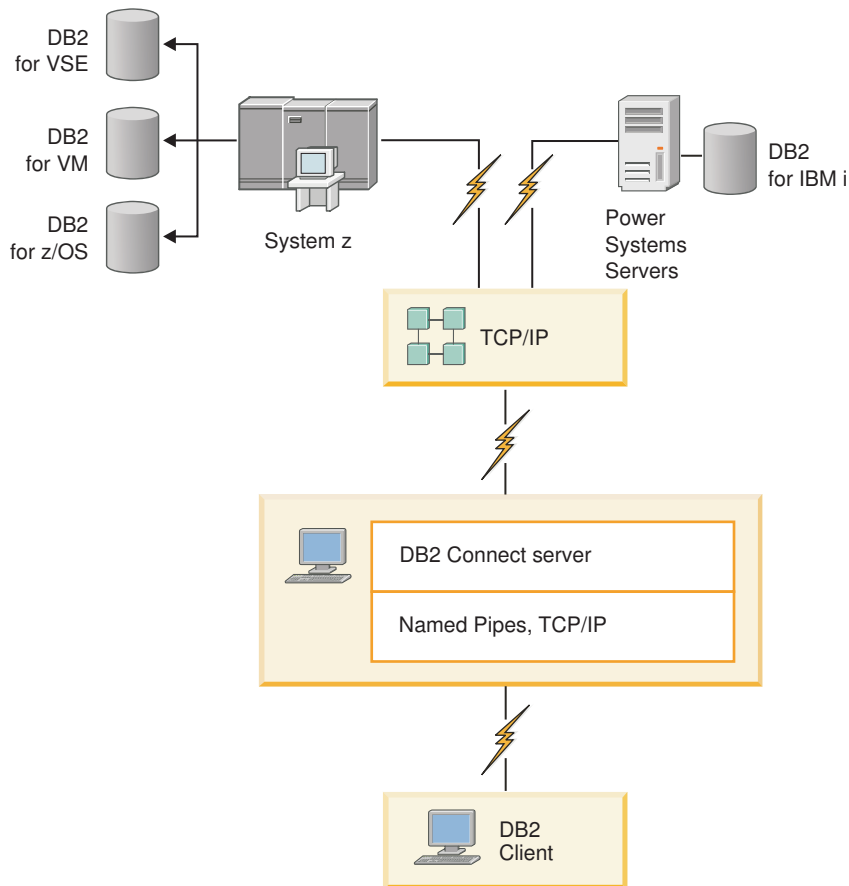


Figure 2. DB2 Connect Enterprise Edition

If a TCP/IP connection to the DB2 Connect server is lost, the client will automatically attempt to reestablish the connection. The client will first attempt to reestablish the connection to the original server. If the connection is not reestablished, the client will failover to an alternate DB2 Connect server. (The alternate server is specified on the server instance and its location is returned to the client during the connection.) If the connection to the alternate server is not reestablished, the client will attempt to reestablish the connection to the original server. The client will continue the attempts to reestablish the connection, switching between the original server and the alternate server, until the connection is established or the number of attempts time out.

DB2 Connect and transaction processing monitors

Transaction processing supports interactive applications in which requests are processed as soon as they are received and returned to the requester in a relatively short period of time. You can use Transaction Processing (TP) monitors to process transactions in an organized way.

An application server permits a large number of users to execute applications using a minimum of system resources. An application server can be extended to allow coordinated transactions to be invoked from applications executed by the application server. This transaction coordination is generally known as a Transaction Processing (TP) monitor. A TP monitor works in conjunction with an application server.

Transaction processing

Every organization has rules and procedures that describe how it is supposed to operate. The user applications which implement these rules can be called *business logic*. The transactions these business applications execute are often referred to as Transaction Processing or Online Transaction Processing (OLTP).

The key characteristics of commercial OLTP are:

Many Users

It is common for transaction processing to be used by the majority of the people in an organization, since so many people affect the current state of the business.

Repetitive

Most interactions with the computer tend to be the same process executed over and over again. For example, entering an order or processing payments are used many times every day.

Short Interactions

Most interactions that people in the organization have with the transaction processing system are short in duration.

Data Sharing

Since data represents the state of the organization, there can only be a single copy of the data.

Data Integrity

The data must represent the current state of the organization, and must be internally consistent. For example, every order must be associated with a customer record.

Low Cost/Transaction

Since the transaction processing represents a direct cost of doing business, the cost of the system must be minimal. DB2 Connect allows applications under the control of an application server running on Linux, UNIX, and Windows to execute transactions against remote LAN, and IBM mainframe database servers and have these transactions coordinated by a TP monitor.

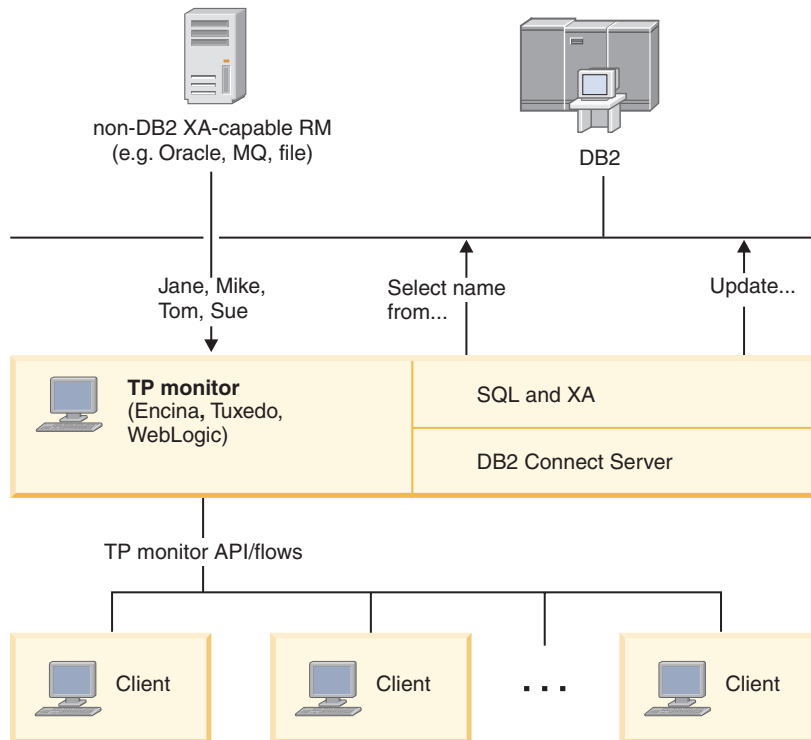


Figure 3. DB2 Connect support for TP monitors

In Figure 3, the APIs, as well as the connectivity mechanism between the application server and the back-end database servers, are provided by a DB2 Connect server product, such as DB2 Connect Enterprise Edition.

Examples of transaction processing monitors

The most common TP monitors on the market today are:

- IBM WebSphere® Application Server
- IBM WebSphere MQ
- IBM TxSeries CICS®
- BEA Tuxedo
- BEA WebLogic
- Microsoft Transaction Server (MTS)

Remote IBM Power Systems, System z, and LAN database servers can be used within transactions coordinated by these TP monitors.

X/Open Distributed Transaction Processing (DTP) model

An application executing business logic might be required to update multiple resources within a single transaction. For example, a bank application which implements a transfer of money from one account to another could require debiting one database (the "from" account) and depositing to another database (the "to" account).

It is also possible that different vendors provide these two databases. For example, one database is a DB2 for z/OS and the other is an Oracle database. Rather than have every TP monitor implement each database vendor's proprietary transaction interface, a common transaction interface between a TP monitor and any resource

accessed by an application has been defined. This interface is known as the *XA Interface*. A TP monitor that uses the XA Interface is referred to as an *XA compliant Transaction Manager (TM)*. An updatable resource that implements the XA interface is referred to as an *XA compliant Resource Manager (RM)*.

The previously listed TP monitors are all XA compliant TMs. Remote host, IBM Power Systems, and DB2 LAN-based databases, when accessed via DB2 Connect, are XA compliant RMs. Therefore, any TP monitor which has an XA compliant TM can use host, IBM Power Systems, and LAN-based DB2 databases within business applications executing transactions.

Chapter 2. Installing DB2 Connect server

Supported DB2 Connect interface languages

DB2 language support for DB2 interfaces can be categorized into server group languages and client group languages.

Server group languages will translate most messages, help, and DB2 graphical interface elements. Client group languages will translate the IBM Data Server Runtime Client component, which will include most messages and certain help documentation.

Server group languages include: Brazilian Portuguese, Czech, Danish, Finnish, French, German, Italian, Japanese, Korean, Norwegian, Polish, Russian, Simplified Chinese, Spanish, Swedish, and Traditional Chinese.

Client group languages include: Arabic, Bulgarian, Croatian, Dutch, Greek, Hebrew, Hungarian, Portuguese, Romanian, Slovak, Slovenian, and Turkish.

Do not confuse languages supported by the DB2 database product with languages supported by the DB2 interface. Languages supported by the DB2 database product means the languages in which data can exist. These languages are a superset of languages supported by the DB2 interface.

Displaying the DB2 Setup wizard in your national language (Linux and UNIX)

The **db2setup** command queries the operating system to determine the existing language settings. If the language setting of your operating system is supported by **db2setup**, then that language will be used when displaying the DB2 Setup wizard.

If your system uses the same code pages but different locale names than those supported by the DB2 interface, you can still see the translated **db2setup** by setting your **LANG** environment variable to the appropriate value by entering the following command:

bourne (sh), korn (ksh), and bash shells:

```
LANG=locale
export LANG
```

C shell:

```
setenv LANG locale
```

where *locale* is a locale supported by the DB2 interface.

Language identifiers for running the DB2 Setup wizard in another language

If you want to run the DB2 Setup wizard in a language different from the default language on your computer, you can start the DB2 Setup wizard manually, specifying a language identifier. The language must be available on the platform where you are running the installation.

On Windows operating systems, you can run **setup.exe** with the **-i** parameter to specify the two-letter language code of the language the installation is to use.

On Linux and UNIX operating systems, it is recommended that you set the **LANG** environment variable to display the DB2 Setup wizard in your national language.

Table 2. Language identifiers

Language	Language identifier
Arabic (available on Windows platforms only)	ar
Brazilian Portuguese	br
Bulgarian	bg
Chinese, Simplified	cn
Chinese, Traditional	tw
Croatian	hr
Czech	cz
Danish	dk
Dutch	nl
English	en
Finnish	fi
French	fr
German	de
Greek	el
Hungarian	hu
Indonesian (available on Windows platforms only)	id
Italian	it
Japanese	jp
Korean	kr
Lithuanian (available on Windows platforms only)	lt
Norwegian	no
Polish	pl
Portuguese	pt
Romanian	ro
Russian	ru
Slovak	sk
Slovenian	sl
Spanish	es
Swedish	se
Turkish	tr

Changing the DB2 Connect product interface language (Windows)

The DB2 interface language is the language that appears in messages, help, and graphical tool interfaces.

About this task

Do not confuse languages supported by a DB2 database product with languages supported by the DB2 interface. Languages supported by a DB2 database product means the languages in which data can exist. These languages are a superset of languages supported by the DB2 interface.

The DB2 interface language you want to use must be installed on your system. The DB2 database product interface languages are selected and installed when you install a DB2 database product using the DB2 Setup wizard. If you change the interface language of a DB2 database product to a supported interface language that has not been installed, the DB2 database product interface language will default to the operating system language first, and if that is not supported, English.

Changing the interface language for a DB2 database product on Windows requires that you change the default language setting for your Windows operating system.

Procedure

To change the DB2 database product interface language on Windows operating systems:

1. Through the Control Panel, select **Regional and Language Options**.
2. On the **Regional Options** tab under **Standards and formats**, select the appropriate language. On Windows, use the **Formats** tab for this step.
3. On the **Regional Options** tab under **Location**, select the location that corresponds to the appropriate language.
4. On the **Advanced** tab under **Language for non-Unicode programs** select the appropriate language. On Windows, on the **Administrative** tab, under **Language for non-unicode programs**, click **Change system locale** and select the appropriate language. You will then be asked to reboot, click **Cancel**.
5. On the **Advanced** tab under **Default user account settings**, check the **Apply all settings to the current user account and to the default user profile** box. On Windows, on the **Administrative** tab under **reserved accounts**, click **Copy to reserved accounts** and check the accounts that you want to copy the language settings to.
6. You will be asked to reboot before these changes come into effect.

What to do next

Refer to your operating system help for additional information about changing the default system language.

Changing the DB2 Connect interface language (Linux and UNIX)

The interface language of the DB2 database product is the language that appears in messages, help, and graphical tool interfaces.

Before you begin

Do not confuse languages supported by the DB2 database product with languages supported by the DB2 interface. Languages supported by the DB2 database product, that is, languages that *data* can exist in, are a superset of languages supported by the DB2 interface.

Support for the DB2 interface language you want to use must be installed on your system. DB2 interface language support is selected and installed when you install a DB2 database product using the DB2 Setup wizard. If you change the interface language of the DB2 database product to a supported interface language that has not been installed, the DB2 interface language will default to the operating system language. If the operating system language is not supported, English is used as the DB2 interface language.

DB2 interface language support is selected and installed when you install your DB2 database product using the DB2 Setup wizard or by using the National Language Package.

About this task

To check which public locales are available in your system, run the **\$ locale -a** command.

Procedure

To change the DB2 interface language:

Set the **LANG** environment variable to the locale you want.

- For bourne (sh), korn (ksh), and bash shells:

```
LANG=locale
export LANG
```

- For C shell:

```
setenv LANG locale
```

For example, to interface with the DB2 database product in French, you must have the French language support installed and you must set the **LANG** environment variable to a French locale, for example, `fr_FR`.

Conversion of character data

When character data is transferred between machines, it must be converted to a form that the receiving machine can use.

For example, when data is transferred between a DB2 Connect server and a host or System i database server, it is usually converted from a server code page to a host CCSID, and vice versa. If the two machines use different code pages or CCSIDs, code points are mapped from one code page or CCSID to the other. This conversion is always performed at the receiver.

Character data sent *to* a database consists of SQL statements and input data. Character data sent *from* a database consists of output data. Output data that is interpreted as bit data is not converted. For example, data from a column declared with the FOR BIT DATA clause. Otherwise, all input and output character data is converted if the two machines have different code pages or CCSIDs.

For example, if DB2 Connect is used to access data, the following happens:

1. DB2 Connect sends an SQL statement and input data to System z.
2. DB2 for z/OS converts the SQL statement and data to the host server's code page and then processes the data.
3. DB2 for z/OS sends the result back to the DB2 Connect server.
4. DB2 Connect converts the result to the code page of the user's environment.

For bidirectional languages, a number of special "BiDi CCSIDS" have been defined by IBM and are supported by DB2 Connect.

If the bidirectional attributes of the database server are different from those of the client you can use these special CCSIDS to manage the difference.

Refer to the supported territory codes and code pages topic for the supported conversions between code pages on the DB2 Connect and CCSIDs on the host or System i server.

Prerequisites for installing DB2 Connect server product

Before you install DB2 Connect server products, ensure that the necessary prerequisites are met, such as disk, memory, and paging space requirements. There are also additional prerequisites that depend on your operating system.

The following topics provide detailed information about the installation prerequisites that you need to meet to install DB2 Connect server products.

Installation requirements for DB2 Connect server products (AIX)

Before you install DB2 Connect server 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.

To install a DB2 Connect server product, such as DB2 Connect Enterprise Edition, the following requirements must be met:

Installation requirements

Table 3. AIX installation requirements

Operating System	Hardware
<p>AIX Version 6.1²</p> <ul style="list-style-type: none"> • 64-bit AIX kernel is required • AIX 6.1 Technology Level (TL) 7 and Service Pack (SP) 6 • Minimum C++ runtime level requires the xLC.rte 12.1.0.0³ and xLC AIX rte 12.1.0.0³ (or later) filesets. <p>AIX Version 7.1</p> <ul style="list-style-type: none"> • 64-bit AIX kernel is required • AIX 7.1 Technology Level (TL) 1 and Service Pack (SP) 6 • Minimum C++ runtime level requires the xLC.rte 12.1.0.0³ and xLC AIX rte 12.1.0.0³ (or later) filesets. 	<p>64-bit Common Hardware Reference Platform (CHRP) architecture, excluding POWER3 processor-based systems.¹</p> <p>All processors that are capable of running the supported AIX operating systems.</p>

- ¹To verify that it is a CHRP architecture system, issue the command **lscfg** and look for the following output: Model Architecture: chrp. For POWER4 processor-based systems, first upgrade to POWER5 processor-based systems before installing DB2 Version 10.5. POWER4 processor-based systems are not supported in DB2 Version 10.5.
- ²In AIX 6.1 there are two types of Workload Partitions (WPARs): system WPARs and application WPARs. DB2 installation is supported only on a system WPAR. AIX 6.1 also supports the ability to encrypt a JFS2 file system or set of files.
- ³In Version 10.5 Fix Pack 2, the minimum xLC.rte level was relaxed from 12.1.0.1 to 12.1.0.0. Prior to Fix Pack 2, if you downloaded and installed Fix Pack 1, you might receive an error message regarding an insufficient xLC.rte level. This error is fixed by downloading and installing Fix Pack 2 that contains the relaxed xLC.rte level.

Software requirements

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

```
ln -sf /usr/lib/boot/unix_64 /unix
ln -sf /usr/lib/boot/unix_64 /usr/lib/boot/unix
bosboot -a
shutdown -Fr
```
- 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 from the IBM AIX XL C and C++ support website.
- One of the following browsers is required to view online help and to run First Steps (**db2fs**):
 - Firefox 3.0 and later
 - Google Chrome
 - Safari 4.0
- For details regarding known AIX issues, see www.ibm.com/support/docview.wss?&uid=swg21165448

Communication requirements

When using a communication protocol, you have the following requirements:

- For TCP/IP connectivity, no additional software is required.
- For LDAP (Lightweight Directory Access Protocol) support, you require an IBM SecureWay Directory Client V3.2.1 or later.

DB2 product installation on NFS (Network File System)

The installation of DB2 products on NFS (Network File System) is not recommended. Running DB2 products on NFS (for example, NFS mounting /opt/IBM/db2/V10.5 and then running off code that was physically installed on a remote system) requires several manual setup steps. There are also a number of potential issues with setting up NFS for a DB2 server. These include possible problems that involve:

- Performance (impacted by network performance)
- Availability (you are allowing a single point of failure)
- Licensing (there is no checking done across machines)
- Diagnosing NFS errors can be difficult

As mentioned, the setup for NFS will require several manual actions including:

- Ensuring that the mount point preserve the install path
- Permission must be controlled (for example, write permission should not be given to the mounting machine)
- DB2 registries have to be set up manually and maintained across all mounting machines
- The **db2ls** command, which lists installed DB2 products and features, must be set up and maintained properly if you need to detect DB2 products and features
- More care is required when updating your DB2 product environment
- More steps are required when cleaning up on the exporting machine and the mounting machine

Installation requirements for DB2 Connect server products (HP-UX)

Before you install DB2 Connect server products on HP-UX 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.

To install a DB2 Connect server product, such as DB2 Connect Enterprise Edition, on HP-UX, the following requirements must be met:

Note: A 64-bit HP-UX operating system is required to support DB2 Connect.

Installation requirements

Table 4. HP-UX installation requirements

Operating System	Hardware
HP-UX 11i v3 (11.31) with: <ul style="list-style-type: none"> • PHSS_37202 • PHKL_41481 • PHKL_42035 • PHKL_42335 • PHKL_41588 • PHSS_41496 	Itanium based HP Integrity Series Systems

Software requirements

- A browser is required to view online help.
- For details regarding known HP-UX issues, see www.ibm.com/support/docview.wss?&uid=swg21257602

Communication requirements

You can use TCP/IP

- For TCP/IP connectivity, no additional software is required.

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.

Installation requirements for DB2 Connect server products (Linux)

Before you install DB2 Connect server products on Linux 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.

To install a DB2 Connect server product, such as DB2 Connect Enterprise Edition, the following requirements must be met:

Hardware requirements

Your processor can be:

- x86 (Intel Pentium, Intel Xeon, and AMD Athlon)
- x64 (Intel EM64T and AMD64)

- POWER® (any Power Systems Servers, pSeries, System i, System p®, and POWER Systems that support Linux)
- System z (formerly eServer™ zSeries)

Distribution requirements

For the latest information about the supported Linux distributions, point your browser to www.ibm.com/db2/linux/validate.

You might be required to update your kernel configuration parameters. The kernel configuration parameters are set in `/etc/sysctl.conf`. See the Modifying kernel parameters (Linux) section of the DB2 Information Center. Refer to your operating system manual for information about setting and activating these parameters using the **sysctl** command.

Software requirements

- 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 DB2 Connect or if you want to use any DB2 graphical tools.
- A browser is required to view online help.

Communication requirements

For TCP/IP connectivity, no additional software is required.

Installation requirements for DB2 Connect products (Solaris)

Before you install DB2 Connect products on the Solaris Operating System, ensure that the system you choose meets the necessary operating system, hardware, software, and communications requirements. The installation requirements are same for the DB2 Connect Enterprise Edition.

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.

To install a DB2 Connect product on Solaris, the following requirements must be met:

Table 5. Solaris installation requirements

Operating System	Hardware
Solaris 10 8/11 Update 10 • 64-bit kernel	Solaris x64 (Intel 64 or AMD64)
Solaris 10 8/11 Update 10 • 64-bit kernel	UltraSPARC or SPARC64 processors

1. Support is only for the DB2 product to be installed on local zones. Installation on the global zone is not supported by the DB2 product at this time.

Operating system requirements

"Recommended & Security Patches" can be obtained from the <http://www.oracle.com/technetwork/java/index.html> Web site. From this website, click on the "Patches" menu item in the left panel.

The J2SE Solaris Operating System Patch Clusters are also required. They can be obtained from the <http://www.oracle.com/technetwork/java/index.html> Web site.

The Fujitsu PRIMEPOWER patches for the Solaris operating system can be downloaded from FTSI at: <http://download.ftsi.fujitsu.com/>. For an additional list of issues that can affect DB2 database systems on Solaris, refer to: www.ibm.com/support/docview.wss?&uid=swg21257606

DB2 database products support Solaris ZFS filesystems and Logical Domains (LDoms).

For details about virtualization technology supported by DB2 products, see <https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/Information+Management/page/Virtualization+Support>.

Software requirements

- SUNWlibC software is required to install DB2 Connect on Solaris. It can be obtained from the <http://www.oracle.com/technetwork/java/index.html> Web site.
- A browser is required to view online help.

Communication requirements

You can use TCP/IP

- For TCP/IP connectivity, no additional software is required.
- DB2 Connect is supported on Sun Cluster 2.2 if:
 - The protocol to the host is TCP/IP
 - Two-phase commit is not used. This restriction is relaxed if the user configures the SPM log to be on a shared disk (this can be done through the **spm_log_path** database manager configuration parameter), and the failover system has an identical TCP/IP configuration (the same host name, IP address, and so on).

Installation requirements for DB2 Connect server products (Windows)

Before you install DB2 Connect server products on Windows 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.

To install a DB2 Connect server product, such as DB2 Connect Enterprise Edition, the following requirements must be met:

Hardware requirements

All Intel and AMD processors capable of running the supported Windows operating systems (32-bit and 64-bit)

Note: For Windows Server operating systems, 32-bit server images are only delivered in DB2 Developer Edition for development use. For Windows client operating systems, 32-bit and 64-bit images are available.

Software requirements

- A browser is required to view online help.

Communication requirements

- TCP/IP is supported and supplied by the operating system.

Windows (64-bit) considerations

- 32-bit UDFs and stored procedures are supported.

DB2 Connect disk and memory requirements

Ensure that an appropriate amount of disk space is available for your DB2 Connect environment, and allocate memory accordingly.

Disk requirements

The disk space required for your product depends on the type of installation you choose and the type of file system you have. The DB2 Setup wizard provides dynamic size estimates based on the components selected during a typical, compact, or custom installation.

Remember to include disk space for required databases, software, and communication products. Ensure that the file system is not mounted with concurrent I/O (CIO) option.

On Linux and UNIX operating systems, 2 GB of free space in the /tmp directory, and 512 MB of free space in the /var directory are required.

Note: On Linux and UNIX operating systems, you must install your DB2 product in an empty directory. If the directory that you have specified as the install path contains subdirectories or files, your DB2 installation might fail.

On Windows operating systems the following free space is recommended in addition to that of your DB2 product:

- 40 MB in the system drive
- 60 MB in the temporary folder specified by the temp environment variable.

Memory requirements

Memory requirements are affected by the size and complexity of your database system, the extent of database activity, and the number of clients accessing your system. At a minimum, a DB2 database system requires 256 MB of RAM¹. For a system running just a DB2 product and the DB2 GUI tools, a minimum of 512 MB of RAM is required. However, 1 GB of RAM is recommended for improved performance. These requirements do not include any additional memory requirements for other software that is running on your system. For IBM data server client support, these memory requirements are for a base of five concurrent client connections. For every additional five client connections, an additional 16 MB of RAM is required.

1. DB2 products that run on HP-UX Version 11i for Itanium-based systems require a minimum of 512 MB of RAM.

For DB2 server products, the self-tuning memory manager (STMM) simplifies the task of memory configuration by automatically setting values for several memory configuration parameters. When enabled, the memory tuner dynamically distributes available memory resources among several memory consumers including sort, the package cache, the lock list, and buffer pools.

Paging space requirements

DB2 requires paging, also called swap to be enabled. This configuration is required to support various functions in DB2 which monitor or depend on knowledge of swap/paging space utilization. The actual amount of swap/paging space required varies across systems and is not solely based on memory utilization by application software. It is only strictly required by DB2 on the Solaris and HP platforms due to their use of early paging space allocation.

A reasonable minimum swap/paging space configuration for most systems is 25-50% of RAM. Solaris and HP systems with many small databases or multiple databases tuned by STMM might require a paging space configuration of 1 x RAM or higher. These higher requirements are due to virtual memory pre-allocated per database / instance, and retained virtual memory in the case of STMM tuning multiple databases. Additional swap/paging space might be wanted to provision for unanticipated memory overcommitment on a system.

Java software support for DB2 Connect

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.

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 6. 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

Table 6. DB2 for Linux, UNIX, and Windows supported levels of SDKs for Java (continued)

	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
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.

Preparing to install DB2 Connect for Linux on zSeries

To install a DB2 database product on an IBM zSeries that is running Linux, you must make the installation image accessible to the Linux operating system.

Before you begin

You have already obtained your DB2 database product installation image.

Procedure

- Using FTP to access the installation image
From the IBM zSeries computer running Linux:

1. Enter the following command: `ftp yourserver.com`
where *yourserver.com* represents the FTP server where the DB2 database product installation image resides.
2. Enter your user ID and password.
3. Enter the following commands:

```
bin
get product_file
```

 where *product_file* represents the appropriate product package name.
- Using the DB2 database product DVD over NFS to access the installation image
 1. Mount the appropriate product DVD.
 2. Export the directory where you mounted the DVD. For example, if you mounted the DVD under `/db2dvd`, then export the `/db2dvd` directory.
 3. On the IBM zSeries computer running Linux, NFS mount this directory using the following command:

```
mount -t nfs -o ro nfsservername:/db2dvd /local_directory_name
```

 where *nfsservername* represents the host name of the NFS server, *db2dvd* represents the name of the directory being exported on the NFS server, and *local_directory_name* represents the name of the local directory.
 4. From the IBM zSeries computer running Linux, change to the directory where the DVD is mounted. You can do this by entering the `cd /local_directory_name` command, where *local_directory_name* represents the mount point of your product DVD.

Kernel parameters (Linux and UNIX)

Modifying kernel parameters for DB2 Connect (HP-UX)

For your DB2 database product to perform properly on HP-UX, you might need to update your system's kernel configuration parameters. If you update your kernel configuration parameter values, you must restart your computer.

Before you begin

You must have root user authority to modify kernel parameters.

Procedure

To modify kernel parameters:

1. Enter the **sam** command to start the System Administration Manager (SAM) program.
2. Double-click the **Kernel Configuration** icon.
3. Double-click the **Configurable Parameters** icon.
4. Double-click the parameter that you want to change and type the new value in the **Formula/Value** field.
5. Click **OK**.
6. Repeat these steps for all of the kernel configuration parameters that you want to change.
7. When you are finished setting all of the kernel configuration parameters, select **Action > Process New Kernel** from the action menu bar.

Results

The HP-UX operating system automatically restarts after you change the values for the kernel configuration parameters.

Tip:

kctune can also be used on HP-UX for adjusting kernel parameters.

Recommended kernel configuration parameters for DB2 Connect (HP-UX)

For HP-UX systems running a DB2 64-bit database system, run the **db2osconf** command to suggest appropriate kernel configuration parameter values for your system.

The **db2osconf** utility can only be run from *\$DB2DIR/bin*, where *DB2DIR* is the directory where you installed your DB2 database product.

Modifying kernel parameters for DB2 Connect (Linux)

Before installing a DB2 database system, update your Linux kernel parameters. The default values for particular kernel parameters on Linux are not sufficient when running a DB2 database system.

Before you begin

You must have root user authority to modify kernel parameters.

Procedure

To update kernel parameters on Red Hat and SUSE Linux:

1. Run the **ipcs -l** command.
2. Analyze the output to determine if there are any necessary changes required for your system. Comments have been added following the **//** to show what the parameter names are.

```
# ipcs -l

----- Shared Memory Limits -----
max number of segments = 4096           // SHMMNI
max seg size (kbytes) = 32768           // SHMMAX
max total shared memory (kbytes) = 8388608 // SHMALL
min seg size (bytes) = 1

----- Semaphore Limits -----
max number of arrays = 1024             // SEMMNI
max semaphores per array = 250          // SEMMSL
max semaphores system wide = 256000    // SEMMNS
max ops per semop call = 32             // SEMOPM
semaphore max value = 32767

----- Messages: Limits -----
max queues system wide = 1024           // MSGMNI
max size of message (bytes) = 65536     // MSGMAX
default max size of queue (bytes) = 65536 // MSGMNB
```

- Beginning with the first section on Shared Memory Limits, SHMMAX and SHMALL are the parameters that need to be looked at. SHMMAX is the maximum size of a shared memory segment on a Linux system whereas SHMALL is the maximum allocation of shared memory pages on a system.

- It is recommended to set the SHMMAX value to be equal to the amount of physical memory on your system. However, the minimum required on x86 systems is 268435456 (256 MB) and for 64-bit systems, it is 1073741824 (1 GB).
 - SHMALL is set to 8 GB by default (8388608 KB = 8 GB). If you have more physical memory than this, and it is to be used for the DB2 database system, then this parameter increases to approximately 90% of your computer's physical memory. For instance, if you have a computer system with 16 GB of memory to be used primarily for the DB2 database system, then SHMALL should be set to 3774873 (90% of 16 GB is 14.4 GB; 14.4 GB is then divided by 4 KB, which is the base page size). The `ipcs` output has converted SHMALL into kilobytes. The kernel requires this value as a number of pages. If you are upgrading to DB2 Version 10.5 and you are not using the default SHMALL setting, you must increase the SHMALL setting by an additional 4 GB. This increase in memory is required by the fast communication manager (FCM) for additional buffers or channels.
 - The next section covers the amount of semaphores available to the operating system. The kernel parameter `sem` consists of 4 tokens, `SEMMSL`, `SEMMNS`, `SEMOPM` and `SEMMNI`. `SEMMNS` is the result of `SEMMSL` multiplied by `SEMMNI`. The database manager requires that the number of arrays (`SEMMNI`) be increased as necessary. Typically, `SEMMNI` should be twice the maximum number of agents expected on the system multiplied by the number of logical partitions on the database server computer plus the number of local application connections on the database server computer.
 - The third section covers messages on the system.
 - `MSGMNI` affects the number of agents that can be started, `MSGMAX` affects the size of the message that can be sent in a queue, and `MSGMNB` affects the size of the queue.
 - `MSGMAX` should be changed to 64 KB (that is, 65535 bytes), and `MSGMNB` should be increased to 65535.
3. To modify these kernel parameters, edit the `/etc/sysctl.conf` file. If this file does not exist, create it. The following lines are examples of what should be placed into the file:
- ```
kernel.sem=250 1024000 32 1024
#Example shmmax for a 64-bit system
kernel.shmmax=1073741824
#Example shmall for 90 percent of 16 GB memory
kernel.shmall=3774873
kernel.msgmax=65535
kernel.msgmnb=65535
kernel.msgmni=2048
```
4. Run **sysctl** with **-p** parameter to load in **sysctl** settings from the default file `/etc/sysctl.conf`:
- ```
sysctl -p
```
5. To make the changes effective after every reboot:
- (SUSE Linux) Make `boot.sysctl` active
 - (Red Hat) The `rc.sysinit` initialization script will read the `/etc/sysctl.conf` file automatically

Modifying kernel parameters for DB2 Connect (Solaris)

For the DB2 database system to operate properly, it is recommended that you update your system's kernel configuration parameters. You can use the **db2osconf** utility to suggest recommended kernel parameters. If you want to take advantage of project resource controls (`/etc/project`), consult your Solaris documentation.

Before you begin

You must have root authority to modify kernel parameters.

To use the **db2osconf** command, you must first install the DB2 database system. The **db2osconf** utility can only be run from *\$DB2DIR/bin*, where *DB2DIR* is the directory where you installed your DB2 database product.

You must restart your system after modifying kernel parameters.

Procedure

To set a kernel parameter:

Add a line at the end of the */etc/system* file as follows:

```
set parameter_name = value
```

For example, to set the value of the *msgsys:msginfo_msgmax* parameter, add the following line to the end of the */etc/system* file:

```
set msgsys:msginfo_msgmax = 65535
```

What to do next

After updating the */etc/system* file, restart the system.

DB2 Connect server products: installation and configuration overview

Setting up a DB2 Connect server product, such as DB2 Connect Enterprise Edition, is a multi-step process. DB2 Connect server products are often installed with hundreds or thousands of clients connecting to IBM mainframe database servers. For this reason, it is recommended to use a test installation. After the test configuration has proven stable, you can use it as the template for an unattended installation of DB2 Connect and your clients across your organization.

The typical steps to installing and configuring a DB2 Connect server product are as follows:

1. Determine how you want to use DB2 Connect in your network.
2. Verify that you have the correct hardware and software prerequisites on both your workstation and the host database server.
3. Verify that your IBM mainframe database server is configured to accept connections from DB2 Connect servers.
4. Install your DB2 Connect software. You will use this workstation to configure and verify your IBM mainframe connections. Use the related links to find the details specific to the installation of a DB2 Connect server product on your operating system.
5. After installation, establish the connection between DB2 Connect and your IBM mainframe database system. DB2 Connect can locate and configure all TCP/IP connections for you. You can use the DB2 command line processor (CLP) commands to configure IBM mainframe databases.
6. Bind the programs and utilities provided with DB2 Connect to your IBM mainframe database.
7. Test the connection.
8. (Optional) Enable the Multisite Update feature.

9. If you are planning to use WebSphere, transaction monitors, or your own application server software, install these products or applications. For information about installing WebSphere consult the documentation provided with these products as part of the DB2 Connect server product package. For other products consult the installation documentation provided with the product.
10. Install and configure the IBM data server client. Use this workstation to test connectivity from the IBM data server client to IBM mainframe database servers, as well as to test applications that use this connectivity.
11. Use the CLP commands to connect the client to the IBM mainframe system through DB2 Connect.
12. Install a IBM data server client on all end-user workstations that will use applications that connect to IBM mainframe database servers.
13. You are now ready to use DB2 Connect with all your applications. Workstations that will be used for application development should have the IBM data server client installed.
14. If you want to use your workstation to administer DB2 for z/OS or DB2 for Linux, UNIX, and Windows, install the IBM data server client.

AIX

Installing a DB2 Connect server product (AIX)

To define your installation preferences and to install a DB2 Connect product on AIX, use the DB2 Setup wizard.

Before you begin

Before you begin your installation:

- You can install DB2 Connect using either root or non-root user authority.
- Ensure that your system meets:
 - Disk and memory requirements
 - Hardware and software requirements. Refer to “Installation requirements for DB2 Connect server products (AIX)” on page 17.
- The DB2 database product DVD must be mounted on your system.
- The DB2 Connect product image must be available. If you are installing a non-English version of a DB2 Connect product, you must also have the appropriate National Language Packages.
- Ensure that asynchronous I/O has been enabled; it must be enabled before your DB2 Connect server product can be successfully installed.
- To locate DB2 database products already installed on your system, use the **db2ls** command. Refer to the “Listing DB2 products installed on your system (Linux and UNIX)” topic in *Installing DB2 Servers* .
- The DB2 Setup wizard is a graphical installer. You must have X windows software capable of rendering a graphical user interface for the DB2 Setup wizard to run on your machine. Ensure that the X windows server is running. Ensure that you have properly exported your display. For example, export `DISPLAY=9.26.163.144:0`.
- If security software such as Lightweight Directory Access Protocol (LDAP) is used in your environment, you must manually create required DB2 users before you start the DB2 Setup wizard.

Note: Network Information Services (NIS) and Network Information Services Plus (NIS+) features are deprecated starting with DB2 Version 9.1 Fix Pack 2. Support for these features might be removed in a future release. Lightweight Directory Access Protocol (LDAP) is the recommended solution for centralized user-management services.

About this task

The DB2 Installer program is a Java-based installation tool that automates the installation and configuration of any DB2 database product. If you prefer not to use this utility, you have two alternatives. You can install a DB2 Connect product:

- Using the response file method
- Manually using the **db2setup** command. You *cannot* manually install a DB2 database product using the operating system's native installation utility **SMIT**. Any existing scripts containing this native installation utility that you use to interface and query with DB2 installations will need to change.

Procedure

To install a DB2 Connect server product, such as DB2 Connect Enterprise Edition, on AIX using the DB2 Setup wizard:

1. Change to the directory where the DVD is mounted:

```
cd /db2dvd
```

where */db2dvd* represents mount point of the DVD.

2. If you downloaded the DB2 Connect product image, you must decompress and untar the product file.

- a. Decompress the product file:

```
gzip -d product.tar.gz
```

where *product* is the name of the database product that you downloaded.

- b. Untar the product file:

```
tar xvf product.tar
```

- c. Change directory:

```
cd ./product/disk1
```

Note: If you downloaded a National Language Package, untar it into the same directory. This will create the subdirectories (for example *./nlpack/disk2*) in the same directory, and allows the installer to automatically find the installation images without prompting

3. Enter the **./db2setup** command from the directory where the product image resides to start the DB2 Setup wizard. After a few moments, the IBM DB2 Setup Launchpad opens. For multiple CD installations, issue the **db2setup** command outside the mounted CD location with either a relative or absolute path name to ensure the DB2 Connect product CD can be unmounted as required. From this window, you can view the installation prerequisites and the release notes or you can proceed directly to the installation.
4. Once you have initiated the installation, proceed through the DB2 Setup wizard installation panels and make your selections. Installation help is available to guide you through the DB2 Setup wizard. Click **Help** to invoke the online help. You can click **Cancel** at any time to exit the installation. DB2 files will only be copied to your system once you have clicked **Finish** on the last DB2 Setup

wizard installation panel. Once completed, the DB2 Connect server product is installed using the /opt/IBM/db2/V9.8 default installation path.

If you are installing on a system where this directory is already being used, the DB2 Connect product installation path will have `_xx` added to it, where `xx` are digits, starting at 01 and increasing depending on how many DB2 copies you have installed.

You can also specify your own DB2 database product installation path.

Results

National Language Packs can also be installed by running the `./db2setup` command from the directory where the National Language Pack resides, after a DB2 Connect product has been installed.

The installation logs, `db2setup.log` and `db2setup.err` will be located, by default, in the `/tmp` directory. You can specify the location of the log files.

If you want your DB2 database product to have access to DB2 documentation either on your local computer or on another computer on your network, then you must install the *DB2 Information Center*. The *DB2 Information Center* contains documentation for the DB2 database and DB2 related products. See the “Installing the DB2 Information Center using the DB2 Setup wizard (UNIX)” topic in *Installing DB2 Servers*.

Mounting CDs or DVDs (AIX)

To mount your DB2 database product CD or DVD on AIX operating systems, use the System Management Interface Tool (SMIT).

Before you begin

Depending on your system configuration, you might need to log on with root user authority to mount discs.

Procedure

To mount the CD or DVD on AIX using SMIT, perform the following steps:

1. Insert the disc in the drive.
2. Create a disc mount point by entering the `mkdir -p /disc` command, where *disc* represents the CD or DVD mount point directory.
3. Allocate a disc file system using SMIT by entering the `smit storage` command.
4. After SMIT starts, select **File Systems > Add / Change / Show / Delete File Systems > CDROM File Systems > Add CDROM File System**.
5. In the Add a File System window:
 - a. Enter a device name for your CD or DVD file system in the **DEVICE Name** field. Device names for CD or DVD file systems must be unique. If there is a duplicate device name, you may need to delete a previously-defined CD or DVD file system or use another name for your directory. In this example, `/dev/cd0` is the device name.
 - b. Enter the disc mount point directory in the MOUNT POINT window. In this example, the mount point directory is `/disc`.
 - c. In the **Mount AUTOMATICALLY at system restart** field, select yes to enable automatic mounting of the file system.
 - d. Click **OK** to close the window, then click **Cancel** three times to exit SMIT.

6. Mount the CD or DVD file system by entering the **smit mountfs** command.
7. In the Mount a File System window:
 - a. Enter the device name for this CD or DVD file system in the **FILE SYSTEM name** field. In this example, the device name is `/dev/cd0`.
 - b. Enter the disc mount point in the **Directory over which to mount** field. In this example, the mount point is `/disc`.
 - c. Enter `cd rfs` in the **Type of Filesystem** field. To view the other kinds of file systems you can mount, click **List**.
 - d. In the **Mount as READ-ONLY system** field, select **yes**.
 - e. Accept the remaining default values and click **OK** to close the window.

Results

Your CD or DVD file system is now mounted. To view the contents of the CD or DVD, place the disk in the drive and enter the `cd /disc` command where *disc* is the disc mount point directory.

HP-UX

Installing a DB2 Connect server product (HP-UX)

To define your installation preferences and to install a DB2 Connect product on HP-UX, use the DB2 Setup wizard.

Before you begin

Before you begin your installation:

- You can install DB2 Connect using either root or non-root user authority.
- Ensure that your system meets:
 - Disk and memory requirements
 - Hardware, distribution and software requirements. Refer to “Installation requirements for DB2 Connect server products (HP-UX)” on page 19.
- The DB2 database product DVD must be mounted on your system.
- The DB2 Connect product image must be available. If you are installing a non-English version of a DB2 Connect product, you must also have the appropriate National Language Packages.
- To locate DB2 database products already installed on your system, use the **db2ls** command. Refer to the “Listing DB2 products installed on your system (Linux and UNIX)” topic in *Installing DB2 Servers*.
- The DB2 Setup wizard is a graphical installer. You must have X windows software capable of rendering a graphical user interface for the DB2 Setup wizard to run on your machine. Ensure that the X windows server is running. Ensure that you have properly exported your display. For example, export `DISPLAY=9.26.163.144:0`.
- If security software such as Lightweight Directory Access Protocol (LDAP) is used in your environment, you must manually create required DB2 users before you start the DB2 Setup wizard.

Note: Network Information Services (NIS) and Network Information Services Plus (NIS+) features are deprecated starting with DB2 Version 9.1 Fix Pack 2. Support for these features might be removed in a future release. Lightweight Directory Access Protocol (LDAP) is the recommended solution for centralized user-management services.

About this task

The DB2 Installer program is a Java-based installation tool that automates the installation and configuration of any DB2 database product. If you prefer not to use this utility, you have two alternatives. You can install a DB2 Connect product:

- Using the response file method
- Manually using the **db2setup** command. You *cannot* manually install a DB2 database product using the operating system's native installation utility **swinstall**. Any existing scripts containing this native installation utility that you use to interface and query with DB2 installations will need to change.

Procedure

To install a DB2 Connect server product, such as DB2 Connect Enterprise Edition, on HP-UX using the DB2 Setup wizard:

1. Change to the directory where the DVD is mounted:

```
cd /db2dvd
```

where */db2dvd* represents mount point of the DVD.

2. If you downloaded the DB2 Connect product image, you must decompress and untar the product file.

- a. Decompress the product file:

```
gzip -d product.tar.gz
```

where *product* is the name of the database product that you downloaded.

- b. Untar the product file:

```
tar xvf product.tar
```

- c. Change directory:

```
cd ./product/disk1
```

Note: If you downloaded a National Language Package, untar it into the same directory. This will create the subdirectories (for example *./nlpack/disk2*) in the same directory, and allows the installer to automatically find the installation images without prompting

3. Enter the **./db2setup** command from the directory where the product image resides to start the DB2 Setup wizard. After a few moments, the IBM DB2 Setup Launchpad opens. For multiple CD installations, issue the **db2setup** command outside the mounted CD location with either a relative or absolute path name to ensure the DB2 Connect product CD can be unmounted as required. From this window, you can view the installation prerequisites and the release notes or you can proceed directly to the installation.
4. Once you have initiated the installation, proceed through the DB2 Setup wizard installation panels and make your selections. Installation help is available to guide you through the DB2 Setup wizard. Click **Help** to invoke the online help. You can click **Cancel** at any time to exit the installation. DB2 files will only be copied to your system once you have clicked **Finish** on the last DB2 Setup wizard installation panel. Once completed, the DB2 Connect server product is installed using the */opt/IBM/db2/V10.5* default installation path.

If you are installing on a system where this directory is already being used, the DB2 Connect product installation path will have *_xx* added to it, where *xx* are digits, starting at 01 and increasing depending on how many DB2 copies you have installed.

You can also specify your own DB2 database product installation path.

Results

National Language Packs can also be installed by running the **./db2setup** command from the directory where the National Language Pack resides, after a DB2 Connect product has been installed.

The installation logs, `db2setup.log` and `db2setup.err` will be located, by default, in the `/tmp` directory. You can specify the location of the log files.

If you want your DB2 database product to have access to DB2 documentation either on your local computer or on another computer on your network, then you must install the *DB2 Information Center*. The *DB2 Information Center* contains documentation for the DB2 database and DB2 related products. See the “Installing the DB2 Information Center using the DB2 Setup wizard (UNIX)” topic in *Installing DB2 Servers*.

Mounting CDs or DVDs for DB2 Connect (HP-UX)

To mount your DB2 database product CD or DVD on HP-UX operating systems, issue the **mount** command.

Before you begin

Depending on your system configuration, you might need root user authority to mount discs.

Procedure

To mount your DB2 database product CD or DVD on HP-UX:

1. Insert the CD or DVD in the drive.
2. If necessary, define a new directory as the mount point for the CD or DVD drive. Define `/cdrom` as the mount point using the **mkdir /cdrom** command.
3. If necessary, identify the drive device file using the **ioscan -fnC disk** command. This command lists all recognized CD or DVD drives and their associated device files. The file name will be something similar to `/dev/dsk/c1t2d0`.
4. Mount the CD or DVD drive to the mount-point directory:

```
mount -F cdfs -o rr /dev/dsk/c1t2d0 /cdrom
```
5. Obtain a file listing to verify the mount using the **ls /cdrom** command.
6. Log out.

Results

Your CD or DVD file system is now mounted. View the contents of the CD or DVD by placing it in the drive and enter the **cd /cdrom** command where `cdrom` is the mount point directory.

Linux

Installing a DB2 Connect server product (Linux)

To define your installation preferences and to install a DB2 Connect product on Linux, use the DB2 Setup wizard.

Before you begin

Before you begin your installation:

- You can install DB2 Connect using either root or non-root user authority.
- Ensure that your system meets:
 - Disk and memory requirements
 - Hardware, distribution and software requirements. Refer to “Installation requirements for DB2 Connect server products (Linux)” on page 20.
- The DB2 database product DVD must be mounted on your system.
- The DB2 Connect product image must be available. If you are installing a non-English version of a DB2 Connect product, you must also have the appropriate National Language Packages.
- To locate DB2 database products already installed on your system, use the **db2ls** command.
- The DB2 Setup wizard is a graphical installer. You must have X windows software capable of rendering a graphical user interface for the DB2 Setup wizard to run on your machine. Ensure that the X windows server is running. Ensure that you have properly exported your display. For example, export `DISPLAY=9.26.163.144:0`.
- If security software such as Lightweight Directory Access Protocol (LDAP) is used in your environment, you must manually create required DB2 users before you start the DB2 Setup wizard.

Note: Network Information Services (NIS) and Network Information Services Plus (NIS+) features are deprecated starting with DB2 Version 9.1 Fix Pack 2. Support for these features might be removed in a future release. Lightweight Directory Access Protocol (LDAP) is the recommended solution for centralized user-management services.

About this task

The DB2 Setup wizard is a Java-based installation tool that automates the installation and configuration of any DB2 database products. If you prefer not to use this utility, you have two alternatives. You can install a DB2 Connect product:

- Using the response file method
- Manually using the **db2setup** command. You *cannot* manually install a DB2 database product using the operating system's native installation utility **rpm**. Any existing scripts containing this native installation utility that you use to interface and query with DB2 installations will need to change.

Procedure

To install a DB2 Connect server product, such as DB2 Connect Enterprise Edition, on Linux using the DB2 Setup wizard:

1. Change to the directory where the DVD is mounted:

```
cd /db2dvd
```

where */db2dvd* represents mount point of the DVD.

2. If you downloaded the DB2 Connect product image, you must decompress and untar the product file.
 - a. Decompress the product file:

```
gzip -d product.tar.gz
```

where *product* is the name of the database product that you downloaded.

- b. Untar the product file:

```
tar xvf product.tar
```

- c. Change directory:

```
cd ./product/disk1
```

Note: If you downloaded a National Language Package, untar it into the same directory. This will create the subdirectories (for example `./nlpack/disk2`) in the same directory, and allows the installer to automatically find the installation images without prompting

3. Enter the `./db2setup` command from the directory where the product image resides to start the DB2 Setup wizard. After a few moments, the IBM DB2 Setup Launchpad opens. For multiple CD installations, issue the `db2setup` command outside the mounted CD location with either a relative or absolute path name to ensure the DB2 Connect product CD can be unmounted as required. From this window, you can view the installation prerequisites and the release notes or you can proceed directly to the installation.
4. Once you have initiated the installation, proceed through the DB2 Setup wizard installation panels and make your selections. Installation help is available to guide you through the DB2 Setup wizard. Click **Help** to invoke the online help. You can click **Cancel** at any time to exit the installation. DB2 files will only be copied to your system once you have clicked **Finish** on the last DB2 Setup wizard installation panel. Once completed, the DB2 Connect server product is installed using the `/opt/IBM/db2/V9.8` default installation path.

If you are installing on a system where this directory is already being used, the DB2 Connect product installation path will have `_xx` added to it, where `xx` are digits, starting at 01 and increasing depending on how many DB2 copies you have installed.

You can also specify your own DB2 database product installation path.

Results

National Language Packs can also be installed by running the `./db2setup` command from the directory where the National Language Pack resides, after a DB2 Connect product has been installed.

The installation logs, `db2setup.log` and `db2setup.err` will be located, by default, in the `/tmp` directory. You can specify the location of the log files.

If you want your DB2 database product to have access to DB2 documentation either on your local computer or on another computer on your network, then you must install the *DB2 Information Center*. The *DB2 Information Center* contains documentation for the DB2 database and DB2 related products. See the “Installing the DB2 Information Center using the DB2 Setup wizard (UNIX)” topic in *Installing DB2 Servers*.

Mounting the CD or DVD for DB2 Connect (Linux)

To mount a CD-ROM on Linux operating systems, issue the `mount` command.

Before you begin

Depending on your system configuration, you might need root user authority to mount discs.

Procedure

To mount the CD or DVD on Linux operating systems:

1. Insert the CD or DVD in the drive and enter the following command:

```
mount -t iso9660 -o ro /dev/cdrom /cdrom
```

where /cdrom represents the mount point of the CD or DVD.

2. Log out.

Results

Your CD or DVD file system is now mounted. View the contents of the CD or DVD by placing the disc in the drive and enter the **cd /cdrom** command where cdrom is the mount point directory.

Solaris

Installing a DB2 Connect server product (Solaris)

To define your installation preferences and to install a DB2 Connect product on the Solaris Operating System, use the DB2 Setup wizard.

Before you begin

Before you begin your installation:

- You can install DB2 Connect using either root or non-root user authority.
- Ensure that your system meets:
 - Disk and memory requirements
 - Hardware, distribution and software requirements. Refer to “Installation requirements for DB2 Connect products (Solaris)” on page 21.
- The DB2 database product DVD must be mounted on your system.
- The DB2 Connect product image must be available. If you are installing a non-English version of a DB2 Connect product, you must also have the appropriate National Language Packages.
- To locate DB2 database products already installed on your system, use the **db2ls** command. Refer to the “Listing DB2 products installed on your system (Linux and UNIX)” topic in *Installing DB2 Servers*.
- The DB2 Setup wizard is a graphical installer. You must have X windows software capable of rendering a graphical user interface for the DB2 Setup wizard to run on your machine. Ensure that the X windows server is running. Ensure that you have properly exported your display. For example, export `DISPLAY=9.26.163.144:0`.
- If security software such as Lightweight Directory Access Protocol (LDAP) is used in your environment, you must manually create required DB2 users before you start the DB2 Setup wizard.

Note: Network Information Services (NIS) and Network Information Services Plus (NIS+) features are deprecated starting with DB2 Version 9.1 Fix Pack 2. Support for these features might be removed in a future release. Lightweight Directory Access Protocol (LDAP) is the recommended solution for centralized user-management services.

About this task

The DB2 Setup wizard is a Java-based installation tool that automates the installation and configuration of any DB2 database products. If you prefer not to use this utility, you have two alternatives. You can install a DB2 Connect product:

- Using the response file method
- Manually using the **db2setup** command. You *cannot* manually install a DB2 database product using the operating system's native installation utility **pkgadd**. Any existing scripts containing this native installation utility that you use to interface and query with DB2 installations will need to change.

Procedure

To install a DB2 Connect server product, such as DB2 Connect Enterprise Edition, on the Solaris operating system using the DB2 Setup wizard:

1. Change to the directory where the DVD is mounted:

```
cd /db2dvd
```

where */db2dvd* represents mount point of the DVD.

2. If you downloaded the DB2 Connect product image, you must decompress and untar the product file.

- a. Decompress the product file:

```
gzip -d product.tar.gz
```

where *product* is the name of the database product that you downloaded.

- b. Untar the product file:

```
tar xvf product.tar
```

- c. Change directory:

```
cd ./product/disk1
```

Note: If you downloaded a National Language Package, untar it into the same directory. This will create the subdirectories (for example *./nlpack/disk2*) in the same directory, and allows the installer to automatically find the installation images without prompting

3. Enter the **./db2setup** command from the directory where the product image resides to start the DB2 Setup wizard. After a few moments, the IBM DB2 Setup Launchpad opens. For multiple CD installations, issue the **db2setup** command outside the mounted CD location with either a relative or absolute path name to ensure the DB2 Connect product CD can be unmounted as required. From this window, you can view the installation prerequisites and the release notes or you can proceed directly to the installation.
4. Once you have initiated the installation, proceed through the DB2 Setup wizard installation panels and make your selections. Installation help is available to guide you through the DB2 Setup wizard. Click **Help** to invoke the online help. You can click **Cancel** at any time to exit the installation. DB2 files will only be copied to your system once you have clicked **Finish** on the last DB2 Setup wizard installation panel. Once completed, the DB2 Connect server product is installed using the */opt/IBM/db2/V9.8* default installation path.

If you are installing on a system where this directory is already being used, the DB2 Connect product installation path will have *_xx* added to it, where *xx* are digits, starting at 01 and increasing depending on how many DB2 copies you have installed.

You can also specify your own DB2 database product installation path.

Results

National Language Packs can also be installed by running the `./db2setup` command from the directory where the National Language Pack resides, after a DB2 Connect product has been installed.

The installation logs, `db2setup.log` and `db2setup.err` will be located, by default, in the `/tmp` directory. You can specify the location of the log files.

If you want your DB2 database product to have access to DB2 documentation either on your local computer or on another computer on your network, then you must install the *DB2 Information Center*. The *DB2 Information Center* contains documentation for the DB2 database and DB2 related products. See the “Installing the *DB2 Information Center* using the DB2 Setup wizard (UNIX)” topic in *Installing DB2 Servers*.

Mounting CDs or DVDs for DB2 Connect (Solaris)

If the CD-ROM is not automatically mounted when you insert it into the drive on Solaris Operating System, issue the `mount` command.

Before you begin

If you are mounting the CD or DVD drive from a remote system using NFS, the CD or DVD file system on the remote computer must be exported with root access. Depending on your local system configuration, you might also need root access on the local computer.

Procedure

To mount the CD or DVD on Solaris:

1. Insert the CD or DVD into the drive.
2. If the Volume Manager (**bold**) is running on your system, the disc is automatically mounted as `/cdrom/cd_label` if the CD or DVD has a label or `/cdrom/unnamed_cdrom` if it is unlabeled.

If the Volume Manager is not running on your system, complete the following steps to mount the CD or DVD:

- a. Determine the name of the device by entering the following command:

```
ls -al /dev/sr* |awk '{print "/" $11}'
```

This command returns the name of the CD or DVD device. In this example, the command returns the string `/dev/dsk/c0t6d0s2`.

- b. Enter the following commands to mount the CD or DVD:

```
mkdir -p /cdrom/unnamed_cdrom
mount -F hsfs -o ro /dev/dsk/c0t6d0s2 /cdrom/unnamed_cdrom
```

where `/dev/dsk/c0t6d0s2` represents the name of the device that was returned in the preceding step and `/cdrom/unnamed_cdrom` represents the CD or DVD mount directory.

3. Log out.

Results

Your CD or DVD file system is now mounted. View the contents of the CD or DVD by placing the disk in the drive and enter the `cd /cdrom` command where `cdrom` is the mount point directory.

Windows

Installing a DB2 Connect server product (Windows)

To install a DB2 Connect server product, such as DB2 Connect Enterprise Edition on Windows operating systems, use the DB2 Setup wizard. Alternatively, you can install DB2 Connect server products using the response file method.

Before you begin

Before you launch the DB2 Setup wizard:

- Ensure that your system meets:
 - Disk and memory requirements
 - Hardware, distribution and software requirements. Refer to “Installation requirements for DB2 Connect server products (Windows)” on page 22.
- If you are planning to use LDAP, you must extend the directory schema. Refer to the “Extending the Active Directory Schema for LDAP directory services (Windows)” topic in *Installing DB2 Servers*.
- It is recommended that you use an Administrator account to perform the installation. The Administrator account must belong to the local administrator's group on the Windows computer where you are installing your DB2 database product and should have the following advanced user rights:
 - Act as part of the operating system
 - Create token object
 - Increase quotas
 - Replace a process level token

You can perform the installation without advanced user rights, but the setup program might be unable to validate accounts.

- If you want to install DB2 Connect with a non-Administrator account, refer to the topic “Non-Administrator installation of DB2 Connect (Windows)”.

Procedure

- To install a DB2 Connect server product, such as DB2 Connect Enterprise Edition, on Windows using the DB2 Setup wizard:
 1. Log on to the system as a user with administrator authority.
 2. Close all programs so the installation program can update files as required.
 3. Insert the DVD into the drive. The auto-run feature automatically starts the DB2 Setup wizard. The DB2 Setup wizard will determine the system language and launch the setup program for that language. If you want to run the setup program in a different language, or the setup program failed to autostart, you can run the DB2 Setup wizard manually.
 4. The DB2 Launchpad opens. From this window, you can view the installation prerequisites and the release notes, or you can proceed directly to the installation.

5. Once you have initiated the installation, proceed by following the setup program's prompts. Online help is available to guide you through the remaining steps. Click **Help** to invoke the online help. You can click **Cancel** at any time to exit the installation.

A log file stores general information and error messages resulting from the install and uninstall activities. The file name of the log follows the format *DB2-Product_Abbreviation-Date_Time.log*, such as *DB2-CEE-10-06-2006_17_23_42.log*. By default, the log file is located in the *My Documents\DB2LOG* directory.

- To invoke the DB2 Setup wizard manually:
 1. Click **Start** and select the **Run** option.
 2. In the **Open** field, enter the following command:

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

where:

- *x*: represents your DVD drive
 - *language* represents the territory code for your language (for example, EN for English).
3. Click **OK**.

What to do next

If you want your DB2 database product to have access to DB2 documentation either on your local computer or on another computer on your network, then you must install the *DB2 Information Center*. The *DB2 Information Center* contains documentation for the DB2 database and DB2 related products.

Required user accounts for installation of DB2 Connect products (Windows)

Before you begin installation tasks you must have an installation user account. During the installation, you can also choose to create one or more setup user accounts, such as a DB2 Administration Server (DAS) user account or a DB2 instance user account.

The installation user account is the account of the user performing the installation. The installation user account must be defined before running the DB2 Setup wizard. The setup user accounts can be defined before installation or you can have the DB2 Setup wizard create them for you.

All user account names must adhere to your system naming rules and to DB2 User, user ID and group naming rules.

If you use an installation user account that contains non-English characters which are not specified in DB2 naming rules, the DB2 installation will fail.

Extended security on Windows

DB2 database products offer extended Windows security. If the extended security feature is selected, you must add the users who will administer or use the DB2 database product to either the DB2ADMNS or DB2USERS group as appropriate.

The DB2 installer creates these two new groups. You can either specify a new name or accept the default names during installation.

To enable this security feature, select the **Enable operating system security** check box on the **Enable operating system security for DB2 objects** panel during the DB2 installation. Accept the default values for the DB2 Administrators Group field, and the DB2 Users Group field. The default group names are DB2ADMNS and DB2USERS. If there is a conflict with existing group names, you will be prompted to change the group names. If required, you can specify your own group names.

DB2 server user accounts

Installation user account

A local or domain user account is required to perform the installation. Normally, the user account must belong to the *Administrators* group on the computer where you will perform the installation.

Alternatively, a non-Administrator user account can be used. This alternative requires that a member of the Windows Administrators group first configure the Windows elevated privileges settings to allow a non-Administrator user account to perform an installation.

On Windows operating system, a non-administrator can perform an installation, but will be prompted for administrative credentials by the DB2 Setup wizard.

The user right "Access this computer from the network" is required for the installation user account.

The installation user ID must belong to the Domain Administrators group on the domain if the installation requires a domain account to be created or verified.

You may also use the built-in LocalSystem account as your Service Logon account for all products, except DB2 Enterprise Server Edition.

User rights granted by the DB2 installer

The DB2 installation program does not grant the Debug Programs user right. The DB2 installer grants the following user rights:

- Act as part of the operating system
- Create token object
- Lock pages in memory
- Log on as a service
- Increase quotas
- Replace a process level token

DB2 Administration Server (DAS) user account

A local or domain user account is required for the DB2 Administration Server (DAS).

Important: The DB2 Administration Server (DAS) has been deprecated in Version 9.7 and might be removed in a future release. The DAS is not supported in DB2 pureScale environments. Use software programs that use the Secure Shell protocol for remote administration. For more information, see "DB2 administration server (DAS) has been deprecated" at .

If you are performing a response file installation, you can also specify the Local System account in the response file. For more details, refer to the sample response files in the db2\windows\samples directory.

The LocalSystem account is available for all products, except DB2 Enterprise Server Edition and can be selected through the DB2 Setup wizard.

The DAS is a special DB2 administration service used to support the GUI tools and assist with administration tasks on local and remote DB2 servers. The DAS has an assigned user account that is used to log the DAS service on to the computer when the DAS service is started.

You can create the DAS user account before installing DB2 or you can have the DB2 Setup wizard create it for you. If you want to have the DB2 Setup wizard create a new domain user account, the user account you use to perform the installation must have authority to create domain user accounts. The user account must belong to the *Administrators* group on the computer where you will perform the installation. This account will be granted the following user rights:

- Act as part of the operating system
- Debug programs
- Create token object
- Lock pages in memory
- Log on as a service
- Increase quotas (adjust memory quotas for a process on Windows Server 2003 operating systems)
- Replace a process level token

If extended security is enabled, the DB2ADMNS group will have all these privileges. You can add users to that group and you do not need to add these privileges explicitly. However, the user still needs to be a member of the Local Administrators group.

The "Debug programs" privilege is only needed when DB2 group lookup is explicitly specified to use the access token.

If the user account is created by the install program, the user account will be granted these privileges and if the user account already exists, this account will also be granted these privileges. If the install grants the privileges, some of them will only be effective on first log on by the account that was granted the privileges or upon reboot.

It is recommended that the DAS user have SYSADM authority on each of the DB2 database systems within your environment so that it can start or stop other instances if required. By default, any user that is part of the *Administrators* group has SYSADM authority.

DB2 instance user account

The user account must belong to the *Administrators* group on the computer where you will perform the installation.

A local or domain user account is required for the DB2 instance because the instance is run as a Windows service and the service will be executing in the security context of the user account. When you use a domain user account to perform a database operation (such as, creating a database) against a DB2 instance, the DB2 service needs to access the domain to authenticate and search for the user's group membership. By default, a domain will only allow a domain user to query the domain and hence, the DB2 service needs to be running in the security context of a domain user.

An error will occur if you use a domain user account to perform a database operation against a DB2 service running with either a Local user account or a LocalSystem account.

You may also use the built-in LocalSystem account to run the installation for all products, except for DB2 Enterprise Server Edition.

You can create the DB2 instance user account before installing DB2 or you can have the DB2 Setup wizard create it for you. If you want to have the DB2 Setup wizard create a new domain user account, the user account you use to perform the installation must have authority to create domain user accounts. This account will be granted the following user rights:

- Act as part of the operating system
- Debug programs
- Create token object
- Increase quotas
- Lock pages in memory
- Log on as a service
- Replace a process level token

If extended security is enabled, then the DB2ADMNS group will have all these privileges. You can add users to that group and you do not need to add these privileges explicitly. However, the user still needs to be a member of the Local Administrators group.

The "Debug programs" privilege is only needed when DB2 group lookup is explicitly specified to use the access token.

If the user account is created by the install program, the user account will be granted these privileges and if the user account already exists, this account will also be granted these privileges. If the install grants the privileges, some of them will only be effective on first log on by the account that was granted the privileges or upon reboot.

Extending the Active Directory Schema for LDAP directory services (Windows)

If you plan to use the Lightweight Directory Access Protocol (LDAP) directory server feature with Windows Server 2003, you have to extend the Active Directory schema to contain DB2 object classes and attribute definitions using the **db2schex** command.

About this task

Extending the directory schema before installing DB2 database products and creating databases provide the following benefits:

- The default DB2 instance, created during the installation, is cataloged as a DB2 node in Active Directory, provided that the installation user ID had sufficient privileges to write to Active Directory.
- Any databases created after installation is automatically cataloged into Active Directory.

Procedure

To extend the directory schema:

1. Log onto any machine that is part of the Windows domain with a Windows user account that has Schema Administration authority.

2. Run the **db2schex** command from the installation DVD . You can run this command without logging off and logging on again, as follows:

```
runas /user:MyDomain\Administrator x:\db2\Windows\utilities\db2schex.exe
```

where x: represents the DVD drive letter.

What to do next

When **db2schex** completes, you can proceed with the installation of your DB2 database product; or if you have already installed DB2 database products or created databases, you have to manually register the node and catalog the databases. For more information, see the “Enabling LDAP support after DB2 installation is complete” topic.

Non-Administrator installation of DB2 Connect (Windows)

There are some additional considerations when you install DB2 Connect on Windows operating systems using a non-Administrator user account.

For a non-Administrator's installation, the account you are logged on as must belong to Power Users group.

Some information about DB2 Connect that must appear in the registry must be entered in the HKEY_CURRENT_USER folder in the registry. Although many items will be stored under the HKEY_LOCAL_MACHINE folder in the registry for non-Administrator installations of DB2 Connect, the environment settings must be changed in HKEY_CURRENT_USER.

A member of the Windows Administrators group must configure the Windows elevated privileges settings to allow a non-Administrator user account to perform an installation. For example, on a 64-bit operating system you must manually grant full permission on HKLM\Software\Wow6432Node before a 32-bit DB2 Connect product can be successfully installed.

Note: If a non-Administrator user account is going to do the product installation, then the VS2010 runtime library must be installed before attempting to install a DB2 product. The VS2010 runtime library is needed on the operating system before the DB2 product can be installed. The VS2010 runtime library is available from the Microsoft runtime library download website. There are two choices: choose `vcredist_x86.exe` for 32-bit systems or `vcredist_x64.exe` for 64-bit systems.

System shortcuts must be changed to user shortcuts for the non-Administrator install. Moreover, since services are required to install any of the DB2 Connect products, but cannot be created without administrative authority, services that would be automatically started are run as processes when a non-administrator installs.

The following scenarios are installation situations that you might encounter in an environment where both administrator and non-administrator installations exist:

- A non-Administrator has installed DB2 Connect, and then an Administrator attempts to install DB2 Connect on the same system. The Administrator will get a message that the product is already installed. The Administrator does have the authority to uninstall and reinstall the product to get around this issue.
- A non-administrator has installed DB2 Connect, and then a second non-Administrator attempts to install DB2 Connect on the same system. In this

scenario, the installation will fail, and return an error message that the user must be an Administrator to install the product.

- An Administrator has installed DB2 Connect, and then a non-Administrator attempts to install DB2 Connect on the same system. In this scenario, the install will fail, and return an error message that the user must be an Administrator to install the product. An Administrator always has the authority to uninstall or reinstall.
- Non-Administrator users cannot uninstall a DB2 product. Those non-Administrator users on a Windows operating system **can** uninstall a DB2 product.

Maintaining license keys

Registering a DB2 Connect license key using the `db2licm` command

Use the **db2licm** command to apply the license entitlement certificate (also referred to as registering a license key).

Before you begin

To complete this task, you must have the appropriate license file (*.lic).

To connect to a z/OS server or a System i server, you must register a DB2 Connect license key. (Retrieve the license file from your Passport Advantage® distribution, for example `db2conpe.lic`, then copy the license file to the license directory under the directory where the driver was installed.)

If you are using DB2 Connect Unlimited Edition for z/OS, then use a server based license key. This one step will prevent the need for client based license keys. For details, see the topic about activating the license key for DB2 Connect Unlimited Edition for System z.

On Windows operating systems, you must belong to the local Administrators or Power Users group to use the **db2licm** command with the **-a** command parameter.

Procedure

- On Windows operating systems, register a DB2 license key by entering the following command:

```
db2install_path\bin\db2licm -a filename
```

where *db2install_path* is the DB2 installation path *filename* is the full path name and file name for the license file that corresponds to the product or feature you have purchased.

- On Linux or UNIX operating systems, register a DB2 license key by entering the following command:

```
INSTHOME/sqllib/adm/db2licm -a filename
```

where *INSTHOME* represents the home directory of the instance owner and *filename* is the full path name and file name for the license file that corresponds to the product or feature you have purchased. The **db2licm** command can also be found in the path where the DB2 database product is installed. For example,

/opt/IBM/db2/V10.5/adm on AIX, HP-UX or Solaris operating systems or
/opt/ibm/db2/V10.5/adm on Linux operating systems, if you use the default
installation directory.

Setting the DB2 Connect license policy using the db2licm command

To set your license policy, issue the **db2licm** command with the command parameters that are appropriate for the license.

Before you begin

Before you set your license policy, you need to know the product identifier. To list the product identifier information, enter the following command:

```
db2licm -l
```

The product identifier is listed in the Product Identifier field.

About this task

For DB2 Connect Enterprise Edition the license policy controls and monitors the number of users that can connect simultaneously to a DB2 Connect server.

For InfoSphere Replication Server or InfoSphere Federation Server, the license policy controls and monitors the number of connectors to a data source that is not a part of DB2.

Procedure

To set your license policy:

Perform *one* of the following depending on the type of licenses that you purchased:

- If you purchased a InfoSphere Replication Server or InfoSphere Federation Server Concurrent Connector policy, enter the following command:

```
db2licm -c isrs concurrent
```

or

```
db2licm -c isfs concurrent
```

- If you purchased a DB2 Connect server Concurrent User policy, enter the following command:

```
db2licm -p db2consv concurrent
```

Post-installation tasks

Adding your user ID to the DB2ADMNS and DB2USERS user groups (Windows)

After successfully completing a DB2 installation, you now have to add users to the DB2ADMNS or the DB2USERS groups for users that need to run local DB2 applications and tools on the machine.

Before you begin

- You must have installed a DB2 database product.

- You must have selected the **Enable operating system security** check box on the Enable operating system security for DB2 object panel during the installation of your DB2 database product.

Procedure

To add users to the appropriate group:

1. Click **Start** and select **Run**.
2. Type **lusrmgr.msc** and click **OK**.
3. Select **Local Users and Groups**.
4. Select **Users**.
5. Select the user you want to add.
6. Click **Properties**.
7. Click the **Member Of** tab.
8. Click **Add**.
9. Select the appropriate group.
10. Click **OK**.

What to do next

If you did the install and chose not to enable the new security feature you can still do so post-install by running the **db2extsec.exe** command. Adding a user to a group takes effect the first time the user logs on after the user has been added. For example, if you add your user ID to the DB2ADMNS group, you need to log out and then log in again for this change to take effect.

Applying fix packs to DB2 Connect

Maintain your DB2 database environment running at the latest fix pack level to ensure problem-free operation. To install a fix pack successfully, perform all of the necessary preinstallation and post-installation tasks.

About this task

A DB2 fix pack contains updates and fixes for problems (authorized program analysis reports, or "APARs") found during testing at IBM, as well as fixes for problems that are reported by customers. For a complete list of the fixes that are contained in each fix pack, see <http://www.ibm.com/support/docview.wss?uid=swg21633303>.

Fix packs are cumulative; the latest fix pack for any given version of DB2 database contains all of the updates from previous fix packs for the same version of DB2 database.

The following types of fix pack images are available:

A single server image.

The single server image contains the new and updated code that is required for DB2 database server products, the IBM Data Server Client, IBM Data Server Runtime Client, and DB2 Connect Server. The DB2 server fix pack can service any one of the following DB2 server editions: DB2 Enterprise Server Edition, DB2 Workgroup Server Edition, DB2 Express[®] Server Edition, DB2 Connect Enterprise Edition, DB2 Connect Application Server Edition, DB2 Connect Unlimited Edition for zSeries, and DB2

Connect Unlimited Edition for i5/OS™). The Data Server Client fix pack is contained within the one DB2 database server fix pack. You can use the DB2 database server fix pack to update Data Server Client.

A single server image can also be used to install any of the DB2 database server products, at a particular fix pack level, with a DB2 try and buy license by default.

The single server fix pack image contains DB2 try-and-buy licenses for all DB2 server products. When you select a new DB2 server product to install or a previously installed DB2 server product to update, the try-and-buy licenses of that particular product is installed. The try-and-buy licenses do not affect any valid licenses that are already installed in the same DB2 installation path.

A fix pack for each of the other DB2 database products

Use this fix pack only for installed non-server database products . For example, IBM Data Server Runtime Client.

Do not use this type of fix pack if the installed DB2 database products are only DB2 database server products or a Data Server Client. Instead, use the single server image fix pack.

For Windows platforms, if you have more than one DB2 database product (which includes at least one product that is not a Data Server Client or a DB2 database server) installed in a single DB2 copy, you must download and uncompress all of the corresponding product-specific fix packs before you start the fix pack installation process.

A universal fix pack

The universal fix pack services installations where DB2 database products and DB2 add-on products are installed in the same path.

The universal fix pack is not needed if the installed DB2 database products are only DB2 database server products or a Data Server Client. In this case, use the single server image fix pack.

On Linux or UNIX operating systems, if national languages are installed, you also require a separate national language fix pack. The national language fix pack cannot be installed alone. A universal or product-specific fix pack must be applied at the same time and they must both be at the same fix pack level. For example, if you are applying a universal fix pack to non-English DB2 database products on Linux or UNIX, you must apply both the universal fix pack and the national language fix pack to update the DB2 database products.

On IBM DB2 pureScale environments, a fix pack image can be applied offline or online. On DB2 environments other than DB2 pureScale environments, a fix pack image can be only applied offline.

Restrictions

- A DB2 Version 10.5 fix pack can be applied only to DB2 Version 10.5 general availability (GA) or DB2 Version 10.5 fix pack copies.
- All DB2 instances, DAS, and applications that are related to the DB2 copy that is being updated must be stopped before you install a fix pack. However, in a DB2 pureScale environment, the DB2 pureScale instance can remain running for online fix pack updates.
- In a partitioned database environment, before you install the fix pack, you must stop the database manager on all database partition servers. You must install the

fix pack on the instance-owning database partition server and all other database partition servers. All of the computers that are participating in the instance must be updated to the same fix pack level.

- On Linux or UNIX operating systems:
 - If you have DB2 database products on a network file system (NFS), you must ensure that the following applications are stopped completely before you install the fix pack: all instances, the DB2 administration server (DAS), interprocess communications (IPC), and applications on other machines that use the same NFS mounted installation.
 - If the system commands **fuser** or **lsof** are not available, the **installFixPack** command cannot detect loaded DB2 database files. You must ensure that no DB2 files are loaded and provide an override option to install the fix pack. On UNIX, the **fuser** command is required to check for loaded files. On Linux, either the **fuser** command or **lsof** command is required.
For details on the override option, see the **installFixPack** command.
- On client applications, after a fix pack is applied, you must have bind authority to perform autobind of applications.
- DB2 fix packs do not update IBM Data Studio.

Procedure

To install a fix pack:

1. Check fix pack prerequisites.
2. Perform the tasks in the "Preparing to install a fix pack" topic.
3. Choose a fix pack installation method and install the fix pack.
4. Perform the tasks in the "after installing the fix pack" topic.
5. Apply the appropriate DB2 database product license.

If a previously licensed copy of a DB2 database server product does not exist on the machine, a single server fix pack image can be used to install any of the DB2 database server products. In this case, the DB2 database product that is installed is treated as a try and buy license, and stops working after a 90 day trial period unless you upgrade the try and buy license.

What to do next

Check the log file for any post-installation steps, or error messages and recommended actions.

For non-root installations on Linux or UNIX, root-based features (such as high availability and operating system-based authentication) can be enabled using the **db2rfe** command. If root-based features were enabled after installing your DB2 database product, you must rerun the **db2rfe** command each time a fix pack is applied in order to re-enable those features.

In an environment that is not a DB2 pureScale environment, if you have multiple DB2 copies on the same system, those copies can be at different version and fix pack levels. If you want to apply a fix pack to one or more DB2 copies, you must install the fix pack on those DB2 copies one by one.

Uninstalling

Uninstalling DB2 Connect (Windows)

This task provides steps for completely removing your DB2 database product from your Windows operating system. Only perform this task if you no longer require your existing DB2 instances and databases.

About this task

If you are uninstalling the default DB2 copy, and you have other DB2 copies on your system, use the **db2swtch** command to choose a new default copy before you proceed with the uninstallation. Also, if your DB2 Administration Server (DAS) is running under the copy being removed, move your DAS to a copy that is not being removed. Otherwise, re-create the DAS using the **db2admin create** command after the uninstall, and you reconfigure the DAS for some function to work.

Procedure

To remove your DB2 database product from Windows:

1. Optional: Drop all databases using the **drop database** command. Be sure that you no longer need these databases. If you drop your databases, all of your data will be gone.
2. Stop all DB2 processes and services. This can be done through the Windows Services panel or by issuing the **db2stop** command. If DB2 services and processes are not stopped before attempting to remove your DB2 database product, you will receive a warning containing a list of processes and services that are holding DB2 DLLs in memory. If you will use Add/Remove Programs to remove your DB2 database product, this step is optional.
3. You have two options for removing your DB2 database product:
 - Add/Remove Programs
Accessible through the Windows Control Panel, use the Add/Remove Programs window to remove your DB2 database product. Refer to your operating system's help for more information about removing software products from your Windows operating system.
 - **db2unins** command
You can run the **db2unins** command from the *DB2DIR\bin* directory to remove your DB2 database products, features, or languages. Using this command, you can uninstall multiple DB2 database products at the same time using the **/p** parameter. You can use a response file to uninstall DB2 database products, features, or languages using **/u** parameter.

What to do next

Unfortunately, your DB2 database product cannot always be removed by using the **Control Panel > Add/Remove Programs** facility or using the **db2unins /p** command or the **db2unins /u** command. The following uninstallation option must ONLY be attempted if the previous method fails.

To forcefully remove all DB2 copies from your Windows system, run the **db2unins /f** command. This command will perform a brute force uninstallation of ALL DB2 copies on the system. Everything except user data, such as DB2 databases, will be forcefully deleted. Before running this command with the **/f** parameter, see the **db2unins** command for details.

Uninstalling DB2 Connect (Linux and UNIX)

This task provides steps for removing a DB2 database product from your Linux or UNIX operating system.

About this task

This task is not required to install a new version of a DB2 database product. Each version of a DB2 database product on Linux or UNIX has a different installation path and can therefore coexist on the same computer.

Note: This task applies to DB2 database products that were installed with root user authority. A separate topic explains how to uninstall DB2 database products that were installed as a non-root user.

Procedure

To remove your DB2 database product:

1. Optional: Drop all databases. You can drop databases using the **DROP DATABASE** command. Database files remain intact on your file systems when you drop an instance without dropping databases first.
2. Stop the DB2 Administration Server. Refer to the *Installing DB2 Servers* manual.
3. Remove the DB2 Administration Server, or run the **dasupdt** command to update the DB2 Administration Server to another installation path. To remove the DB2 Administration Server, refer to the *Installing DB2 Servers* manual.
4. Stop all DB2 instances. Refer to the *Installing DB2 Servers* manual.
5. Remove the DB2 instances, or run the **db2iupdt** command to update the instances to another installation path. To remove the DB2 instances, refer to the *Installing DB2 Servers* manual.
6. Remove the DB2 database products. Refer to the *Installing DB2 Servers* manual.

Chapter 3. Upgrading to the latest version of DB2 Connect

Upgrading to a new version or release of DB2 Connect might require upgrading your environment components if you want them to run on the new release. These components are DB2 Connect servers, DB2 servers, DB2 clients, and database applications.

For example, if you have an existing environment using an earlier version or release of DB2 Connect and you want to install the latest version or release of DB2 Connect, then you can upgrade your DB2 Connect server and you might need to upgrade other components in your environment.

DB2 Connect servers supports the upgrading of DB2 Connect instances, and any existing transaction manager and DB2 Connect federated databases created on previous versions of DB2 Connect servers.

The upgrade process consists of all the tasks that you need to perform to have your environment running successfully on a new release. The upgrading of each of the components in your environment to the latest version or release of DB2 Connect requires that you perform different tasks:

- “Upgrading DB2 Connect servers” on page 58 involves upgrading your existing instances, any existing DB2 Connect federated databases, and any existing transaction manager databases so that they can run in the latest version or release of DB2 Connect.
- Upgrading IBM Data Server client packages involves upgrading your client instances to keep the configuration of your existing IBM Data Server client packages. Refer to the “Clients upgrade” topic in the *Upgrading to DB2 Version 10.5*.
- Upgrading database applications involves testing them in the latest version or release of DB2 Connect and modifying them only when you need to support changes available in the latest version or release of DB2 Connect.

Review changes in existing functionality and discontinued and deprecated functionality for DB2 Connect in “DB2(r) enhancements and changes that affect DB2 Connect(tm)” in *What’s New for DB2 Version 10.5* to determine the changes that could impact your database applications. If your database applications connect to DB2 servers, you might need to upgrade your database applications. Refer to the “Database applications and routines upgrade” topic in the *Upgrading to DB2 Version 10.5*.

- Consideration toward DB2 Connect client, instead of DB2 Connect server, to receive equivalent or superior function. You can reduce complexity, improve performance, and deploy application solutions with smaller footprints. For details, see the topic about client/server connection options.

The best approach to upgrading is to write an upgrade plan. A strategy defines how to approach the upgrading of your environment and gives you the outline for your upgrade plan. The characteristics of your environment and the information in upgrade essentials, especially the upgrade recommendations and restrictions, can help you determine your strategy. An upgrade plan should include the following upgrade details for each component:

- Upgrade prerequisites that indicate all the requirements that you need to meet before upgrading.

- Pre-upgrade tasks which describe all the preparation tasks that you need to perform before upgrading.
- Upgrade tasks which describe step by step the basic upgrade process for a component and how to upgrade environments with special characteristics.
- Post-upgrade tasks which describe all the tasks that you need perform after upgrading to have your DB2 server running at the optimum level.
- Review the need to opt for DB2 Connect client, instead of DB2 Connect server, to receive equivalent or superior function.

You will find that pre-upgrade tasks, upgrading tasks, and post-upgrade tasks for DB2 Connect servers reference pre-upgrade tasks, upgrading tasks, and post-upgrade tasks for DB2 servers because they are exactly the same tasks.

Upgrade essentials for DB2 Connect

If you are upgrading your clients to the latest version or release of DB2 Connect, you need to consider the changes in support and resolve them before you upgrade.

Upgrade essentials for DB2 servers and clients also apply to DB2 Connect servers

Upgrade support and restrictions for DB2 servers and clients also apply when you upgrade your DB2 Connect server.

- Review upgrade essentials for DB2 servers to determine additional changes that impact your upgrade and how to address any issues. Refer to the “Upgrade essentials for DB2 Servers” topic in *Upgrading to DB2 Version 10.5*.
- Review upgrade essentials for clients, especially connectivity support between clients and DB2 servers. Connections to the latest version or release of DB2 Connect servers from a client release two or more versions earlier are not supported. Refer to the “Upgrade essentials for clients” topic in *Upgrading to DB2 Version 10.5*.
- Review the need to opt for DB2 Connect client, instead of DB2 Connect server, to receive equivalent or superior function. You can reduce complexity, improve performance, and deploy application solutions with smaller footprints. For details, see the topic about client/server connection options.

Upgrade recommendations for DB2 Connect

The last two versions of the clients can connect to the latest version or release of DB2 Connect servers. The only restriction is that new features are not available to the clients from the previous versions and releases. However, it is not likely that you need access to these new features because your existing applications do not use them.

If you choose to upgrade your clients first, you need to be aware that there are known limitations about the support for connectivity from a current version or release of the client to DB2 Connect servers from two versions ago. Check the current version or release of the incompatibilities with previous releases, see if these limitations apply to your application in order to take necessary actions.

Perform the pre- and post-upgrade tasks to ensure a successful upgrade.

Pre-upgrade tasks for DB2 Connect servers

To successfully upgrade your DB2 Connect servers, preparation is required to address any issues that may exist.

Procedure

Perform the following pre-upgrade tasks for DB2 servers that also apply to DB2 Connect servers:

1. Review the “Upgrade essentials for DB2 Connect” on page 56 to identify the changes or restrictions that can affect your upgrade and learn how to address any issues before upgrading.
2. If the modification level of your product is higher than 10, install DB2 for z/OS APAR PM35785 on your z/OS system before upgrading to a new release or fix pack of DB2 Connect.
3. Refer to the “Backing up DB2 server configuration and diagnostic information” topic in *Upgrading to DB2 Version 10.5* to have a record of your current configuration that you can compare with the configuration after the upgrade. You can also use this information to create new instances or databases using the same configuration that you had before upgrading.
4. Optional: If you enabled the Syncpoint Manager (SPM) functionality on your DB2 Connect server, ensure that the DRDA sync point managers do not contain any indoubt transactions by using the **LIST DRDA INDOUBT TRANSACTIONS** command to get a list of indoubt transactions and to interactively resolve any indoubt transactions.
5. Optional: If you have transaction manager databases, perform the following pre-upgrade tasks to prepare your databases for upgrading:
 - a. Ensure that the database to be upgraded does not contain any indoubt transactions by using the **LIST INDOUBT TRANSACTIONS** command to get a list of indoubt transactions and to interactively resolve any indoubt transactions.
 - b. Refer to the “Verify that your databases are ready for upgrading” topic in the *Upgrading to DB2 Version 10.5* to identify and resolve any problems before the actual upgrade.
 - c. Refer to the “Backing up databases before upgrading” topic in the *Upgrading to DB2 Version 10.5* to be able to upgrade them to a new upgraded system or restore them in the original pre-upgrade system.
 - d. Review the “disk space requirements” topic in the *Upgrading to DB2 Version 10.5* to ensure that you have enough free disk space, temporary table space and log space for database upgrading and increase table space and log file sizes if necessary.
 - e. Linux only: Review the “Changing raw devices to block devices (Linux)” topic in the *Upgrading to DB2 Version 10.5*.
6. Optional: If you have DB2 Connect federated databases, refer to the “Preparing to migrate to federated systems” topic in the *IBM WebSphere Information Integration: Migrating to Federation Version 9* for details on pre-upgrade tasks for these databases.
7. Windows only: If you obtained customized code page conversion tables from the DB2 support service, you need to backup all of the files in the *DB2OLD\conv* directory where *DB2OLD* is the location of your existing DB2 Connect copy. Upgrading your current version or release of DB2 Connect copy

removes these tables because standard code page tables are contained in a new version or release DB2 Connect library. You do not need to backup standard code page conversion tables.

8. Optional: Upgrade your DB2 Connect server in a test environment to identify upgrade issues and to verify that database applications and routines work as expected before upgrading your production environment.
9. If the **diaglevel** database manager configuration parameter is set to 2 or less, set it to 3 or higher before upgrading.
Refer to the “Setting the diagnostic log file error capture level” topic in the *Troubleshooting and Tuning Database Performance* to set this database manager configuration parameter.
In the latest version or release of DB2 Connect, all significant upgrade events are logged in the **db2diag** log files when the **diaglevel** database manager configuration parameter is set to 3 (default value) or higher.
10. Take the DB2 Connect server offline for upgrading. For details, refer to the “Taking a DB2 server offline before upgrading” topic in the *Upgrading to DB2 Version 10.5*.

Upgrading DB2 Connect servers

DB2 Connect Version 10.5 servers support the upgrade of DB2 Connect instances, and any existing transaction manager and DB2 Connect federated databases created on DB2 Connect Version 9.7 and Version 9.5 servers.

Before you begin

Before upgrading to DB2 Connect Version 10.5:

- Ensure that you have the proper operating system access:
 - Root user authority on UNIX
 - Local Administrator on Windows
- Ensure that you have SYSADM authority.
- Ensure that you meet the installation requirements for DB2 database products. Refer to the “Installation requirements for DB2 database products” topic in the *Installing DB2 Servers*. The requirements for Linux and UNIX operating systems have changed.
- Review the upgrade recommendations. Refer to the “Best practices for upgrading DB2 Servers” topic in the *Upgrading to DB2 Version 10.5*.
- Review the disk space requirements. Refer to the “Disk space requirements for DB2 Server upgrades” topic in the *Upgrading to DB2 Version 10.5*.
- Perform the pre-upgrade tasks, especially backing up your databases.

About this task

Since DB2 Connect server products are host database connectivity servers, the only databases that can exist within a DB2 Connect server instance are transaction manager databases and DB2 Connect federated databases. The DB2 Connect transaction manager database stores transaction state information for DB2 coordinated transactions. The sole purpose of DB2 Connect federated databases is to contain information about data sources.

On Linux and UNIX operating systems, you should manually upgrade your DB2 Connect instances after installing the latest version of DB2 Connect. All the remote nodes and databases that you cataloged on the DB2 clients refer to these instances.

If you create a new instance, again you will have to catalog nodes, DCS databases, and databases on the DB2 clients that existed in the instances from the previous version.

On Windows operating systems, you have an option to automatically upgrade an existing, supported DB2 Connect copy during installation. Your DB2 Connect instances are automatically upgraded. Alternatively, you can install a new copy of the latest version of DB2 Connect and then manually upgrade your DB2 Connect instances.

This procedure describes how to upgrade by installing a new copy of the latest version of DB2 Connect and then upgrade instances and any existing databases. To automatically upgrade an existing, supported DB2 Connect copy on Windows, refer to “Upgrading a DB2 server (Windows)” in the *Upgrading to DB2 Version 10.5*.

Restrictions

- The bit size of the client instance is determined by the operating system where you install DB2 Connect. Refer to the “Support changes for 32-bit and 64-bit DB2 servers” topic in the *Upgrading to DB2 Version 10.5* for details.
- Additional upgrade restrictions for DB2 servers also apply to DB2 Connect servers. Refer to the “Upgrade restrictions for DB2 servers” topic in the *Upgrading to DB2 Version 10.5*.

Procedure

To upgrade your DB2 Connect server Version 10.5:

1. Export your connectivity configuration information for your existing, supported DB2 Connect server to an export profile. Use the **db2cfexp** tool to create a configuration profile:
`db2cfexp cfg_profile backup`

This profile contains all of the instance configuration information, including the database manager configuration and registry profile because the option **backup** is specified. You can use this profile to re-create your connectivity configuration if necessary.

2. Install DB2 Connect by running the DB2 Setup wizard and selecting the option **Install New** on the Install a Product panel. Refer to “DB2 Connect server products: installation and configuration overview” on page 30.
3. Upgrade your DB2 Connect instances using the **db2iupgrade** command. Refer to the “Upgrading instances” topic in the *Upgrading to DB2 Version 10.5*.
4. Upgrade any existing transaction manager and DB2 Connect federated databases. You can also upgrade your databases by restoring a DB2 Connect backup from one of the two previous supported versions. Upgrade any existing transaction manager and DB2 Connect federated databases by referring to the “Upgrading databases” topic in the *Upgrading to DB2 Version 10.5*.

What to do next

After upgrading the DB2 Connect server, perform the recommended post-upgrade tasks such as resetting the diagnostic error level, adjusting log space size, and rebinding packages, and verifying that your upgrade was successful. Refer to “Post-upgrade tasks for DB2 Connect servers” on page 60.

Post-upgrade tasks for DB2 Connect servers

After upgrading your DB2 Connect servers, you should perform several post-upgrade tasks to ensure that your DB2 Connect servers perform as expected and run at their optimum level.

Procedure

Perform the following post-upgrade tasks for DB2 servers that also apply to DB2 Connect servers:

1. If you set the **diaglevel** database manager configuration parameter to 4 as recommended in the pre-upgrade tasks for DB2 Connect servers, reset this parameter to the value set before the upgrade.
2. Manage changes in DB2 server behavior. Refer to the “Manage changes in DB2 server behavior” topic in the *Upgrading to DB2 Version 10.5*. There are new registry variables, new configuration parameters, and new default values for registry variables and configuration parameters introduced in latest version or release of DB2 database products that can impact the behavior of the DB2 database server. There are also changes in physical design characteristics of databases and changes to security that also have an impact.
3. If you obtained customized code page conversion tables from the DB2 support service for previous versions or releases, copy all of the files for those tables from the *DB2OLD/conv* to *DB2DIR/conv*, where *DB2OLD* is the location of your previous supported version of DB2 Connect copy and *DB2DIR* is the location of your new DB2 Connect copy. You do not need to copy standard code page conversion tables.

If you upgraded your existing, supported DB2 Connect copy on Windows operating systems, you can restore the customized code page conversion tables that you backed up as part of the pre-upgrade tasks for DB2 Connect servers to the *DB2PATH\conv* directory, where *DB2PATH* is the location of your new DB2 Connect copy.

4. If you are connecting to a DB2 for z/OS server or a IBM DB2 for IBM i server where euro support is required, set the **DB2CONNECT_ENABLE_EURO_CODEPAGE** registry variable to YES on all DB2 Connect clients and servers so that the current application code page is mapped to the equivalent coded character set ID (CCSID) that explicitly indicates support for the euro sign.
5. Optional: If you upgraded any databases in your DB2 Connect server and changed the log space setting as recommended in the pre-upgrade tasks for DB2 Connect servers, adjust the log space size. Refer to the “Adjusting the log space size in migrated databases” topic in the *Upgrading to DB2 Version 10.5*. Ensure that the amount of log space that you allocate is adequate for your DB2 Connect server.
6. Optional: Back up your databases after the upgrade is complete. Refer to the “Backing up databases before upgrading” topic in the *Upgrading to DB2 Version 10.5*.
7. Optional: If you have DB2 Connect federated databases, review the “Configuring federated systems after migration” topic in *IBM WebSphere Information Integration: Migrating to Federation Version 9* to determine if you need to perform any tasks after you upgrade your federated databases.
8. Verify that your DB2 Connect server upgrade was successful. Test connections to all your cataloged databases. The following example shows how to test a connection from the Command Line Processor (CLP):

```
db2 CONNECT TO DATABASE sample user mickey using mouse
```

You need to specify a user and password when connecting to a remote database. Ensure all connections are successful.

Also, test your applications and tools to ensure that the DB2 Connect server is working as expected.

What to do next

At this point, you should resume all of your maintenance activities. You should also remove any previously supported versions or releases of DB2 Connect copies that you no longer need.

Chapter 4. Configuring

Preparing IBM DB2 for IBM i for connections from DB2 Connect

DB2 Connect gives remote system applications access to data on your IBM DB2 for IBM i system.

Procedure

To set up the connection, you need to know the following information:

1. The local network name. You can get this information by entering **DSPNETA**.
2. The local adapter address. You can get this information by entering the **WRKLIND** command in one of the following ways:

WRKLIND (*elan)

Lists Ethernet adapters

WRKLIND (*trlan)

Lists token ring adapters

WRKLIND (*all)

Lists all adapters

3. The hostname. You can get this information by entering **DSPNETA**.
4. The TCP/IP port or service name. The default is X'07'6DB (X'07F6C4C2'). The default is always used by DB2 for i. If entering a hexadecimal number is not convenient, an alias is QCNTEDDM.
5. The relational database name. You can get this information by entering **DSPRDBDIRE**. This will display a list. The line containing *LOCAL in the Remote Location column identifies the RDBNAME which must be defined to the client. If there is no *LOCAL entry, you can add one, or use the system name obtained from the **DSPNETA** command on the server.

Results

Here is an example:

Display Relational Database Directory Entries

Position to

Type options, press Enter.
5=Display details 6=Print details

	Relational	Remote
Option	Database	Location Text
-		
-	DLHX	RCHAS2FA
-	JORMT2FA	JORMT2FA
-	JORMT4FD	JORMT4FD
-	JOSNAR7B	RCHASR7B
-	RCHASR7B	*LOCAL
-	RCHASR7C	RCHASR7C
-	R7BDH3SNA	RCH2PDH3
-	RCHASDH3	RCHASDH3

When you have obtained these parameters from your IBM Power Systems server, enter your values into the worksheet that follows:

Table 7. Configuration parameters from IBM Power Systems

Item	Parameter	Example	Your value
A-1	Local network name	SPIFNET	
A-2	Local adapter address	400009451902	
A-4	Hostname	SYD2101A	
A-5	TCP/IP port or service name	X'07F6C4C2' (default)	
A-6	Relational database name	NEW_YORK3	

For more information, refer to the “DRDA Considerations” section of the *DB2 Server for VSE & VM SQL Reference* (SC09-2989).

Preparing DB2 for z/OS for connections from DB2 Connect

DB2 Connect gives remote system applications access to data on your DB2 for z/OS system.

Before you begin

If you anticipate that DB2 for z/OS will participate in a multisite update transaction (two-phase commit) then refer to the topic that discusses enabling multisite updates in the *DB2 Connect User's Guide*.

About this task

This topic provides instructions for establishing TCP/IP network connections between DB2 Connect Server or DB2 Connect client and DB2 for z/OS.

Procedure

To prepare DB2 for z/OS to receive connection requests from DB2 Connect, you need to configure your protocol by:

- “Configuring TCP/IP for DB2 for z/OS”
-
- “Configuring DB2 for z/OS” on page 68

Host databases

A host database is a relational database system from which a link request originates.

The term *database* is used throughout this document to describe a relational database management system (RDBMS). Other systems with which DB2 Connect communicates might use the term database to describe a slightly different concept. The DB2 Connect term database can also refer to:

System z

DB2 for z/OS. A DB2 for z/OS subsystem identified by its LOCATION NAME. Use the z/OS **-display ddf** command to get the DB2 server location name, domain name, IP address and port.

A DB2 for z/OS location is the unique name of a database server. An application uses the location name to access a DB2 for z/OS subsystem or a DB2 for z/OS data sharing group. A data sharing group enables applications on different DB2 subsystems to read from and write to the same data concurrently. The application uses a DB2 data sharing group network address to access a DB2 data sharing location. The accessed DB2 subsystem is transparent to the application.

Since DB2 for z/OS supports multiple databases at the same DB2 location, the location name is analogous to a Linux, UNIX, and Windows database alias name. A database alias can be used to override the location or location alias name when accessing a location. A location alias is another name for a location. It is used to control which subsystems in a data sharing group are accessed by an application.

LOCATION NAME is also defined in the Boot Strap Data Set (BSDS) as well as the DSNL004I message (LOCATION=location), which is written when the Distributed Data Facility (DDF) is started. LOCATION NAME supports up to 8 alias location names, allowing applications the ability to use different dbalias names to access a Version 8 z/OS server.

IBM Power Systems Servers

IBM DB2 for IBM i, an integral part of the IBM i operating system. Only one database can exist on an IBM Power Systems server unless the system is configured to use independent auxiliary storage pools.

Configuring TCP/IP for DB2 for z/OS

To configure TCP/IP communications between your DB2 Connect workstation and DB2 for z/OS Version 8 or later, you must first collect network details about the host database server.

Before you begin

The instructions assume the following conditions:

- You are connecting to a single host database server or location via TCP/IP. Multiple host connections will be handled in exactly the same way, although the *port number* and *service number* required in each case might be different. Use the group IP address to connect to a group location.
- The target database resides on DB2 for z/OS Version 8 or later.
- All the necessary software prerequisites are installed.
- DB2 clients have been set up as required.

Procedure

1. Before you can use DB2 Connect over a TCP/IP connection, you must collect information about both the host database server and the DB2 Connect server. For each host server that you are connecting to via TCP/IP, you must have the following information:

- The location of the TCP/IP services and hosts files at the DB2 Connect workstation:

On UNIX and Linux

/etc/

On Windows Server 2003

Usually %SystemRoot%\system32\drivers\etc\, where %SystemRoot% represents the Windows install path directory.

You might want to add the host information to a *domain name server* to avoid maintaining this file on multiple systems.

- The locations of the equivalent files at the target DB2 for z/OS host.
- The TCP/IP port number defined to DB2 for z/OS.

Note: The associated service name information is not exchanged between the DB2 Connect workstation and DB2 for z/OS.

Port number 446 has been registered as the default for communication from a DB2 Connect workstation.

- The TCP/IP addresses and host names for both the host and the DB2 Connect workstation.
 - The LOCATION NAME of the DB2 for z/OS database server.
 - The user ID and password to be used when issuing CONNECT requests to the database at the IBM mainframe server.
2. Refer to your local network administrator and your DB2 for z/OS administrator for help getting this information. Use the tables that follow as a worksheet to plan *each* TCP/IP connection between DB2 Connect and a host database server.

Table 8. User Information

Ref.	Description	Sample Value	Your Value
TCP-1	User name	A.D.B.User	
TCP-2	Contact info	(123)-456-7890	
TCP-5	User ID	ADBUSER	
TCP-6	Database type	db2390	

Table 8. User Information (continued)

Ref.	Description	Sample Value	Your Value
TCP-7	Connection type (must be TCPIP).	TCPIP	TCPIP

Table 9. Network Elements at the Host

Ref.	Description	Sample Value	Your Value
TCP-8	Host name	MVSHOST	
TCP-9	Host IP address	9.21.152.100	
TCP-10	Service name	db2inst1c	
TCP-11	Port number	446	446
TCP-12	LOCATION NAME	NEW_YORK3	
TCP-13	User ID		
TCP-14	Password		

Note:

- To obtain the host's IP address TCP-9, enter at the host:
TSO NETSTAT HOME
- To obtain the port number TCP-11, look for DSNL004I in the DB2 master address space or system log.

Table 10. Network Elements at the DB2 Connect client and server

Ref.	Description	Sample Value	Your Value
TCP-18	Host name	mcook02	
TCP-19	IP address	9.21.27.179	
TCP-20	Service name	db2inst1c	
TCP-21	Port number	446	446

Table 11. DB2 Directory Entries at the DB2 Connect server

Ref.	Description	Sample Value	Your Value
TCP-30	Node name	MVSIPNOD	
TCP-31	Database name	nyc3	
TCP-32	Database alias	mvsipdb1	
TCP-33	DCS database name	nyc3	

- Complete a copy of the worksheet example for each TCP/IP host:
 - Fill in the values to be used for the host name and IP address of the DB2 for z/OS host (TCP-8 and TCP-9).
 - Fill in the values to be used for the host name and IP address of the DB2 Connect workstation (TCP-18 and TCP-19).
 - Determine the service name or port number to be used for the connection (TCP-10 or TCP-20, or TCP-11 or TCP-21).
 - Determine the LOCATION NAME of the DB2 for z/OS database server to which you want to connect.
 - Determine the values to be used for user ID and PASSWORD when connecting to the host database.

4. At your System z server:
 - a. Verify the host address or the host name.
 - b. Verify the port number or the service name.
 - c. Update the services file with the correct port number and service name if necessary.
 - d. Update the hosts file (or the Domain Name Server used by the DB2 for z/OS system) with the host name and IP address of the DB2 Connect workstation if necessary.
 - e. Ensure the new definitions are active before attempting to test the connection. Refer to your host network administrator or change control staff if necessary.
 - f. Check with the DB2 for z/OS administrator that you have a valid user ID, password, and database *LOCATION NAME*.
 - g. PING the DB2 Connect server, using the correct port number if that option is supported by TCP/IP on the host system. For example:

```
ping remote_host_name -p port_number
```

Support for your System z server is available at <http://www.ibm.com/servers/eserver/support/zseries/>

Configuring DB2 for z/OS

Before you can use DB2 Connect, your DB2 for z/OS Administrator must configure DB2 for z/OS to permit connections from DB2 Connect workstations.

About this task

This section indicates the *minimum* updates required to permit a DB2 Connect client to make a connection to the DB2 for z/OS database server. For more detailed examples, refer to the DB2 for z/OS installation documentation: <http://publib.boulder.ibm.com/infocenter/imzic> or refer to the DDF installation steps in the DB2 for z/OS installation manual.

Preparing DB2 for VSE & VM for connections from DB2 Connect

You can set up a DB2 Server for VSE and VM as an application server.

About this task

For information about how to set up DB2 Server for VM and VSE as an application server, refer to the “DRDA Considerations” section of the *DB2 Server for VSE & VM SQL Reference* (SC09-2989) .

Sysplex support

Applications can leverage Sysplex capabilities by either going through a middle-tier DB2 Connect server, or using client Sysplex support, when available.

The client sysplex support is the preferred option as it provides better availability, improved server utilization since it eliminates a point of failure, transaction level balancing and seamless automatic client reroute, whereas DB2 Connect server does not.

DB2 Connect server Sysplex support

Sysplex permits the DB2 Connect server to seamlessly balance connections across different members of a data sharing group. A Sysplex is a collection of System z servers that cooperate, using hardware and software, to process work.

The Sysplex coordinates the cooperation by increasing the number of processors working together, which increases the amount of work that can be processed. In addition to an increase in processing capability, a Sysplex can provide flexibility in mixing levels of hardware and software, and in dynamically adding systems.

Sysplex also provides DB2 Connect server the means to try alternate members should a failure occur with one member. The rerouting capability for Sysplex is a DB2 Connect feature. DB2 Connect server support for Sysplex is enabled by default and so is the rerouting capability for Sysplex. Sysplex support to a host database can be turned off by removing the **SYSPLEX** parameter from its DCS directory entry, but the DCS entry itself should not be removed, even if it has no other parameter specified.

With the automatic client reroute capability for Sysplex, the default behavior is for a Sysplex enabled connection to try connecting again when there is a communication failure. Special register values, up until the last successful transaction not holding resources, are replayed when DB2 Connect is connected to a DB2 for z/OS server.

You can configure the exact automatic client reroute retry behavior, including disablement, by using the **DB2_MAX_CLIENT_CONNRETRIES** and **DB2_CONNRETRIES_INTERVAL** registry variables. The connection timeout registry variable is **DB2TCP_CLIENT_CONTIMEOUT**.

Considerations for System z SYSPLEX exploitation

DB2 Connect provides load balancing and fault-tolerance when routing connections to DB2 Sysplex. When connected to a DB2 for z/OS database server running in a DB2 pureScale environment, DB2 Connect will spread the workload amongst the different DB2 subsystems comprising the data sharing group, based on the system load and health information provided by the Workload Manager (WLM). It uses the Distributor to route connections. Use the group IP address to connect to a group location.

DB2 Connect receives a prioritized list of DB2 members from the WLM. Each Sysplex returns weighted priority information for each connection address that has capacity to run the work. This list is then used by DB2 Connect to handle the incoming CONNECT requests by distributing them among the DB2 members with the best capacity to run work. For load balancing, the list of Sysplex weighted priority information is obtained during each connection. This list is also used when determining where to send each transaction.

Note: System z Distributed Data Facility (DDF) configuration does not need to be changed to take advantage of the DB2 Connect Sysplex exploitation. Refer to DB2 for z/OS Data Sharing Planning and Administration guide.

DB2 Connect also provides fault-tolerance by attempting to connect to an alternate sysplex machine in the event of a connection failure. An error will only be returned to the application if all known connections have failed.

DB2 Connect is designed with a transport tool. With Sysplex enabled, DB2 Connect routes connections using a transport member and associates it with a logical connection.

DB2 Sysplex exploitation

You can exploit DB2 Sysplex to ensure fault tolerance if connections to the database fail.

In a typical scenario, a DB2 Connect server (server A) would be in conversation with a Sysplex containing two DB2 for z/OS servers (servers B and C).

Sysplex server B	Sysplex server C
HOST_NAME=MVSHOST	HOST_NAME=MVSHOST1

Suppose that in this scenario an application now issues:

```
db2 connect to aliasb user xxxxxxxx using xxxxxxxx
```

The connection to database MVSHOST is established. Because Sysplex exploitation is enabled both for the DB2 Connect server and the DCS directory entry, DB2 for z/OS identifies the network addresses to DB2 Connect for each Sysplex participant (MVSHOST and MVSHOST1. DRDA4 protocols and message flows are used to return this information). Once an initial connection has been made, the returned list of addresses is cached at the DB2 Connect workstation. Once the initial CONNECT is issued for a TCP/IP node, then the IP addresses are returned.

Priority information used for load balancing and fault tolerance

The list of addresses provided by DB2 for z/OS also includes priority information, including the number of connections for each network address. The list is refreshed whenever a new connection is made by DB2 Connect. This additional information is used for load balancing purposes, as well as for fault tolerance.

Cached Address List used by DB2 Connect

If the database connection to ALIASB fails, then an error message SQL30081N is issued, and the connection will be dropped. If a further connection request is received for ALIASB, DB2 Connect performs the following actions:

1. It tries the highest priority server from the cached list of addresses based on the priority information that was returned by DB2 for z/OS. This strategy is always used by DB2 Connect, and it is by this means that load balancing is achieved.
2. If this connection attempt fails, then the other addresses in the list are tried, in descending order of priority, as returned by DB2 for z/OS. This is how DB2 Connect exploits the Sysplex information to achieve fault tolerance.
3. If all other attempts to connect fail, then DB2 Connect will try the connection to ALIASB using the address contained in the cataloged node directory.

The **db2pd** command with the **sysplex** parameter (**db2pd -sysplex**) can be used for retrieving information about servers associated with a Sysplex environment.

Configuration requirements for Sysplex

Sysplex exploitation will not be used for a given database unless the DCS directory entry for that database contains Sysplex (not case-sensitive) in the 6th positional parameter.

Configuring connections to IBM mainframe database servers

You can manually configure your TCP/IP connection between a DB2 Connect server and a IBM mainframe database using the DB2 command line processor (CLP). For details on configuring connection using `db2dsdriver.cfg`, see the topic about `db2dsdriver` configuration file.

Before you begin

Before you manually configure a TCP/IP connection between DB2 Connect and a IBM mainframe database server, ensure that:

- TCP/IP is functional on the DB2 Connect server and IBM mainframe system.
- You have identified the following parameter values:
 - Hostname (*hostname*) or IP address (*ip_address*)
 - Connection Service name (*svcename*) or Port number/Protocol (*port_number/tcp*)
 - Target database name (*target_dbname*)
 - Local database name (*local_dcsname*)
 - Node name (*node_name*)

Procedure

To manually configure TCP/IP communications between your DB2 Connect server and an IBM mainframe database:

1. Configure TCP/IP on the DB2 Connect server. Refer to “Configuring TCP/IP for DB2 for z/OS” on page 65.
2. Catalog the TCP/IP node. Refer to the “CATALOG TCPIP/TCPIP4/TCPIP6 NODE command” topic in the *Command Reference*.
3. Catalog the IBM mainframe database as a Database Connection Service (DCS) database. Refer to the “CATALOG DCS DATABASE command” topic in the *Command Reference*.
4. Catalog the IBM mainframe database. Refer to the “CATALOG DATABASE command” topic in the *Command Reference*.
5. Bind utilities and applications to the IBM mainframe database server. Refer to “Binding database utilities on DB2 Connect” on page 81.
6. Test the IBM mainframe connection. Refer to the “CONNECT (Type 1) statement” topic in the *SQL Reference Volume 2*.

Results

Note: Due to the characteristics of the TCP/IP protocol, TCP/IP might not be immediately notified of a partner's failure on another IBM mainframe. As a result, a client application accessing a remote DB2 server using TCP/IP, or the corresponding agent at the server, might sometimes appear to be hung. The TCP/IP `SO_KEEPALIVE` socket option is used to detect when there has been a failure and the TCP/IP connection has been broken.

Registering a DB2 Connect license key using the `db2licm` command

Use the `db2licm` command to apply the license entitlement certificate (also referred to as registering a license key).

Before you begin

To complete this task, you must have the appropriate license file (*.lic).

To connect to a z/OS server or a System i server, you must register a DB2 Connect license key. (Retrieve the license file from your Passport Advantage distribution, for example db2conpe.lic, then copy the license file to the license directory under the directory where the driver was installed.)

If you are using DB2 Connect Unlimited Edition for z/OS, then use a server based license key. This one step will prevent the need for client based license keys. For details, see the topic about activating the license key for DB2 Connect Unlimited Edition for System z.

On Windows operating systems, you must belong to the local Administrators or Power Users group to use the **db2licm** command with the **-a** command parameter.

Procedure

- On Windows operating systems, register a DB2 license key by entering the following command:

```
db2install_path\bin\db2licm -a filename
```

where *db2install_path* is the DB2 installation path *filename* is the full path name and file name for the license file that corresponds to the product or feature you have purchased.

- On Linux or UNIX operating systems, register a DB2 license key by entering the following command:

```
INSTHOME/sqllib/adm/db2licm -a filename
```

where *INSTHOME* represents the home directory of the instance owner and *filename* is the full path name and file name for the license file that corresponds to the product or feature you have purchased. The **db2licm** command can also be found in the path where the DB2 database product is installed. For example, /opt/IBM/db2/V10.5/adm on AIX, HP-UX or Solaris operating systems or /opt/ibm/db2/V10.5/adm on Linux operating systems, if you use the default installation directory.

Chapter 5. Administering

Binding applications and utilities (DB2 Connect server)

Application programs developed using embedded SQL must be bound to each database with which they will operate. For information about the binding requirements for the IBM data server package, see the topic about DB2 CLI bind files and package names.

Binding should be performed once per application, for each database. During the bind process, database access plans are stored for each SQL statement that will be executed. These access plans are supplied by application developers and are contained in *bind files* which are created during precompilation. Binding is a process of processing these bind files by an IBM mainframe database server.

Because several of the utilities supplied with DB2 Connect are developed using embedded SQL, they must be bound to an IBM mainframe database server before they can be used with that system. If you do not use the DB2 Connect utilities and interfaces, you do not have to bind them to each of your IBM mainframe database servers. The lists of bind files required by these utilities are contained in the following files:

- ddcsmvslst for System z
- ddcsvse.lst for VSE
- ddcsvm.lst for VM
- ddcs400.lst for IBM Power Systems

Binding one of these lists of files to a database will bind each individual utility to that database.

If a DB2 Connect server product is installed, the DB2 Connect utilities must be bound to each IBM mainframe database server before they can be used with that system. Assuming the clients are at the same fix pack level, you need to bind the utilities only once, regardless of the number of client platforms involved.

For example, if you have 10 Windows clients, and 10 AIX clients connecting to DB2 for z/OS via DB2 Connect Enterprise Edition on a Windows server, perform one of the following steps:

- Bind ddcsmvslst from one of the Windows clients.
- Bind ddcsmvslst from one of the AIX clients.
- Bind ddcsmvslst from the DB2 Connect server.

This example assumes that:

- All the clients are at the same service level. If they are not then, in addition, you might need to bind from each client of a particular service level
- The server is at the same service level as the clients. If it is not, then you need to bind from the server as well.

In addition to DB2 Connect utilities, any other applications that use embedded SQL must also be bound to each database that you want them to work with. An

application that is not bound will usually produce an SQL0805N error message when executed. You might want to create an additional bind list file for all of your applications that need to be bound.

For each IBM mainframe database server that you are binding to, perform the following steps:

1. Make sure that you have sufficient authority for your IBM mainframe database server management system:

System z

The authorizations required are:

- SYSADM or
- SYSCTRL or
- BINDADD and CREATE IN COLLECTION NULLID

Note: The BINDADD and the CREATE IN COLLECTION NULLID privileges provide sufficient authority **only** when the packages do not already exist. For example, if you are creating them for the first time.

If the packages already exist, and you are binding them again, then the authority required to complete the task(s) depends on who did the original bind.

A) If you did the original bind and you are doing the bind again, then having any of the previously listed authorities will allow you to complete the bind.

B) If your original bind was done by someone else and you are doing the second bind, then you will require either the SYSADM or the SYSCTRL authorities to complete the bind. Having just the BINDADD and the CREATE IN COLLECTION NULLID authorities will not allow you to complete the bind. It is still possible to create a package if you do not have either SYSADM or SYSCTRL privileges. In this situation you would need the BIND privilege on each of the existing packages that you intend to replace.

VSE or VM

The authorization required is DBA authority. If you want to use the GRANT option on the bind command (to avoid granting access to each DB2 Connect package individually), the NULLID user ID must have the authority to grant authority to other users on the following tables:

- system.syscatalog
- system.syscolumns
- system.sysindexes
- system.systabauth
- system.syskeycols
- system.syssynonyms
- system.syskeys
- system.syscolauth
- system.sysuserauth

On the VSE or VM system, you can issue:

grant select on *table* to nullid with grant option

IBM Power Systems

*CHANGE authority or higher on the NULLID collection.

2. Issue commands similar to the following commands:

```
db2 connect to DBALIAS user USERID using PASSWORD
db2 bind path@ddcsmvs.lst blocking all
      sqlerror continue messages ddcsmvs.msg grant public
db2 connect reset
```

Where *DBALIAS*, *USERID*, and *PASSWORD* apply to the IBM mainframe database server, *ddcsmvs.lst* is the bind list file for z/OS, and *path* represents the location of the bind list file.

For example *drive:\sqllib\bnd* applies to all Windows operating systems, and *INSTHOME/sqllib/bnd/* applies to all Linux and UNIX operating systems, where *drive* represents the logical drive where DB2 Connect was installed and *INSTHOME* represents the home directory of the DB2 Connect instance.

You can use the grant option of the **bind** command to grant EXECUTE privilege to PUBLIC or to a specified user name or group ID. If you do not use the grant option of the **bind** command, you must GRANT EXECUTE (RUN) individually.

To find out the package names for the bind files, enter the following command:

```
ddcspkgn @bindfile.lst
```

For example:

```
ddcspkgn @ddcsmvs.lst
```

might yield the following output:

Bind File	Package Name
f:\sqllib\bnd\db2ajgrt.bnd	SQLAB6D3

To determine these values for DB2 Connect execute the **ddcspkgn** utility, for example:

```
ddcspkgn @ddcsmvs.lst
```

Optionally, this utility can be used to determine the package name of individual bind files, for example:

```
ddcspkgn bindfile.bnd
```

Note:

- a. Using the bind option **sqlerror continue** is required; however, this option is automatically specified for you when you bind applications using the DB2 tools or the Command Line Processor (CLP). Specifying this option turns bind errors into warnings, so that binding a file containing errors can still result in the creation of a package. In turn, this allows one bind file to be used against multiple servers even when a particular server implementation might flag the SQL syntax of another to be invalid. For this reason, binding any of the list files *ddcsxxx.lst* against any particular IBM mainframe database server should be expected to produce some warnings.
 - b. If you are connecting to a DB2 database through DB2 Connect, use the bind list *db2ubind.lst* and do not specify **sqlerror continue**, which is only valid when connecting to a IBM mainframe database server. Also, to connect to a DB2 database, it is recommended that you use the DB2 clients provided with DB2 and not DB2 Connect.
3. Use similar statements to bind each application or list of applications.
 4. If you have remote clients from a previous release of DB2, you might need to bind the utilities on these clients to DB2 Connect.

Moving data with DB2 Connect

If you are working in a complex environment in which you need to move data between a host database system and a workstation, you can use DB2 Connect, the gateway for data transfer between the host and the workstation.

About this task

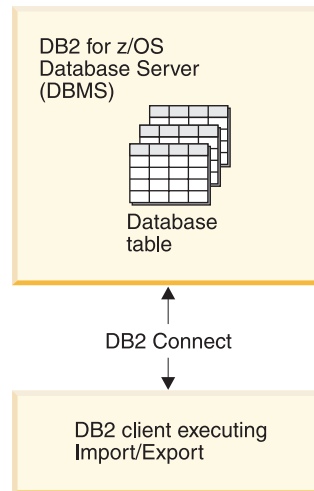


Figure 4. Import/Export through DB2 Connect

The DB2 database export and import utilities allow you to move data from an IBM mainframe server database to a file on the DB2 Connect workstation, and the reverse. You can then use the data with any other application or relational database management system that supports this export or import format. For example, you can export data from an IBM mainframe server database into a PC/IXF file, and then import it into a DB2 for Linux, UNIX, and Windows database.

You can perform export and import operations from a database client or from the DB2 Connect workstation.

Note:

1. The data to be exported or imported must comply with the size and data type restrictions that are applicable to both databases.
2. To improve import performance, you can use compound queries. Specify the compound file type modifier in the import utility to group a specified number of query statements into a block. This can reduce network usage and improve response time.

With DB2 Connect, export and import operations must meet the following conditions:

- The file type must be PC/IXF.
- A target table with attributes that are compatible with the data must be created on the target server before you can import to it. The **db2look** utility can be used to get the attributes of the source table. Import through DB2 Connect cannot create a table, because INSERT is the only supported option.

If any of these conditions is not met, the operation fails, and an error message is returned.

Note: Index definitions are not stored on export or used on import.

If you export or import mixed data (columns containing both single-byte and double-byte data), consider the following considerations :

- On systems that store data in EBCDIC (MVS™, System z, IBM Power Systems, VM, and VSE), shift-out and shift-in characters mark the start and the end of double-byte data. When you define column lengths for your database tables, be sure to allow enough room for these characters.
- Variable-length character columns are recommended, unless the column data has a consistent pattern.

Procedure

- To move data from a workstation to host or System i server database:
 1. Export the data from a DB2 table to a PC/IXF file.
 2. Using the INSERT option, import the PC/IXF file into a compatible table in the host server database.
- To move data from a host server database to a workstation:
 1. Export the data from the host server database table to a PC/IXF file.
 2. Import the PC/IXF file into a DB2 table.

Example

The following example illustrates how to move data from a workstation to a host or System i server database.

Export the data into an external IXF format by issuing the following command:

```
db2 export to staff.ixf of ixf select * from userid.staff
```

Issue the following command to establish a DRDA connection to the target DB2 database:

```
db2 connect to cbc664 user admin using xxx
```

If it doesn't already exist, create the target table on the target DB2 database instance:

```
CREATE TABLE mydb.staff (ID SMALLINT NOT NULL, NAME VARCHAR(9),  
DEPT SMALLINT, JOB CHAR(5), YEARS SMALLINT, SALARY DECIMAL(7,2),  
COMM DECIMAL(7,2))
```

To import the data issue the following command:

```
db2 import from staff.ixf of ixf insert into mydb.staff
```

Each row of data will be read from the file in IXF format, and an SQL INSERT statement will be issued to insert the row into table mydb.staff. Single rows will continue to be inserted until all of the data has been moved to the target table.

What to do next

Detailed information is available in "Moving Data Across the DB2 Family," an IBM Redbooks® publication. This Redbooks publication can be found at the following website: www.redbooks.ibm.com/redbooks/SG246905.

Automatic client reroute description and setup (DB2 Connect server)

The main goal of the automatic client reroute feature is to enable an IBM Data Server Client application to recover from a loss of communications so that the application can continue its work with minimal interruption. As the name suggests, rerouting is central to the support of continuous operations. But rerouting is only possible when there is an alternate location that is identified to the client connection. Rerouting is not required if using the IBM data server client as a DB2 Connect client. For details, see the topic about IBM data server client types.

Automatic client reroute with IBM Data Server feature redirects client applications from a failed server to an alternate server so the applications can continue their work with minimal interruption. Seamless automatic client reroute for DB2 for z/OS Sysplex is on by default and is recommended when WLB is enabled. With this support, applications that access DB2 for z/OS Sysplex should use seamless automatic client reroute capabilities provided by the client, and are not required to go through a DB2 Connect server. For more information about this feature, see the topic about automatic client reroute (client-side) in the DB2 Information Center.

Outside of a DB2 Connect high-availability environment, the database being accessed is typically synchronized between the original DB2 server and the alternate DB2 server through one of various means, such as High availability disaster recovery (HADR) or IBM PowerHA® SystemMirror for AIX.

However, in the case of the DB2 Connect server, because there is no requirement for the synchronization of local databases, you only need to ensure that both the original and alternate DB2 Connect servers have the target IBM mainframe database cataloged in such a way that it is accessible using an identical database alias.

Note: In a DB2 Connect server environment, an alternate DB2 Connect server can be specified to enable automatic rerouting between a client and the DB2 Connect server. For rerouting to occur between the DB2 Connect clients or server products and a IBM mainframe database server, the remote server must provide one or more alternate addresses for itself. In the case of DB2 for z/OS, multiple addresses are known if the database is a Sysplex data sharing environment.

Rerouting capability for Sysplex can be configured between DB2 Connect and the host database server if Sysplex support is enabled. The rerouting capability for Sysplex is a DB2 Connect feature that allows DB2 Connect to try the connection against other members of the Sysplex group following the loss of communication with the original member. The alternate server does not need to be cataloged in the database directory to enable the reroute capability for Sysplex on DB2 Connect. By default, the reroute capability for Sysplex is enabled if Sysplex support is enabled.

In order for an IBM Data Server Client to have the ability to recover from a loss of communications to a DB2 Connect server using automatic client reroute, an alternate DB2 Connect server location must be specified before the loss of communication occurs. The **UPDATE ALTERNATE SERVER FOR DATABASE** command is used to define the alternate DB2 Connect server location for a particular IBM mainframe database. The alternate hostname and port number is given as part of the command. The location is stored in the system database directory file at the DB2 Connect server. In order to ensure the alternate DB2 Connect server location specified applies to that database for all clients, the alternate server location has to be specified at the DB2 Connect server side. The alternate server is ignored if it is set at the client instance.

For example, suppose a IBM mainframe database is catalogued using a database alias of db1 at a DB2 Connect server S1 (with a hostname of db2conn1 and a port number of 122). The database administrator would like to specify an alternate DB2 Connect server S2 at hostname db2conn2 with a port number of 123. Here is the command the database administrator would run at the DB2 Connect server S1:

```
db2 update alternate server for database db1 using hostname db2conn2 port 123
```

After you have specified the alternate DB2 Connect server location for database alias db1 at DB2 Connect server S1, the alternate server location information is returned to the IBM Data Server Client as part of the connection process. If communication between the IBM Data Server Client and the DB2 Connect server S1 is lost for any reason (typically a communication error, such as SQL code -30081 or SQL code -1224), the IBM Data Server Client will attempt to reconnect to db1 through either the original DB2 Connect server (S1) or the alternate DB2 Connect server (S2), alternating the attempts between the two servers. The time interval between attempts starts off rapidly, then gradually lengthens with each attempt.

Once a connection is successful, the SQL code -30108 is returned to indicate that a database connection has been reestablished following the communication failure. The hostname or IP address and service name or port number are returned. The IBM Data Server Client only returns the error for the original communications failure to the application if the reestablishment of the client communications is not possible to either the original or alternative server.

The following considerations involving alternate server connectivity in a DB2 Connect server environment should also be noted:

- When using a DB2 Connect server for providing access to an IBM mainframe database on behalf of both remote and local clients, confusion can arise regarding alternate server connectivity information in a system database directory entry. To minimize this confusion, consider cataloging two entries in the system database directory to represent the same IBM mainframe database. Catalog one entry for remote clients and catalog another for local clients.
- Any SYSPLEX information that is returned from a target DB2 for z/OS server is kept only in cache at the DB2 Connect server. Only one alternate server is written to disk. When multiple alternates or multiple active servers exist, the information is only maintained in memory and is lost when the process terminates.

Administering DB2 Connect systems

Overview

Access DB2 data from remote clients

The IBM data server client provides a runtime environment that enables client applications to access one or more remote databases. With the IBM data server client, you can remotely administer DB2 or DB2 Connect servers.

All applications must access a database through the IBM data server client. A Java applet can access a remote database through a Java-enabled browser.

DB2 Connect client using the IBM data client is supported on Linux, UNIX, and Windows operating systems.

Accessing IBM mainframe DB2 data using DB2 Connect

A DB2 Connect client or Server enables a IBM data server client on a LAN access to data that is stored on IBM mainframe systems.

In organizations with large amounts of data, IBM DB2 for IBM i, DB2 for z/OS, or DB2 Server for VM and VSE are commonly used to manage that data. Applications that run on any of the supported platforms can work with this data transparently, as if a local database server managed it. A DB2 Connect client or Server is required for supporting applications which access IBM mainframe data and exploit transaction monitors as well as applications that are implemented as Java applets.

In addition, you can use a wide range of off-the-shelf or custom-developed database applications with DB2 Connect and its associated tools. For example, you can use DB2 Connect products with:

- *Spreadsheets*, such as Microsoft Excel and Lotus 1-2-3®, to analyze real-time data without having the cost and complexity of data extract and import procedures.
- *Decision support tools*, such as BusinessObjects, Brio and Impromptu®, and Crystal Reports, to provide real-time information.
- *Database products*, such as Lotus Approach® and Microsoft Access.
- *Development tools*, such as PowerSoft PowerBuilder, Microsoft Visual Basic, and Borland Delphi, to create client/server solutions.

A DB2 Connect server product, such as DB2 Connect Enterprise Edition, is most appropriate for the following environments:

- *Federation*.
- *Transaction monitors*, such as BEA Tuxedo and BEA Weblogic. (See Figure 5 on page 81.)

DB2 Connect provides transparent access to IBM mainframe data through a standard architecture for managing distributed data. This standard is known as Distributed Relational Database Architecture (DRDA). DRDA allows your applications to establish a fast connection to IBM mainframe databases without expensive IBM mainframe components or proprietary gateways.

Although DB2 Connect is often installed on an intermediate server machine, it is recommended to connect an IBM data server client to an IBM mainframe database directly by installing the appropriate DB2 Client such as one of the IBM data server client or driver. For more information about the DB2 Connect client, see the topic about IBM data server client types.

DB2 Connect can also be installed on a Web server, Transaction Processor (TP) monitor, or other 3-tier application server machines with multiple local SQL application processes and threads. In these cases, you can choose to install DB2 Connect on the same machine for simplicity, or on a separate machine to off-load CPU cycles.

A DB2 Connect server enables multiple clients to connect to IBM mainframe data and can significantly reduce the effort that is required to establish and maintain access to enterprise data.

To connect to an IBM mainframe database server you require a licensed DB2 Connect product. You cannot connect directly to an IBM mainframe Data Server using a IBM data server client.

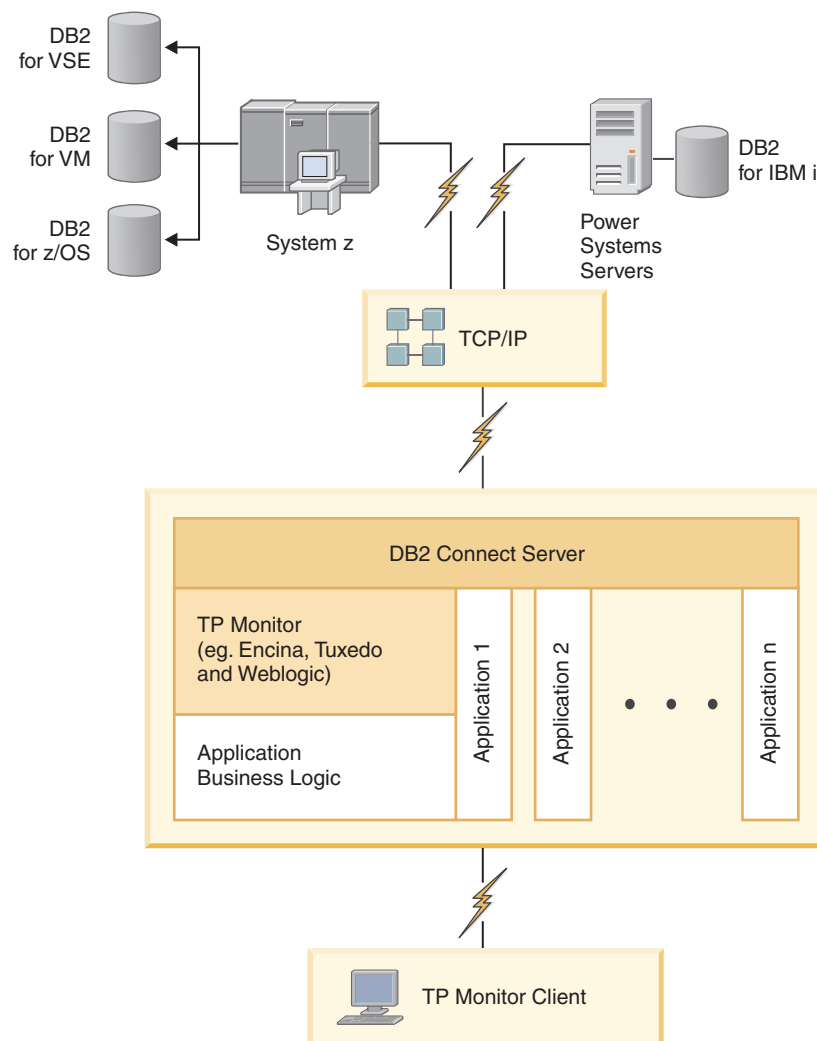


Figure 5. Transaction monitors working with DB2 Connect.

Binding database utilities on DB2 Connect

You must bind the database utilities (import, export, reorg, the Command Line Processor) and CLI bind files to each database before they can be used with that database.

About this task

In a network environment, if you are using multiple clients that use different versions or service levels of DB2, you must bind the utilities once for each version of DB2 used.

Binding a utility creates a *package*, which is an object that includes all of the information that is needed to process specific SQL statements from a single source file.

The bind files are grouped together in different .lst files in the bnd directory, under the installation directory (typically sql1ib for Windows). Each file is specific to a server.

Procedure

- To bind the utilities and applications to the IBM mainframe database server, connect to the IBM mainframe server and use the following example as a template:

```
connect to dbalias user userid using password
bind path/bnd/@ddcsmvs.lst blocking all sqlerror continue
      messages mvs.msg grant public
connect reset
```

where *path* corresponds to the **DB2PATH** registry value.

- To bind database utilities to a DB2 database, use the command line processor:
 1. Change to the bnd directory, which is *x*:\sql11b\bnd, where *x*: represents the drive where you installed DB2.
 2. To connect to the database, enter the following commands in the Command Center® or the Command Line Processor:

```
connect to database_alias
```

where *database_alias* represents the alias of the database to which you want to connect.

3. Enter the following commands in the Command Line Processor:

```
"bind @db2ubind.lst messages bind.msg grant public"
"bind @db2cli.lst messages clibind.msg grant public"
```

In this example, bind.msg and clibind.msg are the output message files, and EXECUTE and BINDADD privileges are granted to public.

4. Reset the connection to the database by entering the following command:

```
connect reset
```

Note:

1. The db2ubind.lst file contains the list of bind (.bnd) files required to create the packages for the database utilities. The db2cli.lst file contains the list of bind (.bnd) files required to create packages for the CLI and the DB2 ODBC driver.
2. Binding might take a few minutes to complete.
3. If you have BINDADD authority, the first time you use the CLI or ODBC driver, the CLI packages will be bound automatically. If the applications that you are using require binding to the database, you can use the **BIND** command to perform the bind action.

Considerations for System z SYSPLEX exploitation

DB2 Connect provides load balancing and fault-tolerance when routing connections to DB2 Sysplex. When connected to a DB2 for z/OS database server running in a DB2 pureScale environment, DB2 Connect will spread the workload amongst the different DB2 subsystems comprising the data sharing group, based on the system load and health information provided by the Workload Manager (WLM). It uses the Distributor to route connections. Use the group IP address to connect to a group location.

DB2 Connect receives a prioritized list of DB2 members from the WLM. Each Sysplex returns weighted priority information for each connection address that has capacity to run the work. This list is then used by DB2 Connect to handle the incoming CONNECT requests by distributing them among the DB2 members with the best capacity to run work. For load balancing, the list of Sysplex weighted priority information is obtained during each connection. This list is also used when determining where to send each transaction.

Note: System z Distributed Data Facility (DDF) configuration does not need to be changed to take advantage of the DB2 Connect Sysplex exploitation. Refer to DB2 for z/OS Data Sharing Planning and Administration guide.

DB2 Connect also provides fault-tolerance by attempting to connect to an alternate sysplex machine in the event of a connection failure. An error will only be returned to the application if all known connections have failed.

DB2 Connect is designed with a transport tool. With Sysplex enabled, DB2 Connect routes connections using a transport member and associates it with a logical connection.

Conversion of character data

When character data is transferred between machines, it must be converted to a form that the receiving machine can use.

For example, when data is transferred between a DB2 Connect server and a host or System i database server, it is usually converted from a server code page to a host CCSID, and vice versa. If the two machines use different code pages or CCSIDs, code points are mapped from one code page or CCSID to the other. This conversion is always performed at the receiver.

Character data sent *to* a database consists of SQL statements and input data. Character data sent *from* a database consists of output data. Output data that is interpreted as bit data is not converted. For example, data from a column declared with the FOR BIT DATA clause. Otherwise, all input and output character data is converted if the two machines have different code pages or CCSIDs.

For example, if DB2 Connect is used to access data, the following happens:

1. DB2 Connect sends an SQL statement and input data to System z.
2. DB2 for z/OS converts the SQL statement and data to the host server's code page and then processes the data.
3. DB2 for z/OS sends the result back to the DB2 Connect server.
4. DB2 Connect converts the result to the code page of the user's environment.

For bidirectional languages, a number of special "BiDi CCSIDS" have been defined by IBM and are supported by DB2 Connect.

If the bidirectional attributes of the database server are different from those of the client you can use these special CCSIDS to manage the difference.

Refer to the supported territory codes and code pages topic for the supported conversions between code pages on the DB2 Connect and CCSIDs on the host or System i server.

System i and mainframe support for DB2 Connect

Before you access DB2 data on System z or System i data servers by using DB2 Connect products, ensure that the data server meets requirements.

DB2 Connect supports connectivity to the following mainframe and System i servers:

Table 12. Supported mainframe and IBM i data servers

Version	Recommended maintenance levels
DB2 for z/OS	<p>See website for IBM z/OS Consolidated Service Test and the RSU (. http://www.ibm.com/servers/eserver/zseries/zos/servicetst/)).</p> <p>In general, install the most recent Recommended Service Upgrade (RSU) to avoid encountering problems that are caused by software defects that IBM has corrected.</p> <p>Important: For Version 10.1, the fix for DB2 for z/OS V10 APAR PM79161 must be applied before you connect to the DB2 for z/OS database. Without this APAR fix, DB2 for z/OS abends when DB2 Connect attempts to connect to the DB2 for z/OS database.</p> <p>DB2 Connect Version 10.5 supports the following versions of DB2 for z/OS servers:</p> <ul style="list-style-type: none"> • DB2 for z/OS Version 9.1 • DB2 for z/OS Version 10 • DB2 for z/OS Version 11
DB2 for i (formerly known as DB2 Universal Database for i5/OS) V5R4	<p>II13348 (Informational APAR)</p> <p>PTFs: MF53402 and MF53403</p> <p>See website for System i Preventative Service Planning (. http://www.ibm.com/servers/eserver/zseries/zos/servicetst/).</p>
DB2 for i V6R1	<p>PTFs: SI30564, SI30588, SI30611, SI30620, SI30621, SI30622, SI30825, SI30827, SI30920, SI30921, SI31019, SI31101, SI31125, SI31238, and SI31480.</p> <p>See website for System i Preventative Service Planning (. http://www-912.ibm.com/s_dir/sline003.NSF/GroupPTFs?OpenView&view=GroupPTFs)</p>
DB2 for i V7R1	<p>PTFs: SI43890, SI43864, SI43863, SI43817, SI43807, SI43806, SI43805, SI43804, SI43803, SI43802, SI43801, SI43768, SI43757, SI43721, SI43658, SI43651, SI43577, SI43550, SI43544, SI43539, SI43532, SI43476, SI43466, SI43446, SI43386, SI43373, SI43111, SI43017, SI43016, SI42986, SI42954, SI42947, SI42928, SI42927, SI42906, SI42872, SI42783, SI42775, SI42769, SI42768, SI42745, SI42716, SI42700, SI42504, and SI42492.</p> <p>See website for System i Preventative Service Planning (. http://www-912.ibm.com/s_dir/sline003.NSF/GroupPTFs?OpenView&view=GroupPTFs).</p> <p>Important: Use DB2 Connect V9.7 Fix Pack 4 or later to connect to DB2 for i V7R1.</p>
DB2 Server for VM and VSE Version 7 and later	<p>See website for DB2 Server for VSE & VM (http://www.ibm.com/software/data/db2/vse-vm/).</p>

Understanding the Administration Server

The DB2 Administration Server (DAS) responds to requests from the DB2 Administration Tools.

The DB2 Administration Tools, for example, allow you to start, stop, and set database manager configuration parameters for servers. The Administration Server helps users to catalog databases on a client. The DAS is available on all supported Linux, Windows, and UNIX operating systems as well as the System z (z/OS only) operating systems.

An Administration Server must reside on each server that you want to administer and detect. The Administration Server is automatically created and started for you. The setup program creates the Administration Server on the instance-owning machine and automatically starts it at boot time. By default the DAS instance is DB2AS, which is the default user ID that is created using the DB2 Setup wizard.

Important: The DB2 Administration Server (DAS) has been deprecated in Version 9.7 and might be removed in a future release. The DAS is not supported in DB2 pureScale environments. Use software programs that use the Secure Shell protocol for remote administration. For more information, see “ DB2 administration server (DAS) has been deprecated” at .

Distributed Relational Database Architecture

Distributed Relational Database Architecture (DRDA) is a set of protocols that permits multiple database systems, which includes both IBM and not IBM systems, as well as application programs, to work together.

Any combination of relational database management products that use DRDA can be connected to form a distributed relational database management system. DRDA coordinates communication between systems by defining what must be exchanged and how it must be exchanged.

Unit of work

A *unit of work (UOW)* is a single logical transaction. It consists of a sequence of SQL statements in which either all of the operations are successfully performed or the sequence as a whole is considered unsuccessful.

Distributed unit of work

A *distributed unit of work (DUOW)*, also known as multisite update, involves more than one database server within a unit of work. A DUOW has the following characteristics:

- More than one database management server is updated per unit of work.
- The application directs the distribution of work, and initiates commit.
- There might be multiple requests per unit of work.
- There is one database management server per request.
- Commitment is coordinated across multiple database servers.

DRDA and data access

Although DRDA defines database communication protocols, it does not define the programming interfaces, or APIs, that should be used by application programmers.

In general, DRDA can be used by an application program to pass any request that a target DRDA server can execute. All of the DRDA servers available today can execute SQL requests forwarded by an application program through DB2 Connect.

IBM provides application programmers with tools to generate SQL requests for the Windows, UNIX, and Linux operating systems. These tools are part of the DB2 client. The DB2 database manager supports several programming interfaces: ADO.NET, JDBC, SQLJ, PHP, Perl DBI, embedded SQL, DB2 Call Level Interface (DB2 Call Level Interface), and OLE DB. These APIs can be used by programmers to build applications in a variety of programming languages.

DB2 Connect and DRDA

DB2 Connect implements the DRDA architecture to reduce the cost and complexity of accessing data stored in IBM DB2 for IBM i, DB2 for IBM Power Systems, DB2 for z/OS, DB2 Server for VM and VSE, and other DRDA-compliant database servers. By fully exploiting the DRDA architecture, DB2 Connect offers a well-performing, low-cost solution with the system management characteristics that customers demand.

In DRDA terminology, an *application requester (AR)* is the code that handles the application end of a distributed connection. The AR is the application that is requesting data. DB2 Connect acts as an application requester on behalf of application programs which can be local to the DB2 Connect workstation or on a separate client remote to DB2 Connect.

An *application server (AS)* is the code that handles the database end of the connection.

DRDA also supports multi-tier connections between an application requester and a server. In this topology, the server that an application requester connects to is an application server, but any other server further downstream is called a database server (DS) as it does not interact directly with the application requester. In addition, to highlight its role as neither the system where a database request originates nor the system that performs the database function for the request, each application server or database server between an application requester and the final database server is also called an intermediate server. The use of database servers and intermediate servers is supported by DB2 Connect.

Figure 6 shows the flow of data between the DB2 Connect workstation and the IBM mainframe server in the case where there are local clients only.

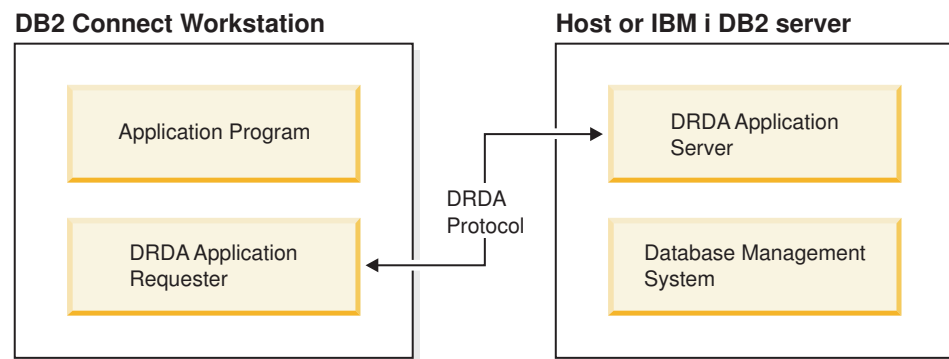


Figure 6. Data flow between a DB2 Connect server and an IBM mainframe server

To implement the connections between DRDA server database management systems and IBM data server clients, DRDA uses the following architectures:

- Character Data Representation Architecture (CDRA)
- Distributed Data Management Architecture (DDM)
- Formatted Data Object Content Architecture (FD:OCA)
- Transmission Control Protocol/Internet Protocol (TCP/IP).

These architectures are used as building blocks. The data streams which flow over the network are specified by the DRDA architecture which documents a data stream protocol supporting distributed relational database access.

A request is routed to the correct destination by means of directories that contain various types of communication information and the name of the DRDA server database being accessed.

Remote unit of work

A *remote unit of work* lets a user or application program read or update data at one location per unit of work. It supports access to one database within a unit of work.

While an application program can update several remote databases, it can only access one database within a unit of work.

Remote unit of work has the following characteristics:

- Multiple requests (SQL statements) per unit of work are supported.
- Multiple cursors per unit of work are supported.
- Each unit of work can update only one database.
- The application program either commits or rolls back the unit of work. In certain error circumstances, the database server or DB2 Connect might roll back the unit of work.

For example, Figure 7 shows a database client running a funds transfer application that accesses a database containing checking and savings account tables, as well as a transaction fee schedule. The application must:

- Accept the amount to transfer from the user interface.
- Subtract the amount from the savings account, and determine the new balance.
- Read the fee schedule to determine the transaction fee for a savings account with the given balance.
- Subtract the transaction fee from the savings account.
- Add the amount of the transfer to the checking account.
- Commit the transaction (unit of work).

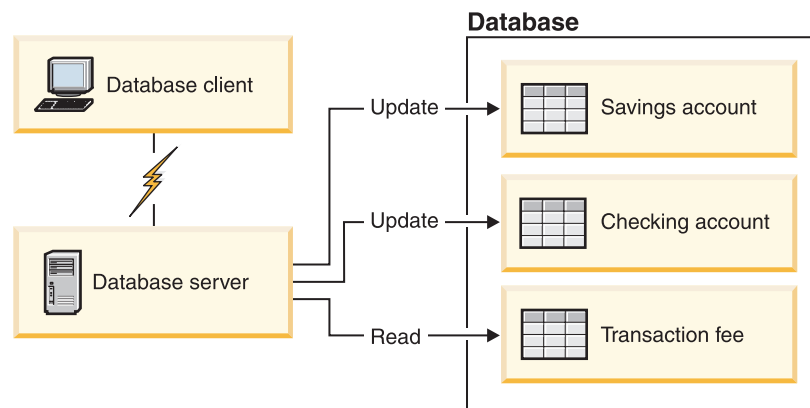


Figure 7. Using a Single Database in a Transaction

To set up such an application, you must:

1. Create the tables for the savings account, checking account and transaction fee schedule in the same database.
2. If physically remote, set up the database server to use the appropriate communications protocol.
3. If physically remote, catalog the node and the database to identify the database on the database server.
4. Precompile your application program to specify a type 1 connection; that is, specify **CONNECT(1)** on the **PREP** command.

Distributed requests

A *distributed request* is a distributed database function that allows applications and users to submit SQL statements that reference two or more DBMSs or databases in

a single statement. For example, a join between tables in two different DB2 for z/OS subsystems. DB2 Connect provides support for distributed requests across databases and DBMSs.

For example, you can perform a UNION operation between a DB2 table and an Oracle view. Supported DBMSs include members of the DB2 Family (such as DB2 for Linux, UNIX, and Windows, DB2 for z/OS, and DB2 for i) and Oracle. Multivendor support is available when using DB2 Connect in conjunction with InfoSphere Federation Server.

Distributed request provides *location transparency* for database objects. If information (in tables and views) is moved, references to that information (called *nicknames*) can be updated without any changes to applications that request the information. Distributed request also provides *compensation* for DBMSs that do not support all of the DB2 SQL dialect, or certain optimization capabilities. Operations that cannot be performed under such a DBMS (such as recursive SQL) are run under DB2 Connect.

Distributed request function in a *semi-autonomous* manner. For example, DB2 queries containing references to Oracle objects can be submitted while Oracle applications are accessing the same server. Distributed request does not monopolize or restrict access (beyond integrity and locking constraints) to Oracle or other DBMS objects.

Implementation of the distributed request function consists of a DB2 Connect instance, a database that will serve as the federated database, and one or more remote data sources. The *federated database* contains catalog entries identifying data sources and their characteristics. A *data source* consists of a DBMS and data. Applications connect to the federated database just like any other DB2 database. DB2 Connect federated database is not licensed for managing user data. Its sole purpose is to contain information about data sources.

After a federated system is set up, the information in data sources can be accessed as though it were in one large database. Users and applications send queries to one federated database, which then retrieves data from DB2 Family and Oracle systems as needed. User and applications specify nicknames in queries; these nicknames provide references to tables and views located in data sources. From an end-user perspective, nicknames are similar to aliases.

Many factors can affect the performance of distributed requests. The most critical factor is to ensure that accurate and up-to-date information about data sources and their objects is stored in the federated database global catalog. This information is used by the DB2 optimizer, and can affect decisions to push down operations for evaluation at data sources.

Updating database directories

You can define where database connection information is stored by updating the database directory.

Before you begin

DB2 Connect uses the system database, node and database connection services (DCS) directory to manage database connection information. Before updating these directories, you should configure communications on the IBM mainframe database server and workstations.

About this task

DB2 Connect uses the following directories to manage database connection information:

- *system database directory*, which contains name, node, and authentication information for every database that DB2 Connect accesses.
- *node directory*, which contains network address and communication protocol information for every IBM mainframe database server that DB2 Connect accesses.
- *database connection services (DCS) directory*, which contains information specific to IBM mainframe database server databases.

Database directories can be updated by cataloging your databases, nodes, or DCS directory.

Procedure

To update database directories:

1. Collect database directory information using the directory customization worksheet. Refer to “Directory customization worksheet” on page 95.
2. Update the directories with information about remote database server machines by cataloging your databases, nodes, or DCS directory. See “Configuring connections to IBM mainframe database servers” on page 71 for details on how to catalog databases, nodes, or DCS directory.

System database directory values

A system database directory exists for each instance of the database manager, and contains one entry for each database that has been cataloged for this instance. In DB2 Connect products, the system database directory contains information about the name, alias, node name, and authentication type of each database.

You can specify the following information in the system database directory:

Database name

The same value that you wrote in the DCS Directory Parameters table.

Database alias

An alias for the IBM mainframe database server. This name will be used by any application program that accesses the database. By default, the value that you specify for Database name is used.

Format: 1-8 single-byte alphanumeric characters, including the number sign (#), at sign (@), dollar sign (\$), and underscore (_). It cannot begin with an underscore or a number.

Node name

The same value that you wrote in the Node Directory Parameters table.

Authentication

Specifies where the validation of the user's name and password will be made for connections originating from the DB2 Connect server. The valid options are: SERVER, SERVER_ENCRYPT, CLIENT, KERBEROS, SERVER_ENCRYPT_AES, and DATA_ENCRYPT. There is no support for the GSSPLUGIN authentication type in the system database directory.

Node directory values

You can specify the following information in the node directory: node name; protocol; security type; TCP/IP host name or IP address; TCP/IP service name or port number.

Node name

A nickname for the IBM mainframe database server system on which the remote database resides. This name is user-defined. Write the same node name in both the Node Directory Parameters table and the System Database Directory Parameters table.

Format: 1-8 single-byte alphanumeric characters, including the number sign (#), at sign (@), dollar sign (\$), and underscore (_). It cannot begin with an underscore or a number.

Protocol

Must be TCP/IP.

Security type

The type of security checking that will be done. For TCP/IP nodes, SECURITY SOCKS is an option which specifies that the node will be SOCKS-enabled, in which case the **SOCKS_NS** and **SOCKS_SERVER** environment variables are mandatory and must be set to enable SOCKS.

TCP/IP remote hostname or IP address

When defining a TCP/IP node, either the remote TCP/IP hostname, or the remote TCP/IP address. If a hostname is specified, then it must be resolved at the DB2 Connect workstation, either through Domain Name Server (DNS) lookup, or by an entry in the local TCP/IP hosts file.

For DB2 for z/OS remote hosts, the hostname appears in the DSNL004I message (DOMAIN=hostname) when the Distributed Data Facility (DDF) is started. The **-DISPlay DDF** command could also be used.

If accessing a z/OS data sharing group, the domain name should map to the DB2 group dynamic VIPA address. This address routes to the least loaded DB2 member. To access a specific member use the specific DB2 member dynamic VIPA address and turn off sysplex routing. Each member DSNL004I message displays the member specific domain name.

TCP/IP service name or port number

When defining a TCP/IP node, either the remote TCP/IP service name or port number. This must be defined to TCP/IP at the remote host. Port number 446 has been registered as the default port number for DRDA.

For DB2 for z/OS remote hosts, the port number is defined in the Boot Strap Data Set (BSDS) as PORT and is also provided in the DSNL004I message (TCPPORT=portnumber) when the Distributed Data Facility (DDF) is started. The **-DISPlay DDF** command could also be used.

If accessing a z/OS data sharing group, the domain name should map to the DB2 group dynamic VIPA address. This address routes to the least loaded DB2 member. To access a specific member use the specific DB2 member dynamic VIPA address and turn off sysplex routing. Each member DSNL004I message displays the member specific domain name.

Note: A second port used for two-phase commit resynchronization operations over TCP/IP connections can be assigned by the server. For example, the DB2 for z/OS bootstrap data set assigns a port number (RESPORT) to be used for resynchronization for inbound connections to DB2 for z/OS only. No service name need be defined for this.

DCS directory values

You can use the Database Connection Services (DCS) to specify values that are used to define how an application connects to a database and what database it connects to.

You can specify the following information in the DCS directory:

Database name

A user-defined nickname for the IBM mainframe database server. Use the same database name in both the DCS Directory Parameters table and the System Database Directory Parameters table.

Format: 1-8 single-byte alphanumeric characters, including the number sign (#), at sign (@), dollar sign (\$), and underscore (_). It cannot begin with an underscore or a number.

Target database name

The database on the IBM mainframe database server system, as follows:

System z

A DB2 for z/OS subsystem identified by its LOCATION NAME or one of the alias LOCATION names defined on the z/OS server.

The LOCATION NAME can be determined by logging in to TSO and issuing the following SQL query using one of the available query tools:

```
select current server from sysibm.sysdummy1
```

multiple LOCATION NAMES are also defined in the Boot Strap Data Set (BSDS) as well as the DSNL004I message (LOCATION=location), which is written when the Distributed Data Facility (DDF) is started. The **-DISPlay DDF** command could also be used.

If accessing a z/OS data sharing group, the domain name should map to the DB2 group dynamic VIPA address. This address routes to the least loaded DB2 member. To access a specific member use the specific DB2 member dynamic VIPA address and turn off sysplex routing. Each member DSNL004I message displays the member specific domain name.

VSE or VM

The database name (DBNAME)

IBM Power Systems

The relational database name (RDBNAME)

Other For Windows, Linux, and UNIX operating systems, the database alias found in the database directory.

Parameter string

If you want to change the defaults, specify any or all the following parameters in the following order.

map-file

The name of an SQLCODE mapping file that overrides the default SQLCODE mapping. To turn off SQLCODE mapping, specify **NOMAP**.

Note: When processing a query request, the DRDA server returns data in the form of a set of rows that represent the

result set. With each row, there is also an SQLCA returned, usually containing a zero or positive sqlcode (such as +12 or +802). If you use a customized mapping file at a DB2 Connect server, such positive sqlcodes will not be mapped if they are contained in the customized mapping file and have customized mappings (for example, they are mapped to a different sqlcode or have customized token mappings).

It is important to emphasize that:

1. Positive sqlcodes represent warnings, as opposed to negative sqlcodes which indicate error conditions. All the negative sqlcodes will always be mapped in all circumstances, regardless of which mapping file is being used. All the positive sqlcodes, contained in the customized mapping file and mapped to themselves with no change, will always be mapped as well. Also, those positive sqlcodes that are not contained in the customized mapping file at the DB2 Connect server will also always be mapped.
2. If you use the default mapping file, or you connect to the host database directly, the sqlcode mapping will always be performed for all sqlcodes.

,D This is the second positional parameter. If it is specified the application will disconnect from the IBM mainframe database server database when one of the following SQLCODES is returned:

SQL3000N
SQL3004N
SQL3005N
SQL30051N
SQL30053N
SQL3006N
SQL3007N
SQL30071N
SQL30072N
SQL30073N
SQL30074N
SQL30090N

When the disconnect parameter **,D** is not specified, a disconnect will be performed only when the following SQLCODEs are returned:

SQL3002N
SQL30021N
SQL30041N
SQL30061N
SQL30081N

For explanations of these codes, refer to the *Message Reference*.

Note: If DB2 Connect disconnects due to an error, a rollback will be done automatically.

„INTERRUPT_ENABLED

This is the third positional parameter. **INTERRUPT_ENABLED** only applies if the end server does not support interrupts.

If a server supports the DRDA interrupt flow DB2 Connect will simply pass the interrupt request on to the server.

If **INTERRUPT_ENABLED** is configured in the DCS directory at the DB2 Connect workstation, and a client application issues an interrupt while connected to the IBM mainframe database server, DB2 Connect will perform the interrupt by dropping the connection and rolling back the unit of work. This interrupt behavior is supported on AIX, and Windows.

The application will receive sqlcode (-30081) indicating that the connection to the server has been terminated. The application must then establish a new connection with the IBM Mainframe database server, in order to process additional database requests. On platforms other than AIX V5.2 and later and Windows, DB2 Connect does not support the option of automatically disconnecting when an application using it receives an interrupt request.

Note: This support works for TCP/IP connections on any platforms. The client might kill the socket, but - depending on the server implementation - there might or might not be an outstanding receive. DB2 for z/OS uses asynchronous socket calls and therefore is able to detect the loss of the connection and roll back any long-running SQL statements that are in progress.

,,,,,SYSPLEX

This parameter, the 6th positional parameter, can be used to explicitly enable DB2 Connect SYSPLEX support for a particular database. The 6th parameter is disabled by default unless it is explicitly specified.

,,,,,LOCALDATE="value"

This parameter, the seventh positional parameter, is used to enable DB2 Connect date formatting support. This is implemented using a date mask for the *value* as follows:

Suppose you issue the following CLP (command line processor) statements:

```
catalog TCP/IP node nynode remote myhost server myport
catalog dcs database nydb1 as new_york
catalog database nydb1 as newyork1 at node nynode
authentication server
```

The database alias newyork1 is to be used for accessing a host database without date transformation because no date mask has been specified.

However, with the new date formatting support, you can now use the following CLP commands. In this case, because the CLP is being used, and the parameter string is itself being specified using double quotation marks, the **LOCALDATE** value has to be specified inside two pairs of double quotation marks. Note the use of the operating system escape character "\" (backslash) to ensure that the double quotation marks are not stripped from the LOCALDATE specification.

```
catalog dcs database nydb2 as new_york
  parms \",,,,,,LOCALDATE=\"\"YYYYMMDD\"\"\"
catalog database nydb2 as newyork2 at node nynode
  authentication server
```

The database alias newyork2 gives you access to the same host database but, in addition, it has a date format mask specified. This example illustrates that the date format mask is specified using the keyword **LOCALDATE** and is the seventh positional parameter in the **PARMS** field of a DCS directory entry.

For the date mask to be valid, ALL of the following conditions must be true:

1. There can only be at most one sequence each of Y's, M's, and D's where Y is a year digit, M is a month digit, and D is a day digit.
2. The maximum number of Y's in a sequence is 4.
3. The maximum number of M's in a sequence is 2.
4. The maximum number of D's in a sequence is 2.

For instance, the following date format masks are all valid date masks:

```
"YYyyMmDd" - Y, M, and D digits are case-insensitive
"MM+DD+YYYY" - OK to have a mask longer than 10 bytes
                  and to have characters other than Y, M,
                  and D in the mask
"abcYY+MM" - OK not to have a sequence of D's
```

The following date format masks are all invalid date masks:

```
"YYYYyMMDD" - invalid there are 5 Y's in a sequence
"YYYYMDDM" - invalid there are 2 sequences of M's
```

If a date format mask is invalid, no error will be issued. It will just be ignored. Just because a date mask is valid does not mean it will be used. Date format transformation based on a valid date mask will only be performed if ALL of the following conditions are true:

1. There is no SQL error.
2. The output is a date value in ISO-like (ISO and JIS) format.
3. The output data area is at least 10 bytes long. This is the minimum size of an output data area in order for a data value to be stored there even if NO date format transformation is to be performed. This requirement applies even if the date format mask ends up being shorter than 10 bytes.
4. There is a valid date format mask specified in the DCS directory entry and this mask fits in the output data area.

////////BIDI=<ccsid>

This parameter, the ninth positional parameter, is used to specify the Bidirectional (BiDi) CCSID to be used to override the default server database BiDi CCSID. For example:

",,,,,,,,,BIDI=xyz"

where *xyz* represents the CCSID override.

Directory customization worksheet

The directory customization worksheet shows the information that you need to collect. You might find it convenient to make a copy of the worksheet and enter your system values.

Node Directory Parameters

Table 13. Node Directory Parameters

Parameter	Example	Your value
Node name	DB2NODE	
Remote hostname (TCP/IP node)	ZOSHOST	
Server (TCP/IP service name or port number)	db2inst1c (or 446)	

Note:

1. The default TCP/IP port number for DRDA is 446
2. Unless you know that the IBM mainframe database server supports SECURITY SOCKS, do not specify **SECURITY** for a TCP/IP node.

DCS Directory Parameters

Table 14. DCS Directory Parameters

Parameter	Example	Your value
Database name	DB2DB	
Target database name	NEW_YORK3	
Application requester		
Parameter string	",,,,,,,,LOCALDATE=\\\\"YYMMDD\\\\"\\\\"	

System Database Directory Parameters

Table 15. System Database Directory Parameters

Parameter	Example	Your value
Database name	DB2DB	
Database alias	NYC3	
Node name	DB2NODE	
Authentication	SERVER	

Defining multiple entries for the same database

For each database, you must define at least one entry in each of the three directories (node directory, DCS directory, and system database directory). In some cases, you might want to define more than one entry for the database.

For example, you might want to turn off SQLCODE mapping for applications that were ported from the IBM mainframe database server but accept the default mapping for applications that were developed for the client/server environment. You would do this as follows:

- Define one entry in the node directory.
- Define two entries in the DCS directory, with different database names. For one entry, specify **NOMAP** in the parameter string.
- Define two entries in the system database directory, with different database aliases and the two database names that you specified in the DCS directory.

Both aliases access the same database, one with SQLCODE mapping and the other without SQLCODE mapping.

Handling BiDi data

The following section applies to z/OS servers only. This feature must not be enabled for a IBM DB2 for IBM i server as full BiDi support is already provided.

The following BiDi attributes are required for correct handling of BiDi data on different platforms:

- Numeral shape (ARABIC versus HINDI)
- Orientation (RIGHT-TO-LEFT versus LEFT-TO-RIGHT)
- Shaping (SHAPED versus UNSHAPED)
- Symmetric swapping (YES or NO)
- Text type (LOGICAL versus VISUAL)

Since defaults on different platforms are not the same, problems appear when DB2 data is sent from one platform to another. For example, Windows platforms use LOGICAL UNSHAPED data, while z/OS data is usually in SHAPED VISUAL format. Therefore, without any support for BiDi attributes, data sent from DB2 for z/OS to DB2 Connect on Windows displays incorrectly.

When data is exchanged between DB2 Connect and a database on a server, it is usually the receiver that performs conversion on the incoming data.

The same convention would normally apply to BiDi layout transformation also, which is in addition to the usual code page conversion.

However, currently no host DB2 database product supports BiDi-specific CCSIDs or BiDi layout transformation. Therefore, DB2 Connect has been enhanced with the optional ability to perform BiDi layout transformation on data it is about to send to the server database in addition to data received from the server database.

For DB2 Connect to perform BiDi layout transformation on outgoing data to a server database, the BiDi CCSID of the server database will have to be overridden. This is accomplished through the use of the **BIDI** parameter in the **PARMS** field of the DCS database directory entry for the server database.

The use of this feature is best illustrated with an example.

Consider a Hebrew IBM data server client running CCSID 62213 (BiDi string type 5) and you would like to access a DB2 host database running CCSID 424 (BiDi string type 4). However, you know that the data contained in the DB2 host database is instead based on CCSID 62245 (BiDi string type 10).

There are two problems in this situation. The first is that the DB2 host database does not know the difference between the BiDi string types with CCSIDs 424 and 62245. The second problem is that the DB2 host database does not recognize the IBM data server client CCSID of 62213. It only supports CCSID 62209 (BiDi string type 10), which is based on the same code page as CCSID 62213.

You will need to make sure that data sent to the DB2 host database is in BiDi string type 6 format to begin with and also let DB2 Connect know that it has to perform BiDi layout transformation on data it receives from the DB2 host database. You will use the following cataloging for the DB2 host database:

```
catalog dcs database nydb1 as TELAVIV parms ",,,,,,,,,BIDI=62245"
```

This tells DB2 Connect to override the DB2 host database CCSID of 424 with 62245. This override includes the following processing:

1. DB2 Connect will connect to the DB2 host database using CCSID 62209 (BiDi string type 10).
2. DB2 Connect will perform BiDi layout transformation on data it is about to send to the DB2 host database from CCSID 62213 (BiDi string type 5) to CCSID 62209 (BiDi string type 10).
3. DB2 Connect will perform BiDi layout transformation on data it receives from the DB2 host database from CCSID 62245 (BiDi string type 10) to CCSID 62213 (BiDi string type 5).

Note:

1. The environment variable or registry value **DB2BIDI** has to be set to YES in order for the BIDI parameter to take effect. **DB2BIDI** must be set at the DB2 Connect workstation where the DCS database directory entry is catalogued. For applications running on a client remote to a DB2 Connect server, the **DB2BIDI** variable must be set at that client as well.
2. If you would like DB2 Connect to perform layout transformation on data it is about to send to the DB2 host database even though you do not have to override its CCSID, you still have to add the BIDI parameter in the DCS database directory PARMS field. In this case, the CCSID that you should provide would be the default DB2 host database CCSID.
3. In some cases, use of a bidirectional CCSID might cause the SQL query itself to be modified such that it is not recognized by the DB2 server. Specifically, you should try to avoid using IMPLICIT CONTEXTUAL and IMPLICIT RIGHT-TO-LEFT CCSIDs when a different string type can be used. CONTEXTUAL CCSIDs can produce unpredictable results if the SQL query contains quoted strings. Avoid using quoted strings in SQL statements, and use host variables instead when possible.

If a specific bidirectional CCSID is causing problems which cannot be rectified by following these recommendations, then you should set the environment variable or registry value **DB2BIDI** to NO.

Parameter string specifications

The following examples show samples of DCS parameters (each line is a set of parameters):

```
NOMAP
/u/username/sqllib/map/dcs1new.map,D
,D
,,INTERRUPT_ENABLED
NOMAP,D,INTERRUPT_ENABLED,,,SYSPLEX,LOCALDATE="YYMMDD",,
```

Alternatively you can accept the defaults by not specifying a parameter string.

Note: You must use the operating system escape character "\" (backslash) when using CLP from the operating system's command line on UNIX systems because of the need to specify two pairs of double quotation marks when specifying the LOCALDATE mask in the parameter string. For example:

```
db2 catalog dcs db x as y parms \",,,,,,LOCALDATE=\\\"YYMMDD\\\"\\\"\\\"
```

This results in the following DCS directory entry:

DCS 1 entry:

Local database name	= X
Target database name	= Y
Application requestor name	=
DCS parameters	= ,,,,,,LOCALDATE="YYMMDD"
Comment	=
DCS directory release level	= 0x0100

DB2 Connect and SQL statements

DB2 Connect forwards SQL statements submitted by application programs to IBM mainframe database servers.

DB2 Connect can forward almost any valid SQL statement, as well as the supported DB2 APIs (application programming interfaces):

- JDBC
- SQLJ
- ADO.NET
- OLE DB
- ODBC
- Perl
- PHP
- pureQuery
- Python
- Ruby
- CLI
- Embedded SQL

Embedded SQL support

Two types of embedded SQL processing exist: static SQL and dynamic SQL. Static SQL minimizes the time required to execute an SQL statement by processing in advance. Dynamic SQL is processed when the SQL statement is submitted to the IBM mainframe database server. Dynamic SQL is more flexible, but potentially slower. The decision to use static or dynamic SQL is made by the application programmer. Both types are supported by DB2 Connect.

Different IBM mainframe database servers implement SQL differently. DB2 Connect fully supports the common IBM SQL, as well as the DB2 for z/OS, DB2 Server for VM and VSE (formerly SQL/DS), and IBM DB2 for IBM i implementations of SQL. IBM SQL is strongly recommended for maintaining database independence.

Multisite Updates

Multisite update, also known as distributed unit of work (DUOW) and two-phase commit, is a function that enables your applications to update data in multiple remote database servers with guaranteed integrity. DB2 database products provide comprehensive support for multisite updates.

For example, a banking transaction that involves the transfer of money from one account to another in a different database server. In such a transaction, it is critical

that updates which implement debit operations on one account do not get committed unless updates required to process credits to the other account are committed as well. The multisite update considerations apply when data representing these accounts is managed by two different database servers.

The multi-site update support provided by DB2 database products is available for applications developed using regular SQL as well as for applications that use transaction processing monitors (TP monitors) that implement the X/Open XA interface specification. Examples of such TP monitors products include IBM TxSeries CICS, IBM Message and Queuing Series, IBM Component Broker Series, IBM San Francisco Project as well as Microsoft Transaction Server (MTS), BEA Tuxedo and several others. There are different setup requirements depending on whether native SQL multisite update or TP monitor multisite update is used.

XA connections using IBM Data Server Driver Package against a z/OS server are supported. However, XA connections against a System i server are not supported. For details, see the topic about IBM data server driver restrictions.

Both the native SQL and TP monitor multisite update programs must be precompiled with the CONNECT 2 SYNCPOINT TWOPHASE options. Both can use the SQL Connect statement to indicate which database they want to be used for the SQL statements that follow. If there is no TP monitor to tell DB2 it is going to coordinate the transaction (as indicated by DB2 receiving the xa_open calls from the TP monitor to establish a database connection), then the DB2 software will be used to coordinate the transaction.

When using TP monitor multisite update, the application must request commit or rollback by using the TP monitor's API, for example CICS SYNCPOINT, MTS SetAbort(). When using native SQL multisite update, the normal SQL COMMIT and ROLLBACK must be used.

TP monitor multisite update can coordinate a transaction that accesses both DB2 resource managers and resource managers that are not part of DB2, such as Oracle, Informix or SQLServer. Native SQL multisite update is used with DB2 servers only.

For a multisite update transaction to work, each of the databases participating in a distributed transaction must be capable of supporting a distributed unit of work (DUOW). Currently, the following DB2 servers provided DUOW support that enabled them to participate in distributed transactions:

- DB2 for Linux, UNIX and Windows Version 8 or later
- DB2 for z/OS Version 7 or later
- IBM DB2 for IBM i

A distributed transaction can update any mix of supported database servers. For example, your application can update several tables in a DB2 database on Windows, a DB2 for z/OS database, and a DB2 for i database, all within a single transaction.

Multisite update and sync point manager for DB2 Connect server

IBM mainframe database servers require DB2 Connect to participate in a distributed transaction originating from Linux, Windows, UNIX, and web applications. In addition, many of the multisite update scenarios that involve IBM mainframe database servers require that the sync point manager (SPM) component be configured.

When a DB2 instance is created, the DB2 SPM is automatically configured with default settings.

The need for SPM is dictated by the choice of protocol (TCP/IP) and use of a TP monitor. The following table provides a summary of scenarios that require the use of SPM. The table also shows if DB2 Connect is required for any access to the IBM mainframe from Intel or UNIX machines. For multisite updates, the SPM component of DB2 Connect is required if you are using a TP monitor.

Table 16. Multisite update scenarios that require SPM - TCP/IP

Transaction Processor Monitor Used?	Sync Point Manager Needed?	Product Required (Choose One)	IBM mainframe Database Supported
Yes	Yes	DB2 Connect server product DB2 Enterprise Server Edition with DB2 Connect license applied	DB2 for z/OS V8 or later
No	No	DB2 Connect server product DB2 Enterprise Server Edition with DB2 Connect license applied	DB2 for z/OS V8 or later

Note: A distributed transaction can update any mix of supported database servers. For example, your application can update several tables in a DB2 database on Windows, a DB2 for z/OS database and a IBM DB2 for IBM i database all within a single transaction.

Configuring DB2 Connect server with an XA compliant transaction manager

This topic describes the configuration steps necessary to use IBM Power Systems and System z database servers within your TP monitor. These steps are not required if you are using the IBM data server package through DB2 Connect client. For details, see the topic about IBM data server client types.

Before you begin

You must have an operational TP monitor and have DB2 Connect installed, as well as have configured and tested a connection to the IBM mainframe database server.

Procedure

To configure DB2 Connect to use IBM Power Systems and System z database servers within your TP monitor, perform the following steps:

1. Configure the TP monitor so that it can access the DB2 XA Switch. The DB2 XA Switch provides the TP monitor with the addresses of DB2 Connect's XA APIs. Every TP monitor has a different way to do this.
2. Configure the TP monitor with the XA_OPEN string of DB2 product. Each TP monitor has its own way to do this. For information about how to configure XA OPEN string of the DB2 product, for use by the TP monitor, refer to your TP monitor's documentation.

3. If required, modify the DB2 Connect sync point manager (SPM) default configuration parameters. IBM host and System i (Version 5 Release 3 and earlier) database servers do not yet support the XA interface. System i Version 5 Release 4 and following has full XA support.

The SPM is a component of DB2 Connect which maps the XA two phase commit protocol into the two phase commit protocol used by IBM mainframe database servers. By default, the DB2 instance has predefined values for the SPM configuration parameters. The most significant parameter is the database manager configuration parameter **spm_name**. It defaults to a variant of the first seven characters of the TCP/IP hostname.

4. On DB2 for Linux, UNIX, and Windows, set the **DB2COMM** registry variable to use TCPIP, and set the **svcename** database manager configuration parameter to a TCP/IP port number or service name.

DB2 Connect support for loosely coupled transactions

The support within DB2 Connect for loosely coupled transactions is intended for users who implement XA distributed applications that access IBM DB2 for IBM i Version 5 Release 4 or later; and DB2 for z/OS Version 7 or later. This support allows different branches of the same global transaction to share lock space on DB2 for z/OS.

Support for loosely coupled transactions is intended for .NET and COM+ applications.

This feature reduces the window where one branch of a distributed transaction encounters lock timeout or deadlock as a result of another branch within the same global transaction.

SQLCODE mapping

Different IBM relational database products do not always produce the same SQLCODEs for similar errors. Even when the SQLCODE is the same, it might be accompanied by tokens that are specified differently. The token list is passed in the SQLERRMC field of the SQLCA. By default, DB2 Connect maps SQLCODEs and tokens from each IBM mainframe database server to the appropriate DB2 SQLCODEs.

If you want to turn off SQLCODE mapping, specify **NOMAP** in the parameter string of the DCS directory.

If you port an application directly from IBM mainframe database server, such as DB2 for z/OS, you might want to turn off SQLCODE mapping. This would let you use the application without changing the SQLCODEs that it references.

Turning off SQLCODE mapping

If you port an application directly from a IBM mainframe database server, such as DB2 for z/OS, you might want to turn off SQLCODE mapping. This would let you use the application without changing the SQLCODEs that it references.

About this task

If you want to turn off SQLCODE mapping, specify **NOMAP** in the parameter string of the DCS directory.

If you port an application directly from a IBM mainframe database server, such as DB2 for z/OS, you might want to turn off SQLCODE mapping. This would let you use the application without changing the SQLCODEs that it references.

Note: SQLCODE mapping can also be turned off using SQLCODEMAP CLI/ODBC configuration keyword or SQL_ATTR_SQLCODEMAP connection attribute when used with DB2 CLI application programming interface (API).

Tailoring the SQLCODE mapping

By default, DB2 Connect maps SQLCODEs and tokens from each IBM mainframe database server to the appropriate DB2 SQLCODEs. You can tailor the SQLCODE mapping if you want to override the default SQLCODE mapping or if you are using a IBM mainframe database server that does not have SQLCODE mapping (not an IBM database server).

About this task

The following files are copies of the default SQLCODE mapping:

- dcs1dsn.map maps DB2 for z/OS SQLCODEs.
- dcs1ari.map maps DB2 Server for VM and VSE SQLCODEs.
- dcs1qsq.map maps IBM DB2 for IBM i SQLCODEs.

No mapping is required for DB2 on Linux or UNIX operating systems.

Each mapping file is an ASCII file, which is created and edited using an ASCII editor. At initial installation, the file is stored in the map directory in the installation path.

Procedure

If you want to create an SQLCODE mapping for a database server that is not an IBM database server or override the default SQLCODE mapping:

1. Copy one of the dcs1dsn.map, dcs1ari.map, or dcs1qsq.map files and use it as the basis for your new SQLCODE mapping file. By copying the file rather than editing it directly, you ensure that you can always refer to the original SQLCODE mapping, if necessary.
2. Specify the file name of your new SQLCODE mapping file in the parameter string of the DCS Directory.
3. Edit the new SQLCODE mapping file.

The file can contain the following special types of lines:

&& The logical beginning of the file. All lines before the first occurrence of && are considered free-form comments and ignored. If the file contains nothing after &&, no SQLCODE mapping is performed. You can also turn off SQLCODE mapping with the **NOMAP** parameter, as described previously.

***** As the first character on a line, indicates a comment.

W As the only character on a line, indicates that warning flags should be remapped. By default, the original warning flags are passed. The W must be uppercase.

All other lines after && must be either blank or mapping statements in the following form:

```
input_code [, output_code [, token_list]]
```

The *input_code* represents one of the following values:

sqlcode The SQLCODE from the IBM mainframe database server.

U All undefined negative SQLCODEs (those not listed in this file) are mapped to the specified *output_code*. If no *output_code* is specified on this line, the original SQLCODE is used. This character must be uppercase.

P All undefined positive SQLCODEs (those not listed in this file) are mapped to the specified *output_code*. If no *output_code* is specified on this line, the original SQLCODE is used. This character must be uppercase.

ccnn The SQLSTATE class code from the IBM mainframe database server. *nn* is one of the following values:

00 Unqualified successful completion

01 Warning

02 No data

21 Cardinality violation

22 Data exception

23 Constraint violation

24 Invalid cursor state

26 Invalid SQL statement identifier

40 Transaction Rollback

42 Access violation

51 Invalid application state

55 Object not in prerequisite state

56 Miscellaneous SQL or Product Error

57 Resource not available or operator intervention

58 System error

The specified *output_code* is used for all SQLCODEs with this class code that are not specified explicitly in the mapping file. If no *output_code* is specified on this line, the original SQLCODE is mapped to itself with no tokens copied over.

The characters **cc** must be lowercase.

If the same *input_code* appears more than once in the mapping file, the first occurrence is used. The *output_code* represents the output SQLCODE. If no value is specified, the original SQLCODE is used.

If you specify an output code, you can also specify one of the following value:

(s) The input SQLCODE plus the product ID (ARI, DSN or QSQ) will be put into the SQLCA message token field.

The original SQLCODE is returned as the only token. This option is designed to handle undefined SQLCODEs, with the exception of +965 and -969. If +965 or -969 is the *output_code*, the token list returned in the SQLERRMC field of the SQLCA includes the original SQLCODE, followed by the product identifier, followed by the original token list.

The character **s** must be lowercase.

(token-list)

A list of tokens, separated by commas. Specify only a comma to skip a particular token. For example, the form *(,t2,,t4)* means that the first and third output tokens are null.

Each token has the form of a number (*n*), optionally preceded by **c**, optionally followed by **c** or **i**. It is interpreted as follows:

- c** The data type of the token in this position is CHAR (the default). If **c** comes before *n*, it refers to the input token; if it comes after *n*, it refers to the output token. The character **c** must be lowercase.
- i** The data type of the token in this position is INTEGER. If **i** comes after *n*, it refers to the output token. **i** should not come before *n*, because IBM mainframe database server products support only CHAR tokens. The character **i** must be lowercase.
- n* A number or numbers indicating which IBM mainframe database server tokens are used. They are arranged in the order required for placement in the output SQLCA. The number indicates the IBM mainframe database server token; the arrangement indicates the order in which the tokens will be placed in the SQLCA.

For example, the IBM mainframe database server might return two tokens, 1 and 2. If you want token 2 to appear before token 1 in the output SQLCA, specify *(2,1)*.

Multiple token numbers can be combined to form one CHAR output token by connecting them with periods.

Commas are used to separate output tokens. If no token is specified before a comma, no output token is included in the SQLCA for that position. Any tokens occurring in the output SQLCA following the last specified token are mapped to a null token.

Example

Figure 8 shows a sample SQLCODE mapping file.

```
&&
-007      ,   -007      ,   (1)
-010
-060      ,   -171      ,   (2)
...
-204      ,   -204      ,   (c1.2c)
...
-633      ,   -206      ,   (,c1i)
-30021    ,   -30021    ,   (c1c,c2c)
cc00      ,   +000
...
U         ,   -969      ,   (s)
P         ,   +965      ,   (s)
```

Figure 8. An SQLCODE Mapping File

The following descriptions correspond to the matching line number in the previous figure:

1. The SQLCODE is mapped from -007 to -007. The first input token received from the IBM mainframe database server is used as the first output token, and it defaults to CHAR. No other tokens are transferred.
2. The SQLCODE is mapped from -010 to -010 (no output SQLCODE is specified). No tokens are put into the output SQLCA.
3. The SQLCODE is mapped from -060 to -171. The first input token received from the IBM mainframe database server is discarded. The second is used as the first token in the output SQLCA, and it is CHAR. There is no second token in the output SQLCA.
4. The SQLCODE is mapped from -204 to -204. The first and second tokens received from the IBM mainframe database server are CHAR. These two input tokens are combined to form one CHAR output token, which will be the first output token in the SQLCA.
5. The SQLCODE is mapped from -633 to -206. The first input token received from the IBM mainframe database server is CHAR. It is converted to INTEGER and is used as the second token in the output SQLCA. The first token in the output SQLCA is null, as indicated by a comma.
6. The SQLCODE is mapped from -30021 to -30021. The first and second input tokens received from the IBM mainframe database server are CHAR, and they are used as the first and second tokens in the output SQLCA.
7. All SQLCODEs in SQLCAs with SQLSTATEs in the 00 class will be mapped to SQLCODE +000.
8. All undefined SQLCODEs are mapped to -969. This option should be used only if all mappable codes are listed, including all those that are identical and require no mapping. The (s) option indicates that the token list to be returned in the SQLERRMC field of the SQLCA includes the original SQLCODE, followed by the product the error occurred in, followed by the original token list. If the U entry is not included, all unlisted codes are passed without any mapping.
9. All undefined positive SQLCODEs are mapped to +965. This option should be used only if all mappable codes are listed, including all those that are identical and require no mapping. The (s) option indicates that the token list to be returned in the SQLERRMC field of the SQLCA includes the original SQLCODE, followed by the product the warning occurred in, followed by the original token list. If the P entry is not included, all unlisted positive codes are passed without any mapping.

Chapter 6. Monitoring DB2 Connect server

Monitoring connections for remote clients

You can use the database system monitor with a DB2 Connect server product, such as DB2 Connect Enterprise Edition, to monitor the remote client connections.

To monitor clients that are local to the DB2 Connect server, that are running on the server itself, you will need to set the following variable:

```
db2set DB2CONNECT_IN_APP_PROCESS=NO
```

For example, when an error occurs at the IBM mainframe system, the system administrator can determine if the problem was on the DB2 Connect workstation. The database system monitor correlates:

- The DRDA correlation token (CRRTKN), for unprotected conversations.
- The unit of work id (UOWID), for two-phase connections protected by the DRDA-3 sync point manager (as used over TCP/IP connections).
- The DB2 Connect connection identifier (the Application ID).

This information shows which DB2 Connect connection caused the problem, which allows the system administrator to force the individual client application from the system without affecting the other clients using the DB2 Connect connection.

Listing the Status of Monitor Switches

To list the status of monitor switches, use the **db2 get monitor switches** command.

Monitoring performance using the Windows Performance Monitor

Windows operating systems provide a useful tool for monitoring the performance of your DB2 applications. The Performance Monitor, which is one of the Windows administrative tools, displays a graphical representation of system performance.

You can choose a variety of system, database, and communications-related items to monitor and map them together in a graphical representation.

For example, the reports available through the **GET SNAPSHOT FOR ALL DCS DATABASES** or **GET SNAPSHOT FOR ALL DCS APPLICATIONS** commands can be graphed in real time using the monitor, and compared directly with values such as CPU usage. You can directly compare the effects of different settings on database or communications performance. You can save your specialized configurations of settings in PMC files that you can later retrieve.

For example in the following figure, several DB2 measures are being graphed against CPU usage. The collection of values being charted was saved in the file `db2chart.pmc`. You can save as many PMC files as you like, each reflecting a different cross-section of system performance.

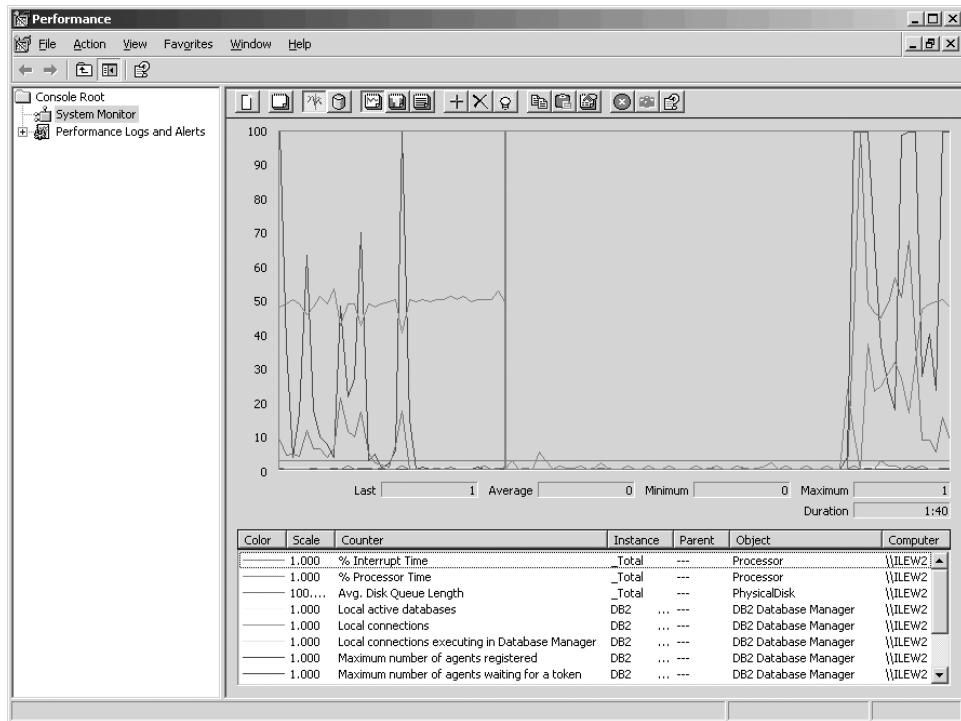


Figure 9. Performance Monitor

To enable monitoring of local applications you will need to turn off the **DB2CONNECT_IN_APP_PROCESS** environment variable.

Using the GET SNAPSHOT commands

The DB2 monitor maintains a running tally of valuable system information. You can get a summary of system status at any time by issuing the **GET SNAPSHOT** command.

You can take monitor snapshots if you have SYSMAINT, SYSCTRL, or SYSADM authority for the database manager instance that you want to monitor.

There are five snapshot commands useful for monitoring DCS information. They are:

- **GET SNAPSHOT FOR ALL DCS DATABASES**
- **GET SNAPSHOT FOR ALL DCS APPLICATIONS**
- **GET SNAPSHOT FOR DCS APPLICATION ...**
- **GET SNAPSHOT FOR DCS DATABASE ON *db_alias***
- **GET SNAPSHOT FOR DCS APPLICATIONS ON *db_alias***

Each snapshot command will produce a detailed report about the area you requested.

For instance, issuing the **GET SNAPSHOT FOR DCS DATABASE ON DCSDB** will produce the following report:

DCS Database Snapshot

DCS database name	= DCSDB
Host database name	= GILROY
First database connect timestamp	= 12-15-2001 10:28:24.596495

```

Most recent elapsed time to connect      = 0.950561
Most recent elapsed connection duration = 0.000000
Host response time (sec.ms)             = 0.000000
Last reset timestamp                    =
Number of SQL statements attempted      = 2
Commit statements attempted             = 1
Rollback statements attempted           = 0
Failed statement operations              = 0
Total number of gateway connections     = 1
Current number of gateway connections  = 1
Gateway conn. waiting for host reply    = 0
Gateway conn. waiting for client request = 1
Gateway communication errors to host    = 0
Timestamp of last communication error    = None
High water mark for gateway connections = 1
Rows selected                           = 0
Outbound bytes sent                     = 140
Outbound bytes received                  = 103

```

This report provides information about the database connections, performance, errors and throughput of SQL requests. DB2 Monitor snapshots can be much more detailed, in fact. For instance, if you issue the **GET SNAPSHOT FOR ALL DCS APPLICATIONS** command, you will receive a report similar to the following one:

DCS Application Snapshot

```

Client application ID          = 09150F74.B6A4.991215152824
Sequence number                = 0001
Authorization ID               = SMITH
Application name                = db2bp
Application handle              = 1
Application status              = waiting for request
Status change time             = 12-15-2001 10:29:06.707086
Client node                     = sys143
Client release level            = SQL06010
Client platform                 = AIX
Client protocol                 = TCP/IP
Client codepage                 = 850
Process ID of client application = 49074
Client login ID                 = smith
Host application ID             = G9150F74.B6A5.991215152825
Sequence number                 = 0000
Database alias at the gateway   = MVSDB
DCS database name               = DCSDB
Host database name              = GILROY
Host release level              = DSN05012
Host CCSID                      = 500

Outbound communication address   = 9.21.21.92 5021
Outbound communication protocol = TCP/IP
Inbound communication address    = 9.21.15.116 46756
First database connect timestamp = 12-15-2001 10:28:24.596495
Host response time (sec.ms)     = 0.000000
Time spent on gateway processing = 0.000000
Last reset timestamp            =
Rows selected                    = 0
Number of SQL statements attempted = 2
Failed statement operations       = 0
Commit statements                 = 1
Rollback statements              = 0
Inbound bytes received           = 404
Outbound bytes sent               = 140
Outbound bytes received          = 103
Inbound bytes sent               = 287
Number of open cursors           = 0
Application idle time            = 1 minute and 32 seconds

```

```

UOW completion status                =
Previous UOW completion timestamp    = 12-15-2001 10:28:25.592631
UOW start timestamp                  = 12-15-2001 10:29:06.142790
UOW stop timestamp                   =
Elapsed time of last completed uow (sec.ms)= 0.034396

Most recent operation                 = Execute Immediate
Most recent operation start timestamp = 12-15-2001 10:29:06.142790
Most recent operation stop timestamp  = 12-15-2001 10:29:06.707053

Statement                           = Execute Immediate
Section number                       = 203
Application creator                   = NULLID
Package name                         = SQLC2C07
SQL compiler cost estimate in timerons = 0
SQL compiler cardinality estimate     = 0
Statement start timestamp             = 12-15-2001 10:29:06.142790
Statement stop timestamp              = 12-15-2001 10:29:06.707053
Host response time (sec.ms)           = 1.101612
Elapsed time of last completed stmt(sec.ms)= 0.564263
Rows fetched                         = 0
Time spent on gateway processing      = 0.013367
Inbound bytes received for statement = 220
Outbound bytes sent for statement     = 130
Outbound bytes received for statement = 49
Inbound bytes sent for statement      = 27
SQL statement text:
create table t12 (col1 int, col2 char)

```

DCS application status

Use the Database Connection Services (DCS) application status to retrieve information about the applications connected to the database. There are three application status commands you can use, which return different levels of detail.

The System Monitor provides three forms of the **LIST DCS APPLICATIONS** command, as follows:

- **LIST DCS APPLICATIONS**
- **LIST DCS APPLICATIONS SHOW DETAIL**
- **LIST DCS APPLICATIONS EXTENDED**

In the output that follows, the format for the Host Application ID and Client Application ID can differ depending on the IBM mainframe database version and the TCP/IP support level.

Table 17. Application ID format based on host version and TCP/IP support level

Scenario	Application ID format
Clients accessing data servers with RDB Manager Level support less than 7	G91A0D3A.P8BC.060306212019
Clients accessing data servers with RDB Manager level support 8 or greater over TCP/IP v4	9.26.13.61.65289.060306213816

Table 17. Application ID format based on host version and TCP/IP support level (continued)

Scenario	Application ID format
Clients accessing data servers with RDB Manager level support 8 or greater over TCP/IP v6	2002:91a:519:13:209:6bff:fe14:4fbb.7684.060306213741

LIST DCS APPLICATIONS

To view the information provided by the monitor at the application level, issue the **DB2 LIST DCS APPLICATIONS** command.

It returns the following information for a TCP/IP connection (DB2 Connect to DB2 for z/OS):

Auth Id	Application Name	Appl. Handle	Host Application Id
NEWTON	db2cli.exe	7	G91A0D3A.P8BC.060306212019
NEWTON	db2cli.exe	25	9.26.13.61.65289.060306213816
NEWTON	db2cli.exe	20	2002:91a:519:13:209:6bff:fe14:4fbb.7684.060306213741

Auth.Id

The authorization ID that was used to log on to the IBM mainframe database server. This identifies who is running the application.

Application Name

The name of the application running at the client as known to DB2 Connect. Only the first 20 bytes after the last path separator are available.

Appl. Handle

The agent that is executing on the DB2 Connect workstation. You can use this element to link the database system monitor information to other diagnostic information. The agent ID is also required when using the FORCE USERS command or API.

Host Application ID

One of the following items:

- The DRDA correlation token (CRRTKN), for unprotected conversations.
- The unit of work id (UOWID), for two-phase connections protected by the DRDA-3 Syncpoint Manager (as used over TCP/IP connections).

This unique identifier is generated when the application connects to the IBM mainframe database server. You can use this element in conjunction with the Application ID to correlate the client and server parts of the application information.

LIST DCS APPLICATIONS SHOW DETAIL

If the **DB2 LIST DCS APPLICATIONS SHOW DETAIL** command format is specified, additional information is shown, including:

Auth Id		Application Name		Appl. Handle	Client Application Id
NEWTON		db2cli.exe		37	2002:91a:519:13:209:6bff:fe14:4fbb.8196.060306214224
Seq#	Client DB Alias	Client Node	Client Release	Client Codepage	Host Application Id
00001	MDB	SAYYID	SQL09000	1252	G91A0D3A.P982.060306214231
Seq#	Host DB Name		Host Release		
00001	MEXICO		DSN08015		

Client Application ID

Uniquely identifies the application connected to the DB2 Connect workstation. There are different formats for the application ID, which are dependent on the communication protocol between the client and the DB2 Connect workstation.

This value lets you correlate connections from clients to the DB2 Connect workstation and from the DB2 Connect workstation to the IBM mainframe database server.

Client Sequence no (Seq#)

The client sequence number is the transaction sequence number. It is used to help correlate a transaction spread over different systems.

Client DB alias

The alias of the database provided by the application to connect to the database. This element can be used to identify the actual database that the application is accessing. The mapping between this name and the database name could be done by using the database directories at the client node and the database manager server node.

Client NNAME (Node)

Identifies the node where the client application is executing. The information varies according to the client protocol in use. For a client connected via TCP/IP, this is the host name.

Client Product ID (Client)

The product and version that is running on the client. The client product IDs will be:

- SQL07010 for Version 7.1 of DB2 Universal Database™ and DB2 Connect products and their clients.
- SQL08010 for Version 8.1 of DB2 Universal Database and DB2 Connect products and their clients.
- SQL08020 for Version 8.2 of DB2 Universal Database and DB2 Connect products and their clients.
- SQL09120 for Version 9.1 of DB2 products, DB2 Connect products, and their clients.

Code Page ID

The code page identifier at the node where the monitored application started.

You can use this information to ensure that data conversion is supported between the application code page and the database code page (or for IBM mainframe database server databases, the IBM mainframe database server CCSID).

If the application code page is different from that under which the database system monitor is running, this code page element can help you to manually convert data that was passed from the application and displayed by the database system monitor. For example, you can use it to help translate the Application Name.

Outbound Sequence No

This represents the outbound sequence number. It is used for correlating transactions on different systems.

Host Database Name

The real name of the database to which the application is connected. In the DCS directory, this is the *target database name*.

Host Product ID

The product and version that is running on the server. It is in the form *PPPVVRRM*, where:

- PPP** Identifies the IBM mainframe database server product (for example, DSN for DB2 Universal Database for z/OS and OS/390®, ARI for DB2 Server for VSE & VM, or QSQ for IBM DB2 for IBM i)
- VV** Represents a two-digit version number, such as 08.
- RR** Represents a two-digit release number, such as 01.
- M** Represents a one-character modification level (0-9 or A-Z).

LIST DCS APPLICATIONS EXTENDED

You can use the **LIST DCS APPLICATIONS** command with the option **EXTENDED** in order to generate an Extended Report. The Extended Report lists all the fields that are listed when the **SHOW DETAIL** option is specified on the command, plus nine new fields:

- DCS application status
- Status change time
- Client platform
- Client protocol
- Host Coded Character Set Identifier (CCSID).
- Client login ID
- Process ID of client application
- Database alias at the gateway
- DCS database name

While the existing command options list the fields horizontally, one line per application, the new option lists them vertically, one field per line.

Here is the new syntax of the command:

```
LIST DCS APPLICATIONS [ SHOW DETAIL | EXTENDED ]
```

And here is sample output from this command, when using the new option **EXTENDED**:

List of DCS Applications - Extended Report

Client application ID	= 2002:91a:519:13:209:6bff:fe14:4fbb.8196.060306214224
Sequence number	= 00001
Authorization ID	= NEWTON
Trusted Authorization ID	=
Application name	= db2cli.exe
Application handle	= 37
Application status	= waiting for request
Status change time	= Not Collected
Client node	= SAYYID
Client release level	= SQL09000
Client platform	= NT
Client protocol	= TCP/IP
Client codepage	= 1252
Process ID of client application	= 1192
Client login ID	= ISAYYID
Host application ID	= G91A0D3A.P982.060306214231
Sequence number	= 00001
Database alias at the gateway	= MDB
DCS database name	= MDB
Host database name	= MEXICO
Host release level	= DSN08015
Host CCSID	= 1208

The application status field contains one of the following three values:

1. connect pending - outbound. This means that the request to connect to a IBM mainframe database has been issued, and DB2 Connect is waiting for the connection to be established.
2. waiting for request. This means that the connection with the IBM mainframe database has been established, and that DB2 Connect is waiting for an SQL statement from the client application
3. waiting for reply. This means that the SQL statement has been sent to the IBM mainframe database.

Also, the status change time is only shown in the report if the System Monitor UOW switch was turned on during processing. Otherwise, "Not Collected" will be shown.

Chapter 7. Developing database applications

Running your own applications

You can build and run DB2 applications with an IBM Data Server Client installed.

Various types of applications can access DB2 databases:

- Applications developed using the IBM data server client that include embedded SQL, APIs, stored procedures, user-defined functions or calls to the CLI
- ODBC applications
- Java applications using the JDBC or SQLJ interfaces
- PHP applications
- Ruby or Ruby on Rails applications
- Perl applications
- Python applications

On Windows operating systems, the following routines or objects can also access DB2 databases:

- ActiveX Data Objects (ADO) implemented in Microsoft Visual Basic and Microsoft Visual C++
- Object Linking and Embedding (OLE) Automation Routines (UDFs and Stored Procedures)
- Object Linking and Embedding Database (OLE DB) table functions

To run an application:

1. Ensure the server is configured and running.
2. On the DB2 server, ensure that the database manager is started on the database server to which the application program is connecting. If it is not, you must issue the **db2start** command at the server before starting the application.
3. Ensure that you can connect to the database that the application uses.
4. Bind the necessary files to support the database application driver being used.
5. Run the application program.

Application compatibility on DB2 for z/OS

In DB2 for z/OS Version 11 and later versions, you can specify whether you want applications to run by using the features and behavior of the current DB2 for z/OS version or of a previous DB2 for z/OS version. For CLI and .NET applications that connect to DB2 for z/OS Version 11 or later data servers, you can control the application compatibility behavior in several ways.

A DB2 for z/OS subsystem that you migrate to a new version goes through several phases. The first phase is conversion mode, in which functions of the new version are not yet available to the system, and the last phase is new-function mode, in which functions of the new version are available. If there is a behavior change between DB2 for z/OS versions, an SQL application uses the old behavior in conversion mode. Application compatibility enables applications that have incompatibilities with the new DB2 for z/OS version to maintain SQL compatibility with a previous DB2 for z/OS version when the subsystem is in

new-function mode. Using application compatibility provides a window for adopting new function and making changes to accommodate incompatibilities. Also, with application compatibility, you can introduce new SQL functions or wait until later when you move to new-function mode.

Application compatibility identifies the functional level that is expected by the application. You can set application compatibility in three ways, in the following order of precedence, where option 1 has the highest precedence:

1. Set the **CURRENT APPLICATION COMPATIBILITY** special register by setting the application compatibility property in the db2dsdriver.cfg file. When a connection is established with the server, this property controls dynamic SQL statement behavior.
2. Bind packages with the APPLCOMPAT option, to control the SQL application compatibility of the packages for static SQL. The APPLCOMPAT value also provides the default for the **CURRENT APPLICATION COMPATIBILITY** special register. You can specify the APPLCOMPAT option for the **GENERIC** parameter of the **BIND** command.

Attention: Setting the APPLCOMPAT option can negatively impact the common packages for JCC, .NET, and CLI. Use a different collection name to find any impact.

3. Set the APPLCOMPAT subsystem parameter on the DB2 for z/OS server.

The following describes the SQL application behavior for application compatibility settings in either conversion mode or new-function mode:

- If you are in conversion mode, you cannot use new DB2 function. Trying to use new DB2 function or trying to set application compatibility might result in SQL4700N.
- If you are in new-function mode, DB2 function behaves based on the application compatibility value for the application on the server.
 - If application compatibility is set for a previous version, and an application uses a DB2 feature that is controlled by application compatibility, an error might occur. In most cases, that error is SQL4743N.
 - If application compatibility is set for the current version, new DB2 function is allowed.

Example

In the following example, the APPLCOMPAT special register is specified in the <specialregisters> subsection of db2dsdriver.cfg configuration file:

```
<configuration>
<dsncollection>
  <dsn alias="sample" name="sample" host="hotel fvt02.torolab.ibm.com" port="21169"/>
  <specialregisters>
    <parameter name="CURRENT APPLICATION COMPATIBILITY" value="V11R1"/>
  </specialregisters>
</dsn>
<dsn alias="sample" name="sample" host="hotel fvt02.torolab.ibm.com" port="21169"/>
</dsn>
</dsncollection>
<databases>
<database name="sample" host="hotel fvt02.torolab.ibm.com" port="21169">
  <specialregisters>
    <parameter name="CURRENT APPLICATION COMPATIBILITY" value="V10R1"/>
  </specialregisters>
</database>
<database name="sample2" host="hotel fvt02.torolab.ibm.com" port="21169">
</database>
</databases>
<parameters>
  <specialregisters>
```

```
<parameter name="CURRENT APPLICATION COMPATIBILITY" value="V10R1"/>
</specialregisters>
</parameters>
</configuration>
```

After a connection is established to the data source name *sample* ('dsn=sample'), the first attempt to run an SQL statement sets the **CURRENT APPLICATION COMPATIBILITY** special register to V11R1. The special register that is specified inside the <specialregisters> subsection of the <dsn> section has precedence over other sections. The <specialregisters> subsection in the <dsn> section takes precedence over the <specialregisters> subsection in the <database> section. The <specialregisters> subsection in the <database> section takes precedence over the <specialregisters> subsection in the <parameters> section.

After a connection is established to the data source name *sample2* ('dsn=sample2'), the first attempt to run an SQL statement sets the **CURRENT APPLICATION COMPATIBILITY** special register to V10R1. The special register is set to V10R1 since it is only found in the <parameters> section.

Chapter 8. Security

Trusted connections through DB2 Connect

Some DB2 database servers support trusted contexts. A *trusted context* allows the database administrator to, among other things, define conditions under which a client application will be allowed to create a trusted connection. A *trusted connection* is allowed to do things that a normal connection cannot.

There are two types of trusted connection, implicit and explicit. When you create a connection, whether you get an explicit trusted connection, an implicit trusted connection, or a regular connection depends on whether you ask for a trusted connection and whether the connection meets the criteria defined in the trusted context on the server, as summarized in Table 18.

Table 18. What type of connections result from different combinations of actions

	The connection meets the server's criteria for being trusted	The connection does not meet the server's criteria for being trusted
You request that the connection be trusted	Explicit trusted connection	Regular connection and warning SQL20360W (SQLSTATE 01679) is returned.
You do not request that the connection be trusted	Implicit trusted connection	Regular connection

An *implicit trusted connection* is identical to a regular connection except that it grants temporary role privileges to the user while they are using the connection. The role privileges that are granted (if any) are specified in the trusted context that caused the connection to be trusted.

Implicit trusted connections can be created by any application that connects using DB2 Connect. Implicit trusted connections are made and used in the same way that regular connections are made and used. This means that no code changes are necessary for an existing application to take advantage of implicit trusted connections as long as the application connects through DB2 Connect.

An *explicit trusted connection* grants temporary role privileges to the user the same way that an implicit trusted connection does. In addition, an explicit trusted connection lets you change the authorization ID used when performing actions across that connection. Changing the authorization ID on an explicit trusted connection is referred to as *switching users*. The authorization IDs to which you can switch and whether a given authorization ID requires a password when switching to it are defined as part of the trusted context that allowed the trusted connection to be created.

User switching can significantly reduce the processing usage of sharing a connection among several users, especially for user names that do not require a password because in that case the database server does not authenticate the authorization ID. When using the feature, however, you must be very certain that

your application does not allow switching to an authorization ID without validating and authenticating that authorization ID. Otherwise you are creating a security hole in your system.

Explicit trusted connections can be created and the user can be switched when connecting through DB2 Connect using CLI or JDBC, including XA established connections. Creating an explicit trusted connection and switching users requires setting special connection attributes. This means that existing applications will need to be modified in order to take advantage of explicit trusted connections.

Other than the differences just mentioned, you can use a trusted connection (whether implicit or explicit) the same way you would use a regular connection. You must be certain, however, to explicitly disconnect an explicit trusted connection when you are done with it, even if it is in a broken or disconnected state. Otherwise resources used by the connection might not be released. This is not a problem with implicit trusted connections.

Note:

1. Explicit trusted connections should not use CLIENT authentication. This does not apply to implicit trusted connections.
2. Applications using explicit trusted connections should be run on secure machines which are password protected and accessible only to authorized personnel. This does not apply to implicit trusted connections.

Creating and terminating a trusted connection through CLI

If the database server you are connecting to is configured to allow it, you can create an explicit trusted connection when connecting through CLI.

Before you begin

This procedure assumes that you are not using an XA transaction manager. If you are using an XA transaction manager you only need to make sure that the transaction manager is configured to set the configuration value TCTX to TRUE when it calls `xa_open`. If that is done then any connection that can be an explicit trusted connection will be. To verify that a connection is an explicit trusted connection see step 3.

- The database that you are connecting to must support trusted contexts.
- A trusted context must be defined that will recognize the client as being trustable.
- You must know the system authorization ID that is specified in the trusted context. The system authorization ID of a trusted connection is the authorization ID you provide to the server as a user name when creating the connection. For your connection to be trusted by a particular trusted context the system authorization ID must be the one specified in that trusted context. Ask your security administrator for a valid system authorization ID and the password for that ID.

About this task

The examples in these instructions use the C language and assume that `conn` is a pointer to a valid, but unconnected, connection handle. The variable `rc` is assumed to have a data type of `SQLRETURN`.

Procedure

1. In addition to setting any connection attributes that you would set for a regular connection, set the connection attribute `SQL_ATTR_USE_TRUSTED_CONTEXT` to `SQL_TRUE` with a call to the `SQLSetConnectAttr` function.

```
rc = SQLSetConnectAttr(  
    conn,  
    SQL_ATTR_USE_TRUSTED_CONTEXT, SQL_TRUE, SQL_IS_INTEGER  
);
```

2. Connect to the database as you would for a regular connection, for example by calling the `SQLConnect` function. Use the system authorization ID as the user name and its password as the password. Be sure to check for errors and warnings, especially those listed in table Table 19.

Table 19. Errors indicating failure to create a trusted connection

SQLCODE	SQLSTATE	Meaning
SQL20360W	01679	The connection could not be established as a trusted connection. It was established as a regular connection instead.

If no errors or warnings tell you differently, then the connection is established and is an explicit trusted connection.

3. Optional: You can verify that an established connection is an explicit trusted connection by checking the value of the connection attribute `SQL_ATTR_USE_TRUSTED_CONTEXT` using the `SQLGetConnectAttr` function. If it is set to `SQL_TRUE` the connection is an explicit trusted connection.
4. When you are finished using the connection you must be very careful to explicitly disconnect it, even if it is in a broken or disconnected state. If you do not explicitly disconnect an explicit trusted connection some of the resources used by the connection might not be released.

Results

Note:

1. Explicit trusted connections should not use `CLIENT` authentication. This does not apply to implicit trusted connections.
2. Applications using explicit trusted connections should only be run on secure computers which are password protected and accessible only to authorized personnel. This does not apply to implicit trusted connections.

Switching users on a trusted connection through CLI

You can switch users on an explicit trusted connection through the command line interface (CLI).

For a description of what it means to switch users using a trusted connection see the topic in the related links.

Before you begin

- The connection must have been successfully created as an explicit trusted connection.
- The explicit trusted connection must not be in a transaction.
- The trusted context that allowed the explicit trusted connection to be created must be configured to allow switching to the authorization ID you are switching to.

About this task

The examples in these instructions use the C language and assume that *conn* is a pointer to a connected explicit trusted connection. The variable *rc* is assumed to have a data type of `SQLRETURN`. The variable *newuser* is assumed to be a pointer to a character string holding the authorization ID of the user you want to switch to. The variable *passwd* is assumed to be a pointer to a character string containing the password for that authorization ID.

Procedure

1. Call the `SQLSetConnectAttr` function to set the `SQL_ATTR_TRUSTED_CONTEXT_USERID` attribute. Set it to the authorization ID you want to switch to.

```
rc = SQLSetConnectAttr(  
    conn,  
    SQL_ATTR_TRUSTED_CONTEXT_USERID, newuser, SQL_NTS  
);  
//Check for errors
```

Be sure to check for errors and warnings, especially those listed in table Table 20.

Table 20. Errors indicating failure to set a new authorization ID when switching users

SQLCODE	Meaning
CLI0106E	The connection is not connected.
CLI0197E	The connection is not a trusted connection.
CLI0124E	There is a problem with the value provided. Check that it is not null, or not too long, for example.
CLI0196E	The connection is involved in a unit of work that prevents it from switching users. To be able to switch users the connection must not be in a transaction.

2. Optional: (This step is optional unless the trusted context that allowed this trusted connection requires a password for the authorization ID you are switching to.) Call the `SQLSetConnectAttr` function to set the `SQL_ATTR_TRUSTED_CONTEXT_PASSWORD` attribute. Set it to the password for the new authorization ID.

```
rc = SQLSetConnectAttr(  
    conn,  
    SQL_ATTR_TRUSTED_CONTEXT_PASSWORD, passwd, SQL_NTS  
);  
//Check for errors
```

Be sure to check for errors and warnings, both those listed in table Table 20 and those listed in table Table 21.

Table 21. Errors indicating failure to set a password when switching users

SQLCODE	Meaning
CLI0198E	The attribute <code>SQL_ATTR_TRUSTED_CONTEXT_USERID</code> has not yet been set.

3. Proceed as with a regular connection. If you are using an XA transaction manager the user switch is attempted as part of the next request, otherwise the user switch is attempted just before initiating the next function call that accesses the database (`SQLExecDirect` for example). In either case, in addition

to the errors and warnings you would normally check for, be sure to check for the errors listed in Table 22. The errors in Table 22 indicate that the user switch failed.

Table 22. Errors indicating failure to switch users

SQLCODE	Meaning
SQL1046N	The trusted context that allowed this trusted connection is not configured to allow switching to the authorization ID you are trying to switch to. You will not be able to switch to that authorization ID until the trusted context is changed.
SQL30082N	The password provided is not correct for the authorization ID you are switching to.
SQL0969N with a native error of -20361	There is some database level constraint that prevent you from switching to the user.

If the user switch fails the connection will be in an unconnected state until you successfully switch to another user. You can switch users on a trusted connection in an unconnected state but cannot access the database server with it. A connection in an unconnected state will remain in that state until you successfully switch users on it.

What to do next

Note:

1. **Important:** Switching users without supplying a password bypasses the database server's authentication. Your application must not allow a switch to an authorization ID without a password unless that application has already validated and authenticated that authorization ID. To do otherwise creates a security hole.
2. Specifying a NULL value for the SQL_ATTR_TRUSTED_CONTEXT_USERID attribute is equivalent to specifying the trusted context system authorization ID (the user id used when the explicit trusted connection was created).
3. When you successfully set the value of the SQL_ATTR_TRUSTED_CONTEXT_USERID connection attribute on an explicit trusted connection the connection is immediately reset. The result of resetting is as if a new connection were created using the original connection attributes of that connection. This reset happens even if the value you set the connection attribute to is the system authorization ID or NULL or the same value that the attribute currently holds.
4. If the SQL_ATTR_TRUSTED_CONTEXT_PASSWORD attribute is set, the password will be authenticated during the switch user processing, even if the trusted context that allowed the trusted connection doesn't require authentication on a switch user for that authorization ID. This results in unnecessary processing time. This rule doesn't apply to the trusted context system authorization ID. If the trusted context system authorization ID doesn't require authentication when you switch to it then it is not authenticated even if a password is provided.

DB2 Connect authentication considerations

As a DB2 Connect administrator, in cooperation with your System z or IBM Power Systems database administrator, you can determine where user names and passwords are validated.

For example:

- At the client
- At the System z or IBM Power Systems server
- Single sign-on and validation through a third-party system (Kerberos).

Note: If the remote client has not specified an authentication type, the client will try to connect using the `SERVER_ENCRYPT` authentication type first. If this type is not accepted by the server, the client will attempt to try using an appropriate value returned from the server. To help optimize performance, always specify the authentication type at the client to avoid this extra network flow.

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

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

The following authentication types are allowed with DB2 Connect:

CLIENT

The user name and password are validated at the client.

DATA_ENCRYPT

Provides the ability to encrypt user data during client/server communications. This authentication type is not supported on IBM Power Systems database server.

KERBEROS

Enables the client to log into the server using Kerberos authentication instead of the traditional ID and password combination. This authentication type requires that both the server and client be Kerberos-enabled.

SERVER

The user name and password are validated at the System z or IBM Power Systems server database.

SERVER_ENCRYPT

As for `SERVER` authentication, the user name and password are validated at the System z or IBM Power Systems database server, but the transferred user IDs and passwords are encrypted at the client.

SERVER_ENCRYPT_AES

The transferred user IDs and passwords are encrypted using an Advanced Encryption Standard (AES) encryption algorithm at the client and validated at the System z database server.

Kerberos authentication is unique in that the client does not pass a user ID and password directly to the server. Instead, Kerberos acts as a third-party authentication mechanism. The user enters an ID and password once at the client terminal, and Kerberos validates this sign-on. After this, Kerberos automatically and securely passes the user's authorization to any local and network services requested. This means that the user does not need to re-enter an ID and password to log into a remote DB2 server. The single sign-on capability provided by Kerberos authentication requires that both DB2 Connect and the database server that it is connecting to provide Kerberos support.

Note: There is no support for the GSSPLUGIN authentication type.

Kerberos support

Kerberos is a third-party network authentication protocol that employs a system of shared secret keys to securely authenticate a user in an unsecured network environment. Ensure that you have the minimum requirements to use Kerberos support on your database.

The Kerberos authentication layer which handles the ticketing system is integrated into the Windows 2000 Active Directory mechanism. The client and server sides of an application communicate with the Kerberos SSP (Security Support Provider) client and server modules. The Security Support Provider Interface (SSPI) provides a high level interface to the Kerberos SSP and other security protocols.

Typical setup

To configure DB2 database products with Kerberos authentication, set up:

- An authorization policy for DB2 (as a service) in the Active Directory that is shared on a network, and
- A trust relationship between Kerberos Key Distribution Centers (KDCs)

In the simplest scenario, there is at least one KDC trust relationship to configure, that is, the one between the KDC controlling the client workstation, and the IBM Power Systems, or System z. OS/390 Version 2 Release 10 or z/OS Version 1 Release 2 provides Kerberos ticket processing through its RACF® facility which allows the host to act as an UNIX KDC.

DB2 Connect provides as usual the router functionality in the 3-tier setting. It does not assume any role in authentication when Kerberos security is used. Instead, it merely passes the client's security token to IBM DB2 for IBM i or to DB2 for z/OS. There is no need for the DB2 Connect gateway to be a member of the client or the host's Kerberos realm.

Downlevel compatibility

Minimum requirements for Kerberos support in DB2 database products:

IBM data server client:

Version 8

DB2 Connect:
Version 8

DB2 for z/OS:
Version 7

Authentication types supported with DB2 Connect server

Certain combinations of authentication and security settings are supported with DB2 Connect.

Authentication types for TCP/IP connections

The TCP/IP communication protocol does not support Authentication options at the network protocol layer. The authentication type determines where authentication takes place. Only the combinations shown in this table are supported by DB2 Connect. The authentication setting is in the database directory entry at the DB2 Connect server.

Table 23. Valid Authentication Scenarios

Scenario	Authentication setting	Validation
1	CLIENT	Client
2	SERVER	IBM mainframe database server
3	SERVER_ENCRYPT	IBM mainframe database server
4	KERBEROS	Kerberos security
5	DATA_ENCRYPT	Host
6	SERVER_ENCRYPT_AES	Host database server

Discussion of Authentication types

The following discussion applies to the connections described previously and listed in Table 23. Each scenario is described in more detail, as follows:

- In scenario 1, the user name and password are validated only at the remote client. For a local client, the user name and password are validated only at the DB2 Connect server.
The user is expected to be authenticated at the location they sign on to. The user ID is sent across the network, but not the password. Use this type of security only if all client workstations have adequate security facilities that can be trusted.
- In scenario 2, the user name and password are validated at the IBM mainframe database server only. The user ID and password is sent across the network from the remote client to the DB2 Connect server and from the DB2 Connect server to the IBM mainframe database server.
- Scenario 3 is the same as scenario 2, except that the user ID and password are encrypted.
- In scenario 4, a Kerberos ticket is obtained by the client from the Kerberos KDC. The ticket is passed unaltered through DB2 Connect to the server, where it is validated by the server.
- Scenario 5 is the same as scenario 3, except that the user data is also encrypted and DATA_ENCRYPT does not support the IBM Power Systems database server.
- Scenario 6 is the same as scenario 3, except that an Advanced Encryption Standard (AES) encryption algorithm is used.

Chapter 9. Tuning

DB2 Connect performance considerations

Performance is the way a computer system behaves given a particular workload. It is affected by the available resources and how they are used and shared. If you want to improve performance, you must first decide what you mean by performance.

You can choose many different *performance metrics*, including:

Response time

The interval between the time that the application sends the database request and the time that the application receives a response.

Transaction throughput

The number of units of work that can be completed per unit of time. The unit of work could be simple, like fetching and updating a row, or complicated, involving hundreds of SQL statements.

Data transfer rate

The number of bytes of data transferred between the DB2 Connect application and the IBM mainframe database per unit of time.

Performance will be limited by the available hardware and software resources. CPU, memory, and network adapters are examples of hardware resources. Communication subsystems, paging subsystems, mbuf for AIX, is an example of a software resource.

Data Flows

Figure 10 on page 128 shows the path of data flowing between the IBM mainframe database server and the workstation through DB2 Connect.

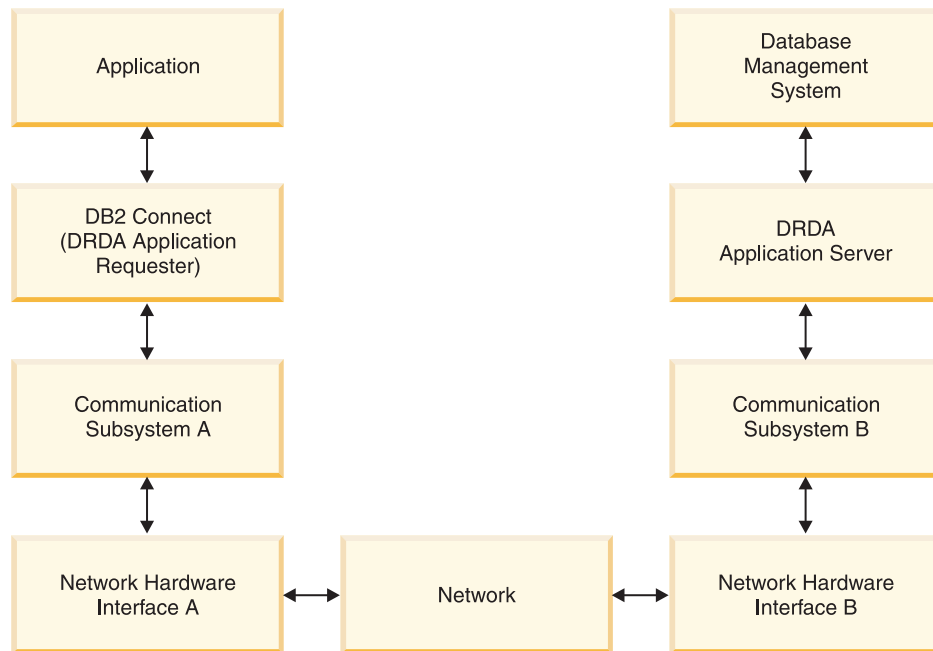


Figure 10. Data Flows in DB2 Connect

- The IBM mainframe database and part of communication subsystem B are usually running on the same system. This system is made up of one or more CPUs, main storage, an I/O subsystem, DASD, and an operating system. Because other programs might share these components, resource contention could cause performance problems.
- The network is composed of a combination of cables, hubs, communication lines, switches, and other communication controllers. For example, the network hardware interface B could be communication controllers such as 3745 or 3172 or a token ring adapter for an IBM Power Systems server. There could be more than one transmission medium involved between network hardware interfaces A and B.
- Network hardware interface A could be token ring, Ethernet**, other LAN adapter, or an adapter which supports the SDLC or X.25 protocols.
- DB2 Connect and the communication subsystem A are usually located on the same system. For the scope of this discussion, it is assumed that the application is also on the same system.

Bottlenecks

Transaction throughput is dependent on the slowest component in the system. If you identify a performance bottleneck, you can often alleviate the problem by changing configuration parameters, allocating more resources to the problem component, upgrading the component, or adding a new component to offload some of the work.

You can use various tools to determine how much time a query spends in each component. This will give you an idea of which components should be tuned or upgraded to improve performance. For example, if you determine that a query spends 60% of its time in the DB2 Connect machine, you might want to tune DB2 Connect or (if you have remote clients) add another DB2 Connect machine to the network.

Benchmarking

Benchmarking compares performance in one environment with performance in another. Benchmarking can begin by running the test application in a normal environment. As a performance problem is narrowed down, specialized test cases can be developed to limit the scope of the function that is tested and observed.

Benchmarking does not need to be complex. Specialized test cases need not emulate an entire application in order to obtain valuable information. Start with simple measurements and increase the complexity only when warranted.

Characteristics of good benchmarks:

- Each test is repeatable.
- Each iteration of a test is started in the same system state.
- The hardware and software used for benchmarking matches your production environment.
- There are no functions or applications active in the system other than those being measured unless the scenario includes some other activity going on in the system.

Note: Applications that are started use memory even when they are minimized or idle. This could cause paging and skew the results of the benchmark.

Performance Tools

The following tables list some of the tools that can help you measure system performance. Because these tools themselves use system resources, you might not want to have them active all the time.

Table 24. Performance tools for CPU and memory usage

System	Tool	Description
AIX	vmstat, time, ps, tprof	Provide information about CPU or memory contention problems on the DB2 Connect workstation and remote clients.
HP-UX	vmstat, time, ps, monitor and glance if available	
Windows	Microsoft Performance Monitor	

Table 25. Performance tools for database activity

System	Tool	Description
All	Database monitor	Determines if the problem originates from the database.
System z	IBM Tivoli OMEGAMON [®] XE for DB2 Performance Monitor on z/OS, ASG-TMON for DB2 (ASG), and CA Insight Performance Monitor for DB2 for z/OS (Computer Associates International, Inc.)	

Table 25. Performance tools for database activity (continued)

System	Tool	Description
Windows	Microsoft Performance Monitor	

Table 26. Performance tools for network activity

System	Tool	Description
AIX	netpmn	Reports low level network statistics, including TCP/IP statistics such as the number of packet or frames received per second.
Network controller such as 3745	NetView® Performance Monitor	Reports utilization of communication control and VTAM®.
Linux and UNIX	netstat	Handles TCP/IP traffic.

Application design

When you create an application, you can improve performance in several ways. For example, consider using compound SQL and stored procedures, grouping related database requests into one database request, refining the predicate logic, implementing data blocking and tuning your dynamic SQL. This section is also relevant for applications using embedded SQL.

Compound SQL and stored procedures

For applications that send and receive many commands and replies, network processing usage can be significant. Compound SQL and stored procedures are two ways to reduce this processing usage.

If an application sends several SQL statements without intervening programming logic, you can use compound SQL. If you require programming logic within the group of SQL statements, you can use stored procedures.

All executable statements except the following statements can be contained within a Compound SQL statement:

- CALL
- FETCH
- CLOSE
- OPEN
- Compound SQL
- Connect
- Prepare
- Release
- Describe
- Rollback
- Disconnect
- Set connection
- execute immediate

Stored procedures help to reduce network traffic by placing program logic at the server. You can commit automatically when exiting the procedure. You can also return results sets, which minimize application logic at the client.

Grouping requests

Grouping related database requests (SQL statements) into one database request can reduce the number of requests and responses transmitted across the network.

For example, grouping the following statements:

```
SELECT COL1, COL2, COL5, COL6 FROM TABLEA WHERE ROW_ID=1  
SELECT COL1, COL2, COL5, COL6 FROM TABLEA WHERE ROW_ID=2
```

into

```
SELECT COL1, COL2, COL5, COL6 FROM TABLEA WHERE ROW_ID=1 OR ROW_ID=2
```

sends fewer requests across the network.

You can also use keywords such as IN and BETWEEN to reduce the number of rows returned. In addition, you can use WHERE, IN, and BETWEEN keywords on UPDATE and DELETE statements.

Predicate logic

You can use predicate logic to request only the rows and columns that are needed. This minimizes the network traffic and CPU usage for data transmission.

For example, do not use the query:

```
SELECT * FROM TABLEA
```

if only the first row of TABLEA with ROW_ID=1 is really needed or if only column 1 and column 2 are needed.

Data blocking

You should use data blocking if you expect large amounts of data from the server. Blocking improves the use of the network bandwidth and reduces the CPU usage of both the IBM mainframe database server and the DB2 Connect server. There is fixed amount of CPU and network usage for each message sent and received regardless of size. Data blocking reduces the number of messages required for the same amount of data transfer.

With blocking, the first row of data from a query will not be delivered to the application until the first block is received. Blocking increases the retrieval time for the first row, but improves the retrieval time for subsequent rows.

Another consideration is the amount of memory that is used. The memory working set usually increases when blocking is turned on.

Within DB2 Connect, you can control the amount of data that is transferred within each block.

To invoke blocking, use the **BLOCKING** option of the **prep** or **bind** command. Blocking is on, if:

- The cursor is read-only, or
- The cursor is ambiguous and blocking is specified during the **prep** or **bind**.

Note: When using dynamic SQL, the cursor is always ambiguous.

SQL statements with BLOCKING

Updatable SELECT statements (using UPDATE/DELETE WHERE CURRENT OF statements) are non-blocking queries, so you should use them only when absolutely necessary.

An updatable SELECT ensures that the row has not changed between the time the SELECT is completed and the UPDATE/DELETE is issued. If this level of concurrency is not important to your application, an alternative is to use a DELETE or UPDATE with search criteria based on the values returned from a non-updateable SELECT.

For read-only SELECT, specify FOR FETCH ONLY, except under VM and VSE, where it is not supported.

Static and dynamic SQL

Use static SQL as much as possible. It avoids runtime SQL section preparation and ambiguous cursors. If dynamic SQL cannot be avoided, you can do the following to minimize the network traffic and improve performance:

- If the statement is a SELECT and must be prepared, perform PREPARE ... INTO SQLDA. The SQLDA should be allocated to the full size needed for your settings. If the maximum number of columns is x and is expected to stay that way, allocate an SQLDA with x SQLVARs. If the number of potential columns is uncertain (and memory is not a problem), use the maximum number of SQLVARs (256).

If the SQLDA allocation is not big enough to store the returning SQLDA, the program must issue another DESCRIBE with a big enough SQLDA to store the result again. This would increase the network traffic.

Do not use the PREPARE and DESCRIBE sequence. Using the PREPARE ... INTO statement provides better performance.

- Execute statically bound SQL COMMIT or ROLLBACK statements instead of dynamic COMMIT or ROLLBACK statements.
- If it is not a SELECT, COMMIT, or ROLLBACK statement, issue EXECUTE IMMEDIATE to execute the statement instead of the PREPARE and EXECUTE sequence.
- ODBC applications use dynamic SQL. You might use the CLI/ODBC static profiling feature to improve performance. This feature allows you to capture and convert ODBC calls into static statements stored in a database package. The actual performance you will get depends on the complexity of your application.

Other SQL considerations

Using the Command Line Processor (CLP) is, in general, slower than having dynamic SQL in the program because the CLP must parse the input before submitting the SQL to the database engine. The CLP also formats data when it is received, which might not be necessary for your application.

SQL statements in an interpreted language, such as REXX, are substantially slower than the same SQL statements in a compiled language, such as C.

There are two types of CONNECT statement, called type 1 and type 2. With type 2 connect, connecting to a database puts the previous connection into a dormant state but does not drop it. If you later switch to a dormant connection, you avoid the processing usage of loading libraries and setting up internal data structures. For this reason, using type 2 connect might improve performance for applications that access more than one database.

Connection management

Connection pooling

DB2 Connect server products, such as DB2 Connect Enterprise Edition, often provide database connections for thousands of simultaneous client requests.

Establishing and severing connections to the database server can be a very resource intensive process that adversely affects both database server and DB2 Connect server performance. To reduce this processing usage, DB2 Connect server products use connection pooling to maintain open connections to the database in a readily accessible pool.

This problem is especially evident in web environments where each visit to a web page can require building a new connection to the database server, performing a query and terminating a connection. Most applications based on web technologies execute large volume of short transactions. A typical web transaction is executed as part of its own connection. In other words, executing a transaction means establishing a database connection and then terminating this connection only after a few SQL statements. This process of establishing and tearing down a connection is very costly. It involves creation of a DB2 Connect agent, establishing a network connection between this agent and the DB2 server, and creation of a DB2 thread on the server. For longer running connections these costs are amortized over all of the transactions executed on this connection but for a typical web transaction these costs will typically exceed the cost of executing the transaction itself.

Connection pooling is a technique that allows reuse of an established connection infrastructure for subsequent connections. When a DB2 Connect instance is started a pool of coordinating agents is created. When a connection request comes in an agent is assigned to this request. The agent will connect to the DB2 server and a thread will be created in DB2. When the application issues a disconnect request, the agent will not pass this request along to the DB2 server. Instead, the agent is put back in to the pool. The agent in the pool still owns its connection to the DB2 server and the corresponding DB2 thread. When another application issues a connect request, this agent is assigned to this new application. To insure secure operation, user identity information is passed along to the DB2 thread which, in turn, performs user authentication.

DB2 Connect's connection pooling provides a significant performance improvement in such environments. DB2 Connect maintains open connections to the database in an available pool. When a client requests a connection, it can be provided from this pool of ready connections. Connection pooling significantly reduces the processing usage typically spent on opening and closing these connections.

Connection pooling is transparent to applications connecting to the host through DB2 Connect. When an application requests disconnection from the host, DB2 Connect drops the inbound connection with the application, but keeps the outbound connection to the host in a pool. When a new application requests a connection, the DB2 Connect uses one from the existing pool. Using the already-present connection reduces the overall connection time, as well as the high CPU connect cost on the host.

DB2 Connect agents can be in one two states: idle or active. An agent is active when it is executing work for an application. Once this work is completed the agent goes into an idle state awaiting further work from the same or a different application. All idle agents are kept together in what is known as the idle agent

pool. You can configure the size of this pool using the **num_poolagents** configuration parameter. This parameter equals the maximum number of idle agents you want the system to maintain. Setting this parameter to zero is equivalent to turning off the connection pooling feature. The default for this configuration parameter is set to AUTOMATIC with a value of 100. By being set to AUTOMATIC, DB2 Connect automatically manages the number of idle agents in the idle agent pool.

DB2 Connect does not establish connections to the database before receiving its first client request. Alternatively, you can fill the pool of idle agents before any clients make a request. The pool can be filled on startup using the **num_initagents** configuration parameter. This parameter determines how many idle agents should be created at start up time. These idle agents initially will not have connections to the host database server.

When a client requests a connection to the host, DB2 Connect will attempt to get an agent from among those in the pool that have a connection to the host database server. If that fails, it will try to find an available agent in the idle pool. If the pool is empty, DB2 Connect will create a new agent.

You can control the maximum number of agents that can be concurrently active using the **max_coordagents** configuration parameter. Once this number is exceeded, new connections will fail with error sqlcode SQL1226. (This code means that the maximum number of concurrent outbound connections has been exceeded.) The default for this configuration parameter is set to AUTOMATIC with a value of 200. By being set to AUTOMATIC, DB2 Connect automatically manages the number of coordinator agents.

The DB2 registry variable **DB2CONNECT_IN_APP_PROCESS** allows applications running on the same machine as a DB2 Connect server product to either have DB2 Connect run within the applications process, default behavior, or to have the application connect to the DB2 Connect server product and then have the host connection run within an agent. For an application to use connection pooling the connections to the host must be made from within the DB2 Connect server product agents and thus **DB2CONNECT_IN_APP_PROCESS** must be set to NO.

DB2 Connect Connection Pooling versus Application Server Connection Pooling

Connection pooling is a must for any web technologies based application that is to support large volumes of transactions. Most web application servers now provide their own way of pooling database connections. For example, both Microsoft MTS (COM+) and IBM WebSphere provide connection pooling.

Application pooling mechanisms implemented by these servers differ significantly from what is provided by the DB2 Connect servers. Since application servers pool connections only for their own use they typically presume that user id, password, isolation levels, and so on, will be exactly the same for all connections. Even more important, application servers only pool connections initiated by the same process. This means that connections from other machines, users or processes are not pooled. While these application server pooling techniques are effective for reusing connections established by the same instance of an application they are absolutely ineffective for pooling connections from multiple users, servers, and so on.

Connection pooling, provided by the DB2 Connect servers, is completely application, machine and user independent. Connections from multiple clients,

application servers all with different user IDs can all reuse each other's connections resulting in a much better utilization of the pooled resources.

Which type of connection pooling is the right one to use? Both. Generally, using both DB2 Connect connection pooling and Application Server connection pooling is a good strategy since they don't interfere with each other. Even when application server connection pooling is enabled, DB2 Connect connection pooling can provide connection reuse for multiple application servers as well as other clients using the DB2 Connect server.

Connection concentrator

The *connection concentrator* reduces the resources required on DB2 for z/OS database servers to support large numbers of workstation and web users. This function can dramatically increase the scalability of your DB2 for z/OS and DB2 Connect solution while also providing for fail-safe operation and transaction level load balancing in DB2 for z/OS data sharing environments.

The connection concentrator allows applications to stay connected without any resources being consumed on the DB2 host server. You can have thousands of users active in applications and only have a few threads active on the DB2 host server.

DB2 Connect's connection concentrator technology allows DB2 Connect server products, such as DB2 Connect Enterprise Edition, to provide support to thousands of users simultaneously executing business transactions, while drastically reducing resources required on the System z host or IBM Power Systems database servers. It accomplishes this goal by concentrating the workload from all applications in a much smaller number of System z host or IBM Power Systems database server connections. While this might seem similar to the connection pooling function described previously, it is in fact a more sophisticated approach to reducing resource consumption for very high volume OLTP (On-line Transaction Processing) applications.

Connection concentrator takes the concept of an agent and splits it into two entities:

- *Logical agent*, which represents an application connection.
- *Coordinating agent*, which owns the DB2 connection and thread, and executes application requests.

When a new application attempts a connection to the host, it is assigned a logical agent. To pass SQL to the database, a coordinating agent is required and is assigned as soon as a new transaction is initiated. The key to this architecture is the fact that the coordinating agent is:

- Disassociated from the logical agent
- Returned to the pool when transaction completes due to a commit or rollback

Another key feature is the method of assigning coordinating agents to new transactions in a DB2 pureScale environment. DB2 Connect implements a sophisticated scheduling algorithm that uses System z Work Load Manager (WLM) information. This information is used to distribute workload across members of a data sharing group according to criteria set up in WLM. WLM is not only aware of the load on each member but also their availability. This allows DB2 Connect to transparently relocate work away from failed or overloaded members to members that are up and underutilized. DB2 Connect connection concentrator is activated

when you set the number of maximum logical agents (**max_connections**) higher than the number of coordinating agents (**max_coordagents**).

Connection pooling saves the cost of establishing a connection when one is no longer needed by a terminating application. In other words, one application has to disconnect before another one can reuse a pooled connection.

Alternatively the connection concentrator allows DB2 Connect to make a connection available to an application as soon as another application has finished a transaction and does not require that other application to disconnect. In essence, a database server connection and its associated host and DB2 Connect resources are used by an application only while it has an active transaction. As soon as the transaction completes, the connection and associated resources are available for use by any other application that needs to have a transaction executed.

In previous versions of DB2 Connect, every active application had an Engine Dispatchable Unit (EDU) which managed the database connection as well as any application requests. This EDU was typically referred to as the *coordinator agent*. Each coordinator agent tracked the state, or context of the application and EDU. Each EDU takes a significant amount of memory when the number of connections increases, and context switching between agents results in additional processing usage.

In the architecture mentioned previously, there is a one-to-one relationship between connections and EDUs. The connection concentrator, however, permits a many-to-one relationship between connections and EDUs. That is, the relationship of connections (X) to EDUs (Y) is now $X \geq Y$.

The connection concentrator splits the agent into two entities, a *logical agent* and a *worker agent*. Logical agents represent an application, but without reference to a particular EDU. The logical agent contains all the information and control blocks required by an application. If there are n applications connected to the server, there will be n logical agents on the server. Worker agents are physical EDUs that execute application requests, but which have no permanent attachment to any given application. Worker agents associate with logical agents to perform transactions, and at the transaction boundary end the association and return to the available pool.

An entity known as the *dispatcher* assigns worker agents to logical agents. Limitations in the number of open file handles on certain computing platforms might result in more than one scheduler instance.

Restrictions for the connection concentrator

There are a number of important restrictions to the use of the DB2 Connect server concentrator. Review the following information in its entirety before attempting to use the connection concentrator on your system.

General restrictions:

- The concentrator relies on the TCP/IP protocol to establish inbound connections from local and remote clients. Only inbound connections using TCP/IP or Local (IPC) will be able to take advantage of pooled outbound connections. The concentrator will accept connections via other communications protocols such as named pipes, but you will not be able to use its XA concentration features with that connection.

- For XA tightly coupled transaction support, all applications that participate in the same XA transaction must use the same DB2 Connect server Instance to connect to the host.
- Only applications that close withhold resources (such as withhold cursors) on transaction boundaries can benefit from the concentrator. Transactions that do not close withhold cursors will still go through, but will be assigned a dedicated worker agent and hence will not be able to use the concentrator's full feature set.
- If you declare temporary tables, they must be dropped explicitly at transaction or branch boundary. Failure to drop the tables will turn off connection concentration but the application will continue to work.
- All applications participating in the same XA transaction must have the same CCSID and use the same user ID to make the connection.
- If an outbound connection was established to support two-phase connection, that connection's agent can only be used to support two-phase connections. Similarly, agents established to support a one-phase connection can only support one-phase connections.
- The concentrator supports applications that use the IBM Data Server Driver for JDBC and SQLJ and also Call Level Interface (CLI) applications that use dynamic SQL. CLI applications should also not use KEEP_DYNAMIC as the concentrator depends on statements being re-prepared on each transaction boundary.
- Dynamic prepare requests from embedded dynamic SQL applications will be rejected. Your applications should be altered so as to either use static SQL or to use the CLI for dynamic SQL statements.
- If the connection concentrator is ON, the inbound request to the DB2 Connect server cannot use SSL. However, the outbound request to the target database server can use SSL. If the connection concentrator is OFF, both the inbound and the outbound requests can use SSL.

When working with DB2 Version 9 or Version 8 FixPak 13 (or higher), to enable DB2 Connect concentrator support requires IBM Power Systems Version 5 Release 4 (PTF SI23726). Otherwise, only the XA portion of the connection concentrator is supported.

Activating the connection concentrator

The database manager configuration parameter **max_coordagents** sets the maximum number of logical agents. You can activate the concentrator feature by setting the value of **max_connections** to any number greater than the default. The default value for **max_connections** is equivalent to the value of **max_coordagents**. Because each application will have one logical agent, **max_connections** actually controls the number of applications that can be connected to the database instance, while **max_coordagents** controls the number of inbound connections that can be active at any time. **max_connections** will take a numeric range from **max_coordagents** up to 64 000. The default number of logical agents is equal to **max_coordagents**.

Both **max_connections** and **max_coordagents** can be set to AUTOMATIC. If **max_connections** is set to AUTOMATIC, the number of connections can be increased beyond the base configured value. If both **max_connections** and **max_coordagents** are set to AUTOMATIC, **max_connections** can be increased beyond the base value, and **max_coordagents** is automatically increased to maintain the concentration ratio between connections and the coordinator agents.

Several existing configuration parameters are used to configure agents. These parameters are as follows:

max_coordagents

Maximum number of active coordinator agents.

num_poolagents

Agents pool size. The agent pool includes inactive agents and idle agents. For improved performance, **num_poolagents** should be configured to equal the average number of clients.

num_initagents

Initial number of worker agents in the pool. These will be idle agents.

XA transaction support

The architecture of the connection concentrator allows DB2 Connect to provide tightly coupled XA transaction support to DB2 for z/OS and IBM DB2 for IBM i. The concentrator will associate a worker agent with a particular XA transaction (single XID) as it would for any other transaction. However, if the XA transaction is ended by `xa_end()` (branch boundary), the worker agent will not release itself into the general pool. Instead, the worker remains associated with that particular XA transaction. When another application joins the same XA transaction, the worker agent will be attached to that application.

Any transaction boundary call will return the agent to the pool. For instance, `xa_prepare()` with read only, `xa_rollback()`, `xa_recover()`, `xa_forget()`, `xa_commit()`, or any XA error that causes rollback will return the agent to the normal pool. `xa_end()` itself only ends the transaction branch, and this is not sufficient to end its association with the XID.

Examples of XA transaction support

1. Consider an environment where 4 000 or more concurrent connections are needed. A web server that uses CGI applications, or an office system with many desktop users can both exceed this requirement. In these cases, efficiency will usually require that DB2 Connect operate as a stand-alone gateway; that is, the database and the DB2 Connect system are on separate machines.

The DB2 Connect server system might not be able to maintain 4 000 simultaneous open connections to the database machine. In most cases, the number of transactions occurring at any given moment will be considerably less than the number of concurrent connections. The system administrator could then maximize the efficiency of the system by setting the database configuration parameters as follows:

```
MAX_CONNECTIONS = 4,000
MAX_COORDAGENTS = 1,000
NUM_POOLAGENTS  = 1,000
```

The concentrator will keep open up to 4 000 concurrent sessions, even though the gateway is only managing 1 000 transactions at a time.

2. In the previous example, worker agents will constantly form and break associations to logical agents. Those agents that are not idle might maintain a connection to the database but are not participating in any particular transaction, hence they are available to any logical agent (application) that requests a connection.

The case of XA transactions is somewhat different. For this example, assume that a TP Monitor is being used with a DB2 Connect gateway and an System z or IBM Power Systems database. When an application requests a connection,

the concentrator will either turn an inactive agent over to serve that request, or create a new worker agent. Assume that the application requests an XA transaction. An XID is created for this transaction and the worker agent is associated with it.

When the application's request has been serviced, it issues an `xa_end()` and detaches from the worker agent. The worker agent remains associated with the XID of the transaction. It can now only service requests for transactions with its associated XID.

At this time, another application might make a request for a non-XA transaction. Even if there are no other available worker agents, the agent associated with the XID will not be made available to the second application. It is considered active. The second application will have a new worker agent created for it. When that second application completes its transaction, its worker agent is released into the available pool.

Meanwhile, other applications requesting the transaction associated with the first agent's XID can attach and detach from that agent, which executes its dedicated XA transaction for them. Any application requesting that particular transaction will be sent to this worker agent if it is free.

The worker agent will not be released back into the general pool until an application issues a transaction boundary call (not `xa_end()`). For instance, an application might end the transaction with `xa_commit()`, at which point the worker agent drops its association with the XID and returns to the available pool. At this point any requesting application can use it for either another XA, or a non-XA, transaction.

Connection pooling and connection concentrator

While connection pooling and connection concentrator seem to have similarities, they differ in their implementation and address different issues. Connection pooling helps reduce the processing usage of database connections and handle connection volume. Connection concentrator helps increase the scalability of your DB2 for z/OS and DB2 Connect solution by optimizing the use of your host database servers.

When using connection pooling, the connection is only available for reuse after the application owning the connection issues a disconnect request. In many 2-tier client-server applications users do not disconnect for the duration of the workday. Likewise, most application servers in multi-tier applications establish database connections at server start up time and do not release these connections until the application server is shut down.

In these environments, connection pooling will have little, if any, benefit. However, in web and client-server environments where the frequency of connections and disconnections is higher than connection pooling will produce significant performance benefits. The connection concentrator allocates host database resources only for the duration of an SQL transaction while keeping user applications active. This allows for configurations where the number of DB2 threads and the resources they consume can be much smaller than if every application connection had its own thread.

When it comes to fail-safe operation and load balancing of workload connection concentrator is clearly the right choice as it allows reallocation of work with every new transaction. Alternatively, connection pooling can only offer very limited balancing and only at connect time.

Connection pooling and connection concentrator should be used together although they address different issues.

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

When running applications in an IBM WebSphere MQ (formerly known as IBM MQSeries®) environment, WebSphere MQ can act as an XA-compliant transaction manager, coordinating any distributed, two-phase commit transactions. When WebSphere MQ is acting as a transaction manager in this way, and the data sources are from the DB2 family of products, there are several configuration requirements.

Most of the configuration requirements in such a transaction manager environment are already documented elsewhere. For example, you must set the DB2 configuration parameter **tp_mon_name** to MQ at the DB2 runtime client.

However, there is a configuration requirement that was missing. The requirement is specific to DB2 Connect when connecting to data sources that are DB2 for z/OS servers: when using WebSphere MQ to coordinate distributed transactions involving DB2 for z/OS and IBM DB2 for IBM i servers, the DB2 Connect connection concentrator feature must be enabled at the gateway. The connection concentrator is enabled when the value of the **max_connections** configuration parameter is greater than the value of the **max_coordagents** configuration parameter.

If you do not enable the connection concentrator, unexpected transaction behavior will result.

If you are using the WebSphere MQ Transaction Manager and DB2 for z/OS server, the application must set the special registers for each local or global transaction.

DB2 Connect server tuning

Various parameters in the database manager configuration file can be used to tune DB2 Connect.

RQRIOBLK

The **RQRIOBLK** parameter sets the maximum size of network I/O blocks. A larger block size might improve the performance of large requests. The block size does not usually affect the response time for small requests, such as a request for a single row of data.

A larger block size usually requires more memory on the DB2 Connect server. This increases the size of the working set and might cause large amounts of paging on small workstations.

Use the default DRDA block size (32767) if it does not cause too much paging on executing your application. Otherwise, reduce the I/O block size until there is no paging. Once paging begins, a noticeable degradation of performance will occur. Use performance monitoring tools (such as the **vmstat** tool for Linux and UNIX operating systems) to determine whether paging is occurring on your system.

DIR_CACHE

The **DIR_CACHE** parameter determines whether directory information is cached. With caching (**DIR_CACHE=YES**), directory files are read and cached in memory to minimize the processing usage of creating the internal directory structure and reading the directory files every time a connection is established.

Without caching (**DIR_CACHE=NO**), whenever you connect to a database the appropriate directory is read from a disk and then the search is performed. After the requested entries are found, all memory related to directory searches is freed.

With caching, a shared directory cache is built during **db2start** processing and freed when DB2 stops. This cache is used by all DB2 server processes (**db2agent**). Also, a private application directory cache is built when an application issues its first connect to a database and freed when the application ends.

Each cache provides an image of the system database directory, the database connection services directory and the node directory. The cache reduces connect costs by eliminating directory file I/O and minimizing directory searches.

If a cached directory is updated, the changes are not immediately propagated to the caches. If a directory entry is not found in a cache, the original directory is searched.

Caching increases the private memory that is needed for the life of an application. Without caching, this memory is needed only when a directory lookup is processed. Overall use of shared memory by DB2 increases slightly because directory information that is shared among database agents is moved to shared memory. The size of the memory required for a cache depends on the number of entries defined in each directory.

NUMDB

The behavior of DB2 Connect was unaffected by the **NUMDB** configuration parameter in previous versions, however, this changed starting with Version 8. This parameter indicates the maximum number of databases the clients can connect to through the DB2 Connect server. More specifically, the maximum number of different database aliases that can be catalogued on DB2 Connect server.

Other DB2 Connect parameters

The **AGENTPRI** and **MAXAGENTS** are deprecated in Version 9.5

Commands to update the value for **MAXAGENTS** will continue to work so that existing applications are not broken, but the values will be ignored. The parameter name will not appear in any configuration lists. In the past, the total number of agents allowed to be created on a given DB2 partition was controlled through the **MAXAGENTS** configuration parameter. You now have the ability to automate the configuration of agents.

By default, **NUM_POOLAGENTS** will be set to **AUTOMATIC** with a value of 100 as the default. Also by default, **MAX_COORDAGENTS** will be set to **AUTOMATIC** with a value of 200 as the default.

To send accounting strings from your client applications to the DB2 Connect server, use the API-specific means for setting accounting information. The API-specific means perform faster than setting the **DB2ACCOUNT** environment variable.

IBM Data Server Driver for JDBC and SQLJ

com.ibm.db2.jcc.DB2BaseDataSource.clientAccountingInformation property

IBM Data Server Provider for .NET

DB2Connection.ClientAccountingInformation property

CLI/ODBC

ClientAcctStr CLI/ODBC configuration keyword

Embedded SQL (C, C++, and COBOL)

sqlsact function

If you do not need a tailored SQLCODE mapping file, you can improve performance by using the default SQLCODE mapping or turning off SQLCODE mapping. The default mapping file is imbedded in the DB2 Connect library; a tailored mapping file must be read from disk, which affects performance.

Host database tuning

System performance will be affected by the performance of the IBM mainframe database server. Different database management systems have different performance features. SQL optimizers of different systems, for example, could behave differently with the same application.

Check your IBM mainframe database server system performance documentation for more information.

You might be able to improve performance by using the uncommitted read (UR) or no commit (NC) bind options, where available, to avoid journaling.

Note: When using UR, unjournaled data can only be read, not updated, and then only if blocking is set to ALL.

Depending on the application server and the lock granularity it provides, the isolation level used for a query or application might have a significant effect on performance. The database should have the appropriate level of normalization, effective use of indexes, and suitable allocation of database space. Performance can also be affected by the data types that you use, as described in the following sections.

Network tuning considerations

The best way to improve overall performance in a distributed database environment is to eliminate delays from the network.

It is common for network administrators to consider a network to be more efficient if it collects as much data as possible between transmissions. This approach doesn't work for applications such as distributed databases because it builds delays into the network. The end-user doesn't see the efficiency of the network, only the delays.

Most network devices have delay parameters, and most of them default to values that are very bad for distributed databases. To improve performance you should locate these parameters and if possible, set them to zero. In addition you should ensure that the buffer size on the device is large enough to prevent retransmits due

to lost data. For instance, UNIX systems typically have a Transmit or Receive queue depth default of 32. For better results, set the queue depth to 150. A corresponding parameter on DLC settings is the Receive Depth, which should also be 150.

The IOBUF parameter is set too low at most sites. It is usually set at 500, but experience has shown that a value of 3992 works best if you are moving large amounts of data, especially for channel connections such as ESCON® or 3172.

On a LAN system the DLC or LLC transmit and receive window sizes can have a dramatic effect on performance. The send value should be set to seven or more, and for most configurations a receive value of four or less works best.

If you are running Ethernet, you should set the TCP segment size to 1500 bytes. On a token ring or FDDI network this value should be 4400 bytes, and if you are using an ESCON adapter with TCP/IP, the segment size should always be 4096.

Finally, for TCP/IP networks, the TCP Send and Receive buffer sizes should be set higher than 32768. A value of 65536 is generally best.

Note: Establishing a connection from the gateway to the server (outbound connection) is much more expensive than establishing a connection from a client to the gateway (inbound connection). In an environment where thousands of clients frequently connect to and disconnect from the server through the gateway, a substantial amount of processing time is spent establishing outbound connections. DB2 Connect provides connection pooling over TCP/IP. When a client requests disconnection from the server, the gateway drops the inbound connection with the client, but keeps the outbound connection to the server in a pool. When a new client comes into the gateway to request a connection, the gateway provides an existing one from the pool thus reducing the overall connection time and saving the high CPU connect cost on the server.

A summary of network performance tuning methods is provided in Table 27.

Table 27. Network performance tuning methods

What to Look For	Example	Setting	Notes
Deliberate Delays	Delay parameters on network devices	Set to 0.	Defaults are usually higher.
Buffers	IOBUF parameter	Set up to 3992.	Particularly useful for ESCON or other channel adapter.
Buffers	RUSIZE	Optimum size is 4096.	Setting RUSIZE and RQRIOBLK to same size might give the best performance.
Buffers	Pacing	VPACING, PACING, and Mode Profiles should be set to 63.	Use adaptive pacing where applicable.
Adapter Settings	Transmit/Receive queue depth	Recommended value is 150.	Default is usually 32.
TCP Settings	Segment Sizes	1500 on Ethernet, 4400 on token ring and FDDI.	ESCON adapters used for TCP/IP should always be set to 4096.

Table 27. Network performance tuning methods (continued)

What to Look For	Example	Setting	Notes
TCP Settings	Send/Receive Space Sizes	Should be 64K for both.	Default is only 8192 for Windows. Can be set in the Windows registry.

System resources contention

Performance could be degraded if many tasks in the system are contending for system resources.

Consider the following questions:

- Is the CPU saturated? Consider upgrading the system, reducing the system workload, and tuning the system to reduce processing usage.
- Is the memory over-committed? Consider upgrading memory, reducing system workload and tuning the system to reduce the memory working set.
- Is the communication adapter/communication controller too busy? Consider upgrading the network or pairing up token-ring cards.
- Is one of the subsystems too busy, and is this subsystem on the data path?
- Are any unnecessary processes or tasks running on the system? The general rule is not to configure or start services unless they are used regularly since they will waste system resources.
- Do a few processes or tasks use most of the resource? Can they be stopped? Can their priorities be reduced? Can they be refined so that they don't use as much resource?

DB2 Connect performance troubleshooting

If DB2 Connect users are experiencing long response times during large queries from IBM mainframe servers, there are some configuration settings that can help you troubleshoot the performance problem.

The following areas should be examined for the possible cause of the performance problem:

1. For queries which result in returning large data blocks from the IBM mainframe server (usually 32K of data and above), ensure that the database manager configuration parameter **RQRIOBLK** is set to 32767. This can be done using the Command Line Processor (CLP) as follows:

```
db2 update database manager configuration using RQRIOBLK 32767
```
2. Ensure the maximum RU size defined in the IBMRDB mode definition is set to a suitable value. It is recommended that the size is not less than 4K for connections using Token-ring hardware. For connections using Ethernet hardware, note the maximum Ethernet frame size of 1536 bytes, which might be a limiting factor.

Tuning DB2 for z/OS

You can optimize inactive thread processing in z/OS.

In V5, you are allowed up to 25,000 concurrently connected clients. In all cases, the maximum number that can be concurrently active, however, is 1999. Each workstation client can stay connected when it is inactive; its thread is placed on an inactive chain at each commit.

The DSNZPARM parameters CMTSTAT, CONDBAT and MAXDBAT affect thread processing. For best performance, set CMTSTAT to INACTIVE, adjust CONDBAT to the maximum number of connected DBATs that provide good performance, and MAXDBAT to the maximum acceptable number of active DBATs.

Increasing DB2 Connect data transfer rates

In addition to blocking of rows for a query result set, DB2 for z/OS can also return multiple such query blocks in response to an OPEN or FETCH request to a remote client, such as DB2 Connect.

Instead of the client repeatedly sending requests to the DB2 for z/OS server requesting one block of row data at a time, the client can now optionally request that the server send back some number of query blocks in addition to the one that it will always send back. Such additional query blocks are called extra query blocks.

As such, this new feature allows the client to minimize the number of network line turnarounds, which constitute a major cost to network performance. The decrease in the number of requests sent by the client to the server for query blocks translates into a significant performance boost. This performance boost is due to the fact that switching between a send and receive is an expensive operation performance-wise. DB2 Connect can now exploit this performance enhancement by requesting extra query blocks from a DB2 for z/OS server by default.

To fully take advantage of the return of extra query blocks (each of which can be up to 32K bytes long) for the preferred network protocol of TCP/IP, window scaling extensions have been enabled as architected under RFC-1323 in DB2 Connect. This feature that allows TCP/IP to dynamically adjust the send and receive window sizes to accommodate the potentially large amounts of data returned by way of the extra query blocks efficiently.

Extra query block

Extra query block support on servers with DB2 for z/OS Version 7 or later is configured via the EXTRA_BLOCKS SRV parameter on the DB2 DDF installation panel. This support is configured by way of controlling the maximum number of extra query blocks that DB2 can send back to a client for a request.

You can set this parameter to a value between 0 and 100. Setting the parameter value to 0 disables the return of extra query blocks. The default value of 100 should always be used to get the most benefit out of this feature, barring any idiosyncrasies in the network that would render this setting less than ideal.

On the client side, where the application accesses DB2 for z/OS either directly through a co-located DB2 Connect installation, or through a separate DB2 Connect server installation, there are various means for activating the corresponding DB2 Connect support on a per cursor or statement basis:

- The use of a query rowset size for a cursor
- The use of the 'OPTIMIZE for N ROWS' clause on the select statement associated with a cursor
- The use of the 'FETCH FIRST N ROWS ONLY' clause on the select statement associated with a cursor

DB2 Connect can enable extra query block support using different SQL APIs:

Embedded SQL

- The user can invoke extra query block support for a query by specifying either the 'OPTIMIZE for N ROWS' clause, or the 'FETCH FIRST N ROWS ONLY' clause, or both on the select statement itself.
- With the 'OPTIMIZE for N ROWS' clause, DB2 for z/OS will attempt to block the desired number of rows to return to DB2 Connect, subject to the EXTRA_BLOCKS SRV DDF installation parameter setting. The application can choose to fetch beyond N rows as DB2 for z/OS does not limit the total number of rows that could ultimately be returned for the query result set to N.
- The 'FETCH FIRST N ROWS ONLY' clause works similarly, except that the query result set is limited to N rows by DB2 for z/OS. Fetching beyond N rows would result in SQL code +100 (end of data).

CLI/ODBC

- The user can invoke extra query block support for a query through its SQL_MAX_ROWS statement attribute.
- The 'FETCH FIRST N ROWS ONLY' clause is used instead for a DB2 for z/OS 7.1 or later server.
 - For Version 7, the query result set is limited to N rows by DB2 for z/OS. Fetching beyond N rows would result in SQL_NO_DATA_FOUND.
 - For Version 8 or later, the CLI ensures that only the first N rows are returned to the application via the client Cursor Manager.

JDBC The user can invoke extra query block support for a query through the setMaxRows method. Similar to the CLI/ODBC enablement, DB2 Connect will tag on the 'OPTIMIZE for N ROWS' clause for a DB2 for z/OS 6.x server. DB2 Connect will also tag the 'FETCH FIRST N ROWS ONLY' clause for a DB2 for z/OS 7.1 or later server.

RFC-1323 Window scaling

Window scaling is supported on all Windows, Linux, and UNIX platforms that support the RFC-1323 extensions for TCP/IP. You can enable this feature on DB2 for Windows, Linux, or UNIX using the DB2 registry variable **DB2SORCVBUF**.

To turn window scaling on, this registry variable should be set to any value above 64K. For example, on DB2 for Windows, Linux, or UNIX, you can issue `db2set DB2SORCVBUF=65537`.

The maximum send and receive buffer sizes are dependent on the specific operating system. To ensure that buffer sizes configured have been accepted, the user can set the database manager configuration parameter **diaglevel** to 4 (informational) and check the administration notification log file for messages.

For window scaling to take effect it must be enabled on both ends of a connection; on both the workstation and the host, either directly through the operating system TCP/IP stack, or indirectly through the DB2 database product. For instance, for DB2 for z/OS, window scaling can currently only be activated through the operating system by setting TCPRCVBUFRSIZE to any value above 64K. If you are using a remote IBM data server client to access an IBM mainframe DB2 database through a DB2 Connect server workstation, you can enable window scaling on the client as well. By the same token, you can also enable window scaling between a remote IBM data server client and a workstation DB2 server when no IBM mainframe DB2 database is involved.

While window scaling is designed to enhance network performance, it is important to note that the expected network performance improvement does not always materialize. Interaction among factors such as the frame size used for the ethernet or token ring LAN adapter, the IP MTU size, and other settings at routers throughout the communication link could even result in performance degradation once window scaling has been enabled. Therefore, by default, window scaling is disabled with both the send and receive buffers set to 64K.

You should be prepared to assess the impact of turning on window scaling and perform any necessary adjustments to the network. For an introduction to tuning the network for improved network performance, refer to www.networking.ibm.com/nhd/webnav.nsf/pages/netdocs.html.

High availability and load balancing for host database connectivity

In today's information technology market, there is a high demand for around the clock availability of data.

This demand must be met in order for a business to compete with its competitors and maintain continued growth. Many of today's web and spreadsheet applications require access to enterprise data.

A reliable, fast and secure connection to IBM mainframe databases must be established. This connection must be constantly available and be able to handle the high connection demands under critical load conditions.

How can this connection be built?

High availability scenario

A company has several workstations and application servers running on Windows, Linux, and UNIX. These machines require access to data residing on several IBM mainframe databases. Applications running on these machines demand fast and reliable connections to the databases. The entire system is connected by an Ethernet network using TCP/IP.

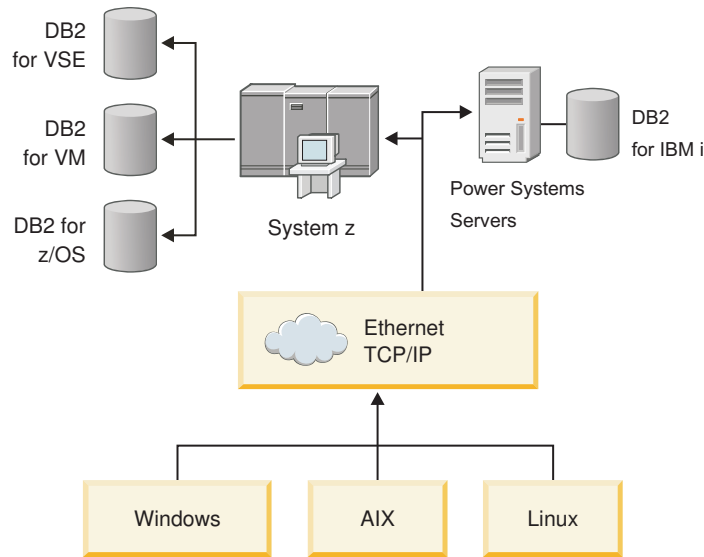


Figure 11. Sample network scenario

For workstations and application servers to access IBM mainframe databases, you need a connectivity component as an intermediary. This component must provide a highly available, robust, and fast connection to IBM mainframe databases. It must also be scalable to anticipate for future growth in connection volume.

Use the related links from this topic to see details regarding a solution using DB2 Connect and the automatic client reroute feature .

Host data conversion

When information is transferred between different environments (such as Intel [Windows], IEEE [Linux and UNIX operating systems], System z [VM, VSE, z/OS], IBM Power Systems [IBM i]), numeric data types (such as decimal, integer, floating point) might need to be converted. This conversion can affect performance.

The CPU cost of single-byte character data conversion is generally less than that of numeric data conversion (where data conversion is required).

The data conversion cost of DATE/TIME/TIMESTAMP is almost the same as that of single-byte CHAR. FLOATING point data conversion costs the most. The application designer might want to take advantage of these facts when designing an application based on DB2 Connect.

If a database table has a column defined 'FOR BIT DATA', the character data being transferred between the application and the database does not require any data conversion. This can be used when you are archiving data on the IBM mainframe database server.

Data types for character data

Character data can have either the CHAR or VARCHAR data type.

Which data type is more efficient depends on the typical length of data in the field:

- If the size of actual data varies significantly, VARCHAR is more efficient because CHAR adds extra blank characters to fill the field. These blank characters must be transmitted across the network like any other characters.

- If the size of actual data does not vary much, CHAR is more efficient because each VARCHAR field has a few bytes of length information which must be transmitted.

Network hardware

The following considerations relate to the hardware: speed of the network or transmission media; network adapter or communication controller; network topology; network traffic; and network reliability.

- Speed of the network or transmission media

Performance improves with a faster transmission medium. For example, the following list shows some typical raw data transfer rates:

Channel-to-channel (fiber optics)

4.0 MB/s

16 Mbps LAN

2.0 MB/s

Channel-to-channel (regular)

1.0 MB/s

4 Mbps LAN

0.5 MB/s

High speed T1 carrier (1.544 Mbps)

0.193 MB/s

Fast remote 56 Kbps phone line

0.007 MB/s

19.6 Kbps modem

0.002 MB/s

9600 bps modem

0.001 MB/s

The data transfer rate is limited by the slowest transmission medium in the path to the IBM mainframe database server.

- Network adapter or communication controller

You should carefully plan the memory usage of the network adapter and communication controller. In addition, you should work with a network specialist to ensure that the controller has the capability to handle the extra traffic generated by DB2 Connect.

- Network topology

If data crosses from LAN to LAN, and from one network to another network, consider the travel time. Bridges, routers, and gateways will add to the elapsed time. For example, reducing the number of bridges that are crossed reduces the number of hops required for each request.

The physical distance between nodes should also be considered. Even if a message is transferred by satellite, the transfer time is limited by the speed of light (3×10^8 m/s) and the round-trip distance between the sender and receiver.

- Network traffic

If the bandwidth of the network has been fully utilized, both the response time and the data transfer rate for a single application will decrease.

Congestion can occur in the network when data accumulates at a particular part of the network; for example, at an old NCP with a very small buffer size.

- Network reliability
If the error rate of the network is high, the throughput of the network will decrease and this will cause poor performance because of data retransmission.

CLI/ODBC application performance tuning

CLI/ODBC is an SQL application programming interface that can be called by your database applications. CLI functions invoke DB2 stored procedures which, in turn, access the system catalog tables. If CLI/ODBC applications are encountering performance problems, consider tuning their behaviour with CLI/ODBC keywords.

Some applications use ODBC APIs to gather metadata information that is used in further processing. The ten metadata API calls that can be made are:

- SQLTables
- SQLColumns
- SQLSpecialcolumns
- SQLStatistics
- SQLPrimarykeys
- SQLForeignkeys
- SQLTablePrivileges
- SQLColumnPrivileges
- SQLProcedures
- SQLProcedureColumns

Certain CLI/ODBC applications that use the metadata APIs listed previously might query all of the objects within the database. For example, an SQLTables call requests metadata for all the tables in the database. On a large system, such requests can result in a lot of network traffic, take a considerable amount of time and consume a considerable amount of server resources.

Several CLI/ODBC initialization keywords can be used to limit the amount of data that will be returned by the initial API calls during the "information gathering" stage after the database is first connected to. These keywords can be set by:

1. Manually editing the `db2cli.ini` file.
2. Updating the database CLI configuration using the DB2 Command Line Interface.

The keywords are:

- DBName
- TableType
- SchemaList
- SysSchemae
- GrantorList
- GranteeList

Chapter 10. Troubleshooting

Troubleshooting DB2 Connect servers

The DB2 Connect environment involves multiple software, hardware and communications products. Troubleshooting is best approached by a process of elimination and refinement of the available data to arrive at a conclusion (the location of the error).

After gathering the relevant information and based on your selection of the applicable topic, proceed to the referenced section.

Gathering relevant information

Troubleshooting includes narrowing the scope of the problem and investigating the possible causes. The proper starting point is to gather the relevant information and determine what you know, what data has not been gathered, and what paths you can eliminate.

At a minimum answer the following questions.

- Has the initial connection been successful?
- Is the hardware functioning properly?
- Are the communication paths operational?
- Have there been any communication network changes that would make previous directory entries invalid?
- Has the database been started?
- Is the communication breakdown between one or more clients and the DB2 Connect Server (gateway); between the DB2 Connect gateway and the IBM mainframe database server
- What can you determine by the content of the message and the tokens returned in the message?
- Will using diagnostic tools such as **db2trc**, **db2pd**, or **db2support** provide any assistance at this time?
- Are other machines performing similar tasks working correctly?
- If this is a remote task, is it successful if performed locally?

Initial connection is not successful

If you have configured a new connection in DB2 Connect and cannot connect successfully, troubleshoot the problem by answering a set of questions that are structured into a checklist.

Review the following questions and ensure that the installation steps were followed:

1. *Did the installation processing complete successfully?*
 - Were all the prerequisite software products available?
 - Were the memory and disk space adequate?
 - Was remote client support installed?
 - Was the installation of the communications software completed without any error conditions?

2. *For UNIX operating systems, was an instance of the product created?*
 - As root did you create a user and a group to become the instance owner and SYSADM group?
3. *If applicable, was the license information processed successfully?*
 - For UNIX operating systems, did you edit the nodelock file and enter the password that IBM supplied?
4. *Were the IBM mainframe database server and workstation communications configured properly?*
 - There are three configurations that must be considered:
 - a. The IBM mainframe database server configuration identifies the application requester to the server. The IBM mainframe server database management system will have system catalog entries that will define the requester in terms of location, network protocol and security.
 - b. The DB2 Connect workstation configuration defines the client population to the server and the IBM mainframe server to the client.
 - c. The client workstation configuration must have the name of the workstation and the communications protocol defined.
 - Problem analysis for not making an initial connection includes verifying that PU (physical unit) names are complete and correct, or verifying for TCP/IP connections that the correct port number and hostname have been specified.
 - Both the IBM mainframe server database administrator and the Network administrators have utilities available to diagnose problems.
5. *Do you have the level of authority required by the IBM mainframe server database management system to use the IBM mainframe server database?*
 - Consider the access authority of the user, rules for table qualifiers, the anticipated results.
6. *If you attempt to use the Command Line Processor (CLP) to issue SQL statements against a IBM mainframe database server, are you unsuccessful?*
 - Did you follow the procedure to bind the CLP to the IBM mainframe database server?

If the checklist does not guide you to a resolution, contact IBM Support.

Problems encountered after an initial connection

If DB2 Connect can no longer connect successfully, troubleshoot the problem by answering a set of questions that are structured into a checklist.

Answering the following questions can help you to identify the source of the connection problem:

1. *Are there any special or unusual operating circumstances?*
 - Is this a new application?
 - Are new procedures being used?
 - Are there recent changes that might be affecting the system? For example, have any of the software products or applications been changed since the application or scenario last ran successfully?
 - For application programs, what application programming interface (API) was used to create the program?
 - Have other applications that use the software or communication APIs been run on the user's system?

- Has a fix pack recently been installed? If the problem occurred when a user tried to use a feature that had not been used (or loaded) on their operating system since it was installed, determine IBM's most recent fix pack and load it *after* installing the feature.
2. *Has this error occurred before?*
 - Are there any documented resolutions to previous error conditions?
 - Who were the participants and can they provide insight into a possible course of action?
 3. *Have you explored using communications software commands that return information about the network?*
 - TCP/IP might have valuable information retrieved from using TCP/IP commands and daemons.
 4. *Is there information returned in the SQLCA (SQL communication area) that can be helpful?*
 - Problem handling procedures should include steps to examine the contents of the SQLCODE and SQLSTATE fields.
 - SQLSTATEs allow application programmers to test for classes of errors that are common to the DB2 family of database products. In a distributed relational database network this field might provide a common base.
 5. *Was START DBM executed at the Server?* Additionally, ensure that the **DB2COMM** environment variable is set correctly for clients accessing the server remotely.
 6. *Are other machines performing the same task able to connect to the server successfully?* The maximum number of clients attempting to connect to the server might have been reached. If another client disconnects from the server, is the client who was previously unable to connect, now able to connect?
 7. *Does the machine have the proper addressing?* Verify that the machine is unique in the network.
 8. *When connecting remotely, has the proper authority been granted to the client?* Connection to the instance might be successful, but the authorization might not have been granted at the database or table level.
 9. *Is this the first machine to connect to a remote database?* In distributed environments routers or bridges between networks might block communication between the client and the server. For example, when using TCP/IP, ensure that you can PING the remote host.

Diagnostic tools

DB2 Connect provides diagnostic tools to troubleshoot problems. You can also use the tools and diagnostic files provided with the operating system.

When you encounter a problem, you can use the following troubleshooting information:

- All diagnostic data including dump files, trap files, error logs, notification files, and alert logs are found in the path specified by the diagnostic data directory path (**diagpath**) database manager configuration parameter:

If the value for this configuration parameter is null, the diagnostic data is written to one of the following directories or folders:

- For Linux and UNIX environments: `INSTHOME/sqllib/db2dump/ $m`, where *INSTHOME* is the home directory of the instance.
- For supported Windows environments:
 - If the **DB2INSTPROF** environment variable is not set then `x:\SQLLIB\DB2INSTANCE` is used where `x:\SQLLIB` is the drive reference and

the directory specified in the **DB2PATH** registry variable, and the value of **DB2INSTANCE** has the name of the instance.

Note: The directory does not have to be named SQLLIB.

- If the DB2 registry variable **DB2INSTPROF** is set then x:\DB2INSTPROF**DB2INSTANCE** is used where x:\DB2INSTPROF is the path specified in the **DB2INSTPROF** registry variable and **DB2INSTANCE** is the name of the instance (by default, the value of **DB2INSTDEF** on Windows 32-bit operating systems).
- For Windows operating systems, you can use the Event Viewer to view the administration notification log.
- The available diagnostic tools that can be used include **db2trc**, **db2pd**, **db2support** and **db2diag**
- For Linux and UNIX operating systems, the **ps** command, which returns process status information about active processes to standard output.
- For UNIX operating systems, the core file that is created in the current directory when severe errors occur. It contains a memory image of the terminated process, and can be used to determine what function caused the error.

Chapter 11. Messages

Common DB2 Connect problems

There are common symptoms and solutions for connection problems that you can encounter when using DB2 Connect.

In each case, you are provided with:

- A combination of a message number and a return code (or protocol specific return code) associated with that message. Each message and return code combination has a separate heading, and the headings are ordered by message number, and then by return code.
- A symptom, usually in the form of a sample message listing.
- A suggested solution, indicating the probable cause of the error. In some cases, more than one suggested solution might be provided.

SQL0965 or SQL0969

Symptom

Messages SQL0965 and SQL0969 can be issued with a number of different return codes from IBM DB2 for IBM i, DB2 for z/OS, and DB2 Server for VM and VSE.

When you encounter either message, you should look up the original SQL code in the documentation for the database server product issuing the message.

Solution

The SQL code received from the IBM mainframe database cannot be translated. Correct the problem, based on the error code, then resubmit the failing command.

SQL5043N

Symptom

Support for one or more communications protocols failed to start successfully. However, core database manager functionality started successfully.

Perhaps the TCP/IP protocol is not started on the DB2 Connect server. There might have been a successful client connection previously.

If `diaglevel = 4`, then the **db2diag** log files might contain a similar entry, for example:

```
2001-05-30-14.09.55.321092 Instance:svtdbm5 Node:000
PID:10296(db2tcpm) Appid:none
common_communication sqlcctcpconnmgr_child Probe:46
DIA3205E Socket address "30090" configured in the TCP/IP
services file and
required by the TCP/IP server support is being used by another
process.
```

Solution

This warning is a symptom which signals that DB2 Connect, acting as a server for remote clients, is having trouble handling one or more client communication protocols. These protocols can be TCP/IP and others, and

usually the message indicates that one of the communications protocols defined to DB2 Connect is not configured properly.

Often the cause might be that the **DB2COMM** profile variable is not defined, or is defined incorrectly. Generally, the problem is the result of a mismatch between the **DB2COMM** variable and names defined in the database manager configuration (for example, **svcname** or **nname**).

One possible scenario is having a previously successful connection, then getting the SQL5043 error message, while none of the configuration has changed. This could occur using the TCP/IP protocol, when the remote system abnormally terminates the connection for some reason. When this happens, a connection might still appear to exist on the client, and it might become possible to restore the connection without further intervention by issuing the following commands.

Most likely, one of the clients connecting to the DB2 Connect server still has a handle on the TCP/IP port. On each client machine that is connected to the DB2 Connect server, enter the following commands:

```
db2 terminate
db2stop
```

SQL30020

Symptom

SQL30020N Execution failed because of a Distributed Protocol Error that will affect the successful execution of subsequent commands and SQL statements.

Solutions

Service should be contacted with this error. Run the **db2support** command before contacting service.

SQL30060

Symptom

SQL30060N "<authorization-ID>" does not have the privilege to perform operation "<operation>".

Solution

When connecting to DB2 for z/OS, the Communications Database (CDB) tables have not been updated properly.

SQL30061

Symptom

Connecting to the wrong IBM mainframe database server location - no target database can be found.

Solution

The wrong server database name might be specified in the DCS directory entry. When this occurs, SQLCODE -30061 is returned to the application.

Check the DB2 node, database, and DCS directory entries. The target database name field in the DCS directory entry must correspond to the name of the database based on the platform. For example, for a DB2 for z/OS database, the name to be used should be the same as that used in the Boot Strap Data Set (BSDS) "LOCATION=*locname*" field, which is also provided in the DSNL004I message (LOCATION=*location*) when the Distributed Data Facility (DDF) is started.

The correct commands for a TCP/IP node are:

```
db2 catalog tcpip node node_name remote host_name_or_address
      server_port_no_or_service_name
db2 catalog dcs database local_name as real_db_name
db2 catalog database local_name as alias at node node_name
      authentication server
```

To connect to the database you then issue:

```
db2 connect to alias user user_name using password
```

SQL30081N with Return Code 79

Symptom

```
SQL30081N  A communication error has been detected.
Communication protocol
being used: "TCP/IP".  Communication API being used: "SOCKETS".
Location
where the error was detected: "".  Communication function
detecting the error:
"connect".  Protocol specific error code(s): "79", "*", "*".
SQLSTATE=08001
```

Solution(s)

This error can occur in the case of a remote client failing to connect to a DB2 Connect server. It can also occur when connecting from the DB2 Connect server to a IBM mainframe database server.

1. The **DB2COMM** profile variable might be set incorrectly on the DB2 Connect server. Check this. For example, the command `db2set db2comm=tcpip` should appear in `sql1lib/db2profile` when running DB2 Enterprise Server Edition on AIX.
2. There might be a mismatch between the TCP/IP service name and port number specifications at the IBM data server client and the DB2 Connect server. Verify the entries in the TCP/IP services files on both machines.
3. Check that DB2 is started on the DB2 Connect server. Set the Database Manager Configuration **diaglevel** to 4, using the command:

```
db2 update dbm cfg using diaglevel 4
```

After stopping and restarting DB2, look in the **db2diag** log files to check that DB2 TCP/IP communications have been started. You should see output similar to the following one:

```
2001-02-03-12.41.04.861119 Instance:svtdbm2 Node:00
PID:86496(db2sysc) Appid:none
common communication sqlcctcp_start_listen Probe:80
DIA3000I "TCPIP" protocol support was successfully started.
```

SQL30081N with Protocol Specific Error Code 10032

Symptom

```
SQL30081N  A communication error has been detected.
Communication protocol
being used: "TCP/IP".  Communication API being used: "SOCKETS".
Location
where the error was detected: "9.21.85.159".  Communication
function detecting
the error: "send".  Protocol specific error code(s): "10032",
"*, "*".
SQLSTATE=08001
```

Solution

This error message might be received when trying to disconnect from a machine where TCP/IP communications have already failed. Correct the problem with the TCP/IP subsystem.

On most machines, simply restarting the TCP/IP protocol for the machine is the way to correct the problem. Occasionally, recycling the entire machine might be required.

SQL30082 RC=24 During CONNECT**Symptom**

SQLCODE -30082 The username or the password supplied is incorrect.

Solution

Ensure that the correct password is provided on the CONNECT statement if necessary. Password not available to send to the target server database. A password has to be sent from the IBM data server client to the target server database. On certain platforms, for example AIX, the password can only be obtained if it is provided on the CONNECT statement.

Appendix A. 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. 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 28. 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 28. 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 29. DB2 Connect technical information

Name	Form number	Available in print	Availability date
<i>Installing and Configuring DB2 Connect Servers</i>	SC27-5517-00	Yes	28 July 2013
<i>DB2 Connect User's Guide</i>	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 B. 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

&&

SQLCODE mapping file 102

A

about this book v

agentpri database manager configuration parameter 140

AIX

CD mounting 33

DVD mounting 33

installing

DB2 Connect server products 17, 31

application design

overview 130

application development

IBM Data Server Driver Package 6

application name monitor element 110

application requesters

DRDA definition 86

parameters 95

application servers

DRDA definition 86

applications

binding 73

compatibility

DB2 for z/OS 115

compound SQL 130

DB2 for z/OS 115

ODBC 81

performance

application design 130

running 115

stored procedures 130

AS target database name 91

ATOMIC compound SQL

not supported in DB2 Connect 130

authentication

DB2 Connect 124, 126

directory customization worksheet 95

system database directory 89

types

CLIENT 124

DATA_ENCRYPT 124

default 124

KERBEROS 124

SERVER 124

SERVER_ENCRYPT 124

SERVER_ENCRYPT_AES 124

validation 124

authorities

binding 73

automatic client reroute

details 78

setup 78

B

benchmarking

performance 127

bidirectional CCSID support

BIDI parameter 91

language support 16, 83

bind list

DB2 Connect 73

BINDADD authority

DB2 Connect 73

binding

applications 73

authority 73

packages

DB2 Connect 73

utilities

DB2 Connect 73, 81

block size

DB2 Connect 140

blocking

data 130

bootstrap data set (BSDS) parameters 90

bottlenecks

performance 127

transactions 127

C

cached address list 70

CDs

mounting

AIX 33

HP-UX 36

Linux 38

Solaris 41

CHAR data type

details 148

character data representation architecture (CDRA) 86

character data types 148

CLI

overview 150

trusted connections 119

client and server connections

overview 1

client applications

communication recovery 78

CLIENT authentication type

DB2 Connect 124

client DB alias 110

clients

overview 79

remote 79

code pages

conversion

exceptions 16, 83

supported 13

coded character set identifier (CCSID)

bidirectional languages 16, 83

bidirectional support

details 91

languages 16, 83

command line processor (CLP)

performance 130

SQL statements 5

- commands
 - db2licm
 - setting license policy 49
 - db2osconf
 - determining kernel configuration parameter values 28
 - db2setup
 - displaying DB2 Setup wizard in your national language 13
 - GET SNAPSHOT
 - overview 108
- COMMIT statement
 - statically bound 130
- communication protocols
 - DRDA host access configuration 66
- communications
 - recovery 78
- configuration
 - DB2 Connect server products 30
 - host connections 6
 - TCP/IP
 - using CLP 71
- configuration parameters
 - AGENTPRI 140
 - DIR_CACHE 140
 - max_coordagents
 - details 135
 - overview 133
 - MAXAGENTS 140
 - num_initagents 133, 135
 - num_poolagents 133, 135
 - NUMDB 140
 - RQRIOBLK 140
- connection concentrator
 - connection pooling comparison 139
 - DB2 Connect 140
 - overview 133, 135
 - worker agents 135
- connection pooling
 - connection concentrator comparison 139
 - overview 133
- connections
 - DB2 Connect Enterprise Edition 7
 - DRDA hosts through communications server 66
 - hosts directly 6
 - IBM mainframe directly 6
 - pooling
 - advantages 135
 - connection concentrators 135
 - overview 133
 - reestablishing
 - DB2 Connect Enterprise Edition 7
 - direct to host 6
- connectivity servers
 - DB2 Connect Enterprise Edition 7
- contention
 - system resources 144
- conversion
 - character 16, 83
 - host 148
- core files
 - problem determination 153
- CPUs
 - performance tools 127
- CREATE IN COLLECTION NULLID authority 73

D

- D (disconnect) parameter 91
- DAS (DB2 administration server)
 - see DB2 administration server (DAS) 84
- data
 - accessing
 - DB2 Connect 80
 - blocking 130
 - flows
 - DB2 Connect 86, 127
 - sources 88
 - transferring
 - between hosts and workstations 76
 - performance 149
 - rates 127, 149
- data movement
 - DB2 Connect 76
- data types
 - CHAR 148
 - character 148
 - conversion
 - performance effect 148
 - floating-point
 - host data conversion 148
 - INTEGER
 - host data conversion 148
 - packed decimal 148
 - VARCHAR
 - overview 148
 - zoned decimal 148
- DATA_ENCRYPT authentication type 124
- Database Connection Services (DCS) directory
 - updating entries 88
 - values 91
- database directories
 - Database Connection Services (DCS) 88
 - multiple entries 95
 - node 88
 - updating 88
- database requests
 - grouping for performance 130
- database system monitor
 - overview 5
 - remote clients 107
- databases
 - aliases
 - directory customization worksheet 95
 - system database directory 89
 - grouping requests 130
 - host 4, 65
 - names
 - DCS directory 91
 - directory customization worksheet 95
 - system database directory 89
 - performance tools 127
 - tuning 142
- dates
 - time zone support 91
- DB2 administration server (DAS)
 - overview 84
- DB2 Connect
 - administration utilities 5
 - configuring 100
 - connection concentrators 140
 - DB2 for VSE & VM 68
 - disk requirements 23

- DB2 Connect (*continued*)
 - Enterprise Edition
 - connectivity servers 7
 - transaction processing monitors 8
 - XA-compliant transaction managers 100
 - host support 80, 83
 - IBM i connections 63
 - installing
 - non-Administrator installation 47
 - prerequisites 17
 - mainframe support 80, 83
 - memory requirements 23
 - moving data 76
 - overview 1, 2, 80
 - prerequisites 17
 - scenarios 6
 - server products
 - configuring 30
 - installing (AIX) 17, 31
 - installing (HP-UX) 19, 34
 - installing (Linux) 20, 37
 - installing (overview) 30
 - installing (Solaris Operating System) 21, 39
 - installing (Windows) 22, 42
 - post-upgrade tasks 60
 - pre-upgrade tasks 57
 - Sysplex support 68, 69
 - System i support
 - overview 83
 - upgrading
 - overview 55, 56
 - procedure 58
 - zSeries support 83
 - DB2 documentation
 - available formats 159
 - DB2 documentation versions
 - IBM Knowledge Center 162
 - DB2 for VM & VSE
 - preparing for connections from DB2 Connect 68
 - DB2 for z/OS
 - configuring 68
 - node directory values 90
 - updating system tables 68
 - DB2 Setup wizard
 - language identifiers 14
 - DB2ADMNS group
 - adding users 49
 - db2licm command
 - registering licenses 48, 72
 - setting license policy 49
 - db2osconf command
 - determining kernel configuration parameter values 28
 - db2setup command
 - language setting 13
 - DB2USERS user group
 - adding users 49
 - DCS (Database Connection Services) directory
 - see Database Connection Services (DCS) directory 91
 - dcs1ari.map file 102
 - dcs1dsn.map file 102
 - dcs1qsq.map file 102
 - ddcs400.lst file 73
 - ddcsmvslst file 73
 - ddcsvm.lst file 73
 - ddcsvse.lst file 73
 - default language setting
 - Windows 15
 - DESCRIBE statement
 - compound SQL statements 130
 - performance with PREPARE statement 130
 - diagnostic information
 - overview 153
 - dir_cache parameter 140
 - directories
 - customizing 95
 - system database
 - updating 88
 - values 89
 - directory cache support configuration parameter
 - DB2 Connect tuning 140
 - directory schema
 - extending
 - Windows 46
 - Distributed Data Management (DDM)
 - Distributed Relational Database Architecture (DRDA) 86
 - Distributed Relational Database Architecture (DRDA)
 - data access 85
 - DB2 Connect 86
 - overview 85
 - distributed requests
 - overview 88
 - distributed units of work
 - multisite updates 98
 - overview 85
 - servers supported 98
 - two-phase commit 98
 - documentation
 - PDF files 160
 - printed 160
 - terms and conditions of use 163
 - DVDs
 - mounting
 - AIX 33
 - HP-UX 36
 - Linux 38
 - Solaris 41
 - dynamic SQL
 - performance
 - techniques 130
 - processing effects 4, 98
- ## E
- error messages
 - DB2 Connect 155
 - errors
 - troubleshooting 151
 - examples
 - connection concentrators 135
 - XA concentrators 135
 - EXECUTE IMMEDIATE statement
 - application design 130
 - export utility
 - transferring data between hosts and workstations 76
 - extra query blocks
 - EXTRA BLOCKS SRV parameter 145
 - overview 145
- ## F
- federated databases
 - distributed requests 88

- fix packs
 - installing
 - DB2 Connect 50
- floating-point data types
 - conversion 148
- FOR FETCH ONLY clause
 - SELECT statement 130
- FORCE command 110
- Formatted Data Object Content Architecture (FDOCA) 86

G

- GET SNAPSHOT command
 - overview 108

H

- hardware
 - network performance 149
- help
 - SQL statements 162
- host databases 6
 - configuring TCP/IP 71
 - connectivity
 - high availability 147
 - load balancing 147
- HP-UX
 - installing
 - DB2 Connect servers 19, 34
 - kernel configuration parameters
 - modifying 27
 - recommended values 28
 - mounting media 36

I

- IBM Data Server Driver for JDBC and SQLJ
 - levels for DB2 Connect versions 24
- IBM i
 - DB2 Connect 83
- IBM Knowledge Center
 - DB2 documentation versions 162
- import utility
 - transferring data between host and workstation 76
- InfoSphere Federation Server
 - overview 6
- installation
 - DB2 Connect
 - prerequisites 17
 - server products 30
 - user accounts (Windows) 43
 - zSeries running Linux
 - DB2 Connect 26
- INTEGER data type
 - host data conversion 148
- interface languages
 - changing
 - UNIX 16
 - Windows 15
 - overview 13
- INTERRUPT_ENABLED (disconnect) parameter 91

J

- Java
 - DB2 Connect product support 24
- JDBC
 - drivers
 - details 24

K

- Kerberos authentication protocol
 - DB2 Connect 124
 - OS/390 125
 - z/OS 125
- kernel configuration parameters
 - HP-UX
 - db2osconf command 28
 - modifying 27
 - recommended 28
 - Linux
 - modifying 28
 - Solaris 30

L

- LANG environment variable
 - setting 13, 16
- languages
 - bidirectional support 16, 83
 - DB2 Connect interface 13
 - DB2 interface 15
 - DB2 Setup wizard for language identifiers 14
- licenses
 - registering
 - db2licm command 48, 72
 - setting
 - db2licm command 49
- Linux
 - installing
 - DB2 Connect on zSeries 26
 - DB2 Connect server products 20, 37
 - kernel parameters
 - modifying 28
 - mounting
 - CDs 38
 - DVDs 38
 - uninstalling DB2 Connect
 - root 54
- LIST DCS APPLICATIONS command
 - output 110
- LOCALDATE parameter 91
- locales
 - DB2 Connect interface languages 13

M

- max_coordagents database manager configuration parameter
 - details 135
 - overview 133
- maxagents database manager configuration parameter
 - deprecated 140
- memory
 - usage tools 127
- monitoring
 - connections 107
 - Windows Performance Monitor 107

mounting CDs or DVDs

- AIX 33
- HP-UX 36
- Linux 38
- Solaris 41

multisite updates

- distributed unit of work (DUOW) 98
- enabling 98
- sync point manager 100

N

national language support (NLS)

- converting character data 16, 83
- displaying DB2 Setup wizard 13

networks

- data transfer rates 149
- performance tools 127
- tuning 142

node directories

- updating 88
- values 90

nodes

- names
 - directory customization worksheet 95
 - system database directory values 89

NOMAP parameter

- turning off SQLCODE mapping 101

NOT ATOMIC compound SQL

- application design 130

notices 165

NULLID 73

num_initagents database manager configuration parameter

- configuring idle agents pool 133
- overview 135

num_poolagents database manager configuration parameter

- configuring idle agents pool 133
- overview 135

numdb database manager configuration parameter

- DB2 Connect 140

O

ODBC

- binding packages 81
- CLI/ODBC application performance tuning 150

online DB2 documentation

- IBM Knowledge Center 162

P

packages

- host database servers 73
- System i database servers 73

packed decimal data type 148

paging block size 140

parameter strings

- commas 91
- double commas 91

parameters

- directories 95
- strings 96
- SYSPLEX 91

performance

- application design 130
- command line processor (CLP) impact 130

performance (*continued*)

- connection concentrator 139
- connection pooling 139
- DB2 Connect
 - increasing transfer rates 145
 - overview 127
 - troubleshooting 144
- DB2 for z/OS 144
- network hardware 149
- system resources 144

post-upgrade tasks

- DB2 Connect servers 60

pre-upgrade tasks

- DB2 Connect servers 57

predicates

- performance of logic 130

PREPARE statement

- application design 130
- performance effect 130

problem determination

- connections 151
- diagnostic tools
 - overview 153
- post-connection 152

process status utility 153

ps command

- overview 153

Q

query blocks

- increasing DB2 Connect data transfer rates 145

R

references

- defining multiple database entries 95

remote units of work

- example 87
- overview 87

response times

- DB2 Connect 127

ROLLBACK statement

- statically bound 130

rqrioblk configuration parameter

- tuning 140

S

scenarios

- TCP/IP security 126

SDKs

- product levels 24

security

- Kerberos 125
- node directory values 90
- TCP/IP 126
- types 95
- user groups 49

SELECT statement

- application design 130
- FOR FETCH ONLY clause 130
- updatable 130

SERVER authentication type

- DB2 Connect 124

- SERVER_ENCRYPT authentication type
 - DB2 Connect 124
- SERVER_ENCRYPT_AES authentication type 124
- SOCKS
 - nodes
 - mandatory environment variables 90
- Solaris operating systems
 - DB2 Connect 30, 41
 - DB2 Connect server products 21, 39
 - modifying kernel parameters 30
 - mounting CDs or DVDs 41
- SQL
 - dynamic 130
 - static 130
- SQL statements
 - COMMIT 130
 - DB2 Connect 4, 98
 - DESCRIBE 130
 - EXECUTE IMMEDIATE 130
 - FOR FETCH ONLY clause of SELECT 130
 - help
 - displaying 162
 - PREPARE 130
 - ROLLBACK 130
 - SELECT 130
- SQL_ATTR_
 - TRUSTED_CONTEXT_PASSWORD
 - switching users on trusted connection through CLI 121
 - TRUSTED_CONTEXT_USERID
 - switching users on trusted connection through CLI 121
 - USE_TRUSTED_CONTEXT
 - creating trusted connection through CLI 120
- SQL0965 error code 155
- SQL0969 error code 155
- SQL30020 error code 155
- SQL30060 error code 155
- SQL30061 error code 155
- SQL30073 error code 155
- SQL30081N error code 155
- SQL30082 error code 155
- SQL5043N error code 155
- SQLCODE
 - mapping
 - overview 101
 - tailoring 102
- SQLCODE mapping
 - turning off 101
- SQLDA
 - allocation size 130
- SQLSTATE
 - class codes 102
- static SQL
 - performance 130
 - processing effects 4, 98
- sync point manager (SPM)
 - configuration parameters
 - default values 100
 - scenarios 100
- Sysplex
 - configuration requirements 71
 - DB2 Connect support 69, 70
 - fault tolerance 70
 - load balancing 70
 - overview 68
 - parameter 91
 - priority information 70
 - System z 69, 82

- system database directory
 - updating 88
 - values 89
- System i
 - database servers
 - configuring connections 71
 - DB2 Connect support 83
- system resources
 - contention 144
- system status
 - GET SNAPSHOT command 108
- System z
 - DB2 Connect
 - support overview 83

T

- target databases
 - names 91, 95
- TCP/IP
 - authentication scenarios 126
 - configuring
 - host connections 64, 66, 71
 - System i database servers 71
 - DB2 for z/OS 64, 66, 71
 - DOMAIN 90
 - host names 95
 - port numbers 95
 - remote host names 90, 95
 - RESPORT 90
 - resynch port 90
 - RFC-1323 extensions 146
 - service names 90
 - TCPPORT 90
- terms and conditions
 - publications 163
- territory codes
 - page support 16, 83
- throughput
 - transactions 127
- tokens
 - SQLCODE values 101
- transaction processing monitors
 - BEA Tuxedo 8
 - DB2 Connect 8
 - multisite updates 98
 - OLTP 8
- transactions
 - DB2 Connect 8, 101, 127
 - distributed 98
 - loosely coupled
 - DB2 Connect 101
 - multisite updates 85, 98
 - performance 127
 - transaction processing monitors 8
 - two-phase commit 85
 - unit of work (UOW) 85
 - XA distributed applications 101
- troubleshooting
 - connections 151, 152
 - DB2 Connect 144, 151, 155
 - gathering information 151
 - performance
 - DB2 Connect 144
- trusted connections
 - CLI
 - creating 120

- trusted connections (*continued*)
 - CLI (*continued*)
 - switching users 121
 - terminating 120
 - DB2 Connect 119
- trusted contexts
 - CLI 120
 - DB2 Connect 119
- tuning
 - DB2 Connect
 - overview 140
 - parameters 140
 - DB2 for z/OS 144
 - host databases 142
 - networks 142
- Tuxedo
 - DB2 Connect Enterprise Edition 8
- two-phase commit
 - enabling 98
 - port for two-phase commit resynchronization operations 90

U

- uninstallation
 - DB2 Connect 53, 54
 - root installations 54
- units of work
 - distributed 98
 - overview 85
 - remote 87
- UNIX
 - changing DB2 Connect interface language 16
 - uninstalling
 - DB2 Connect 54
- updates
 - database directories 88
- upgrades
 - DB2 Connect
 - overview 55, 56
 - procedure 58
- user accounts
 - DB2 administration server (Windows) 43
 - instance user (Windows) 43
 - required for installation (Windows) 43
- user groups
 - DB2ADMNS 49
 - DB2USERS 49
 - security 49
- utilities
 - binding 73, 81
 - database system monitor 5
 - DB2 Connect administration 5
 - ddcspkgn 73
 - ps (process status) 153

V

- VARCHAR data type
 - overview 148
- VTAM
 - preparing z/OS for connections from DB2 Connect 64

W

- WebSphere MQ
 - DB2 Connect 140
- window scaling
 - RFC-1323 extensions 146
- Windows
 - applications 6
 - default language setting 15
 - installing
 - DB2 Connect (with non-Administrator access) 47
 - DB2 Connect server products (procedure) 42
 - DB2 Connect server products (requirements) 22
 - Performance Monitor
 - monitoring DB2 applications 107
 - uninstalling DB2 Connect 53
 - user accounts
 - DB2 Connect product installation 43
- worksheets
 - directory customization 95

X

- X/Open distributed transaction processing (DTP) model
 - overview 8
- XA
 - concentrator examples 135
 - resource managers 8
 - trusted connections 119
- XA transaction managers
 - connection concentrator 135
 - overview 8

Z

- zoned decimal data types 148
- zSeries
 - installing DB2 Connect for Linux 26



Printed in USA

SC27-5518-01



Spine information:

IBM DB2 Connect 10.5

DB2 Connect User's Guide

