IBM® DB2® Connect

IBM® DB2 Connect™ User's Guide

Version 8

SC09-4835-00

IBM® DB2® Connect

IBM® DB2 Connect™ User’s Guide

Version 8

SC09-4835-00

Before using this information and the product it supports, be sure to read the general information under *Notices*.

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

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

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

Contents

About this Book

This book contains general usage information about the following IBM DB2 Connect products:

- DB2 Connect Personal Edition for Linux and Windows.
- DB2 Connect Enterprise Edition (EE) for AIX, HP-UX, Linux, Solaris Operating Environment, and Windows.
- DB2 Connect Unlimited Edition, used to access DB2 UDB for OS/390 and z/OS.

Who Should Read this Book

This book is intended for programmers and administrators who are responsible for setting up and maintaining DB2 Connect connections. These connections can exist between DB2 clients and any of the following application server database management systems:

- DB2 Universal Database (UDB) for OS/390 Version 5 or later, and DB2 UDB for OS/390 and z/OS Version 7 or later
- DB2 for VSE & VM
- DB2 for AS/400 Version 3.1 or later, and DB2 UDB for iSeries Version 5.1 or later
- Other relational database management system that implements a DRDA application server function.

Notes:

1. DB2 UDB does not require DB2 Connect to allow host or iSeries applications to access DB2 UDB data.
2. DB2 UDB for OS/390 Version 5.1 or higher is required to use DRDA Level 3 functions, including TCP/IP database connections, and stored procedures with multi-row answer sets.
3. DB2 UDB for OS/390 Version 6.1 or higher is required to use the DRDA Level 4 functions for which support is provided in DB2 Connect. These functions include support for big integer, large object, row ID, and user-defined distinct data types.

Part 1. DB2 Connect concepts

Chapter 1. DB2 Connect concepts

DB2 Connect

DB2® Connect provides fast and robust connectivity to IBM® mainframe databases for e-business and other applications running under UNIX® and Windows® operating systems.

DB2 Connect™ has several connection solutions. DB2 Connect Personal Edition provides direct connectivity to host and iSeries™ database servers, while DB2 Connect Enterprise Edition provides indirect connectivity that allows clients to access host and iSeries database servers through the DB2 Connect server. DB2 Connect Unlimited Edition provides a unique packaging solution that makes product selection and licensing easier.

Related concepts:

- "DB2 Connect and SQL statements" on page 11
- "DB2 Connect and DRDA" on page 16
- "DB2 Connect scenarios" on page 21

Related reference:

- "DB2 Connect product offerings" on page 3
- "Host databases" on page 10
- "DB2 Connect administration utilities" on page 12
- "Functions delivered in previous releases" on page 6

DB2 Connect product offerings

DB2 Connect has several connection solutions. DB2 Connect Personal Edition provides direct connectivity to host and iSeries database series, while DB2 Connect Enterprise Edition provides indirect connectivity that allows clients to access host and iSeries database servers through the DB2 Connect server. DB2 Connect Unlimited Edition provides a unique packaging solution that makes product selection and licensing easier.

DB2 Connect Enterprise Edition

DB2 Connect Enterprise Edition is a connectivity server that concentrates and manages connections from multiple desktop clients and web applications to DB2 database servers running on host or iSeries systems. IBM's DB2 Universal Database (UDB) for iSeries, DB2 for OS/390 and z/OS, and DB2 for VSE & VM databases continue to be the systems of choice for managing most critical data for the world's largest organizations. While these host and iSeries databases manage the data, there is a great demand to integrate this data with applications running on Windows and UNIX workstations.

DB2 Connect Enterprise Edition enables local and remote client applications to create, update, control, and manage DB2 databases and host systems using Structured Query Language (SQL), DB2 APIs (Application Programming Interfaces), ODBC (Open Database Connectivity), JDBC (Java Database Connectivity), SQLJ (Embedded SQLJ for Java), or DB2 CLI (Call Level Interface). In addition, DB2 Connect supports Microsoft Windows data interfaces such as ActiveX Data Objects (ADO), Remote Data Objects (RDO), and Object Linking and Embedding (OLE) DB.

DB2 Connect Enterprise Edition is currently available for AIX, HP-UX, Linux, Solaris, and Windows operating systems. These servers provide support for applications running on UNIX (AIX, HP-UX, Linux, and Solaris operating environment), and Windows workstations.

DB2 Connect Enterprise Edition is often installed on an intermediate server to connect DB2 clients to a host or iSeries database. It can also be used on machines where multiple local users want to access the host or iSeries servers directly.

For example, DB2 Connect Enterprise Edition may be installed on a large machine with many local users. It may also be installed on a web server, a machine that is running a Transaction Processor (TP) monitor, or other 3-tier application servers with multiple local SQL application processes and threads. In these cases, you can install DB2 Connect Enterprise Edition on the same machine for simplicity, or on a separate machine to off-load CPU cycles.

DB2 Connect Enterprise Edition is most appropriate for environments where:

- Host and iSeries database servers do not support native TCP/IP connectivity and direct connectivity from desktop workstations via SNA is not desirable.
- Web servers run web-based applications.
- Web servers run web-based application using data-aware Java applications.
- A middle-tier application server is used.
- TP monitors, such as CICS, Encina, Microsoft Transaction Server (MTS), Tuxedo, Component Broker, and MQSeries, are used.

DB2 Connect Personal Edition

DB2 Connect Personal Edition provides access from a single workstation to DB2 databases residing on servers such as OS/390, z/OS, OS/400, VM and VSE, as well as to DB2 Universal Database servers on UNIX and Windows operating systems. DB2 Connect Personal Edition provides the same rich set of APIs as DB2 Connect Enterprise Edition.

This product is currently available for Linux and Windows operating systems.

DB2 Connect Personal Edition is used to connect a single Windows operating system, or Linux workstation, to a host or iSeries database. DB2 Connect Personal Edition is best suited for environments where native TCP/IP support is provided by the database servers, and the application being deployed is a traditional 2-tier client-server application.

For example, DB2 Connect Personal Edition is a good choice for enabling traditional 2-tier VisualBasic and Microsoft Access applications. Applications that require a mid-tier application server need to use DB2 Connect Enterprise Edition.

DB2 Connect Unlimited Edition

DB2 Connect Unlimited Edition is a unique package offering that allows complete flexibility of DB2 Connect deployment and simplifies product selection and licensing. This product contains both DB2 Connect Personal Edition and DB2 Connect Enterprise Edition with license terms and conditions that allow the unlimited deployment of any DB2 Connect product. License charges are based on the size of the S/390 or zSeries server that DB2 Connect users will be working with.

This package offering is only available for OS/390 and z/OS systems, and licensing is only valid for DB2 for OS/390 and z/OS data sources.

Related concepts:

- "DB2 Connect" on page 3
- "DB2 Connect and SQL statements" on page 11
- "Thin clients" in the *Installation and Configuration Supplement*

Related tasks:

- "Response file installation of DB2 on UNIX" in the *Installation and Configuration Supplement*
- "Response file installation of DB2 on Windows" in the *Installation and Configuration Supplement*

Related reference:

- "Host databases" on page 10
- "DB2 Connect administration utilities" on page 12

Functions delivered in previous releases

This section provides a summary of the enhancements introduced at each version and release is presented.

Functions delivered in DB2 Connect Version 7 Release 2

DB2 Connect Version 7.2 included the following enhancements:

- Improved support for Microsoft Transaction Server (MTS) and COM+ technologies
- DB2 Connect Web Starter Kit
- DB2 Connect for Linux on S/390

Functions delivered in DB2 Connect Version 7 Release 1

DB2 Connect Version 7.1 included the following enhancements:

- XA Concentrator
- Multisite update improvements
- Support DCL SNA product

Functions delivered in DB2 Connect Version 6 Release 1

DB2 Connect Version 6.1 included the following enhancements:

- Using the TCP/IP Communications Protocol
- Two-Phase Commit
- Multi-Row Stored Procedures
- DCE Security
- DCE Cell Directory Support and Host Systems
- Enhanced Security Failure Notification
- Enhanced System/390 SYSPLEX Exploitation
- Optimized Catalog Access for ODBC and JDBC Applications
- New BIND Options
- Microsoft Transaction Server Support
- Simplified Password Management
- Client Information Enhancements
- Bidirectional Language Support
- Monitoring DB2 Connect Applications
- Two-Phase Commit Enhancements
- Simplified DB2 Syncpoint Manager Configuration

- Additional Data Objects and Types Supported
- DB2 Connect for Personal Communications

Functions delivered in DB2 Connect Version 5 Release 2

DB2 Connect Version 5.2 included the following enhancements for the host and AS/400 DRDA functions:

- DCE Cell Directory support
- Enhanced security failure notification
- Enhanced System/390 SYSPLEX exploitation
- Optimized Catalog Access for ODBC and JDBC Applications
- Microsoft Transaction Server support
- New BIND options (DYNAMICRULES)
- Set Client Information API enhancements
- SQLDescribeParam support for DB2 Connect
- Support for Bidirectional Languages
- System Monitor enhancements
- Two-phase commit support enhancements
- Simplified DB2 Syncpoint Manager Configuration
- Support for the SCO** operating system
- Support for Big Integer, Large Object, Row ID, and User Defined Distinct data types.

Functions delivered in DB2 Connect Version 5.0

DB2 Connect Version 5.0 included the following enhancements:

- New easier to purchase packaging:
  - A single DB2 Connect Personal Edition package that contains OS/2, Windows 3.1, Windows 95, and Windows NT versions of the product. This package contains everything that is needed to get started, including a complimentary copy of Lotus Approach.
  - A single DB2 Connect Enterprise Edition package that contains OS/2, Windows NT, and all UNIX versions.
- Capability:
  - New Level 3 ODBC driver with many improvements
  - Updated JDBC driver for better Java support
  - Support for stored procedures that return multi-row result sets and multiple result sets (requires DB2 UDB for OS/390Version 5.1 or higher)
  - Built-in replication support
  - Generic bind option: you can specify any bind option supported by the host database.

  - SYSPLEX exploitation (DB2 Connect Personal Edition only; requires DB2 UDB for OS/390 Version 5.1 or higher)
- Usability:
  - New installation method
  - TCP/IP database connections are much easier to configure (requires DB2 UDB for OS/390 Version 5.1 or higher, or DB2 for AS/400 Version 4.2)
  - Integrated SNA support with point-and click configuration (DB2 Connect Personal Edition only)
  - New point and click configuration utility for configuring host connections.
  - Much easier process for connecting desktop client systems to DB2 Connect Enterprise Edition servers. Clients can discover DB2 Connect servers and all of the databases that are defined on each server
  - Improved ODBC traces with detailed information for performance analysis
  - Control Center and other GUI tools that simplify several DBA tasks
- Security:
  - DCE security (requires DB2 UDB for OS/390 Version 5.1 or higher)
  - Ability to run ODBC applications without having to authorize each user to base tables. Users can now bind their ODBC driver in such a way as to allow applications to run under the authority of the person that bound the ODBC driver.
- Performance:
  - Faster access to the DB2 catalog for ODBC applications
  - Reduced network traffic:
    - Early close for cursors
    - Deferred prepare
    - Reduced byte count on Compound SQL
    - Several other network flow enhancements
    - Support for ASCII storage on the host (requires DB2 UDB for OS/390 Version 5.1 or higher)
- Connectivity:
  - Support for DRDA over TCP/IP connections to other IBM DRDA Application Servers, as they introduce support for TCP/IP.
  - SNA over TCP/IP via integrated MPTN support (requires AnyNet on the host).

  - Support for additional SNA connectivity options:
    - IBM Communication Server for Windows NT
    - IBM Personal Communications
- Other:
  - Ability to initiate 2-phase commit transactions over TCP/IP (requires DB2 UDB for OS/390 Version 5.1 or higher)
  - Ability for desktop applications to participate in a 2-phase commit transactions without the need for a gateway (TCP/IP only, requires DB2 UDB for OS/390 Version 5.1 or higher)
  - Ability to use DB2 UDB for OS/390 for added reliability of transaction coordination (requires DB2 UDB for OS/390 Version 5.1 or higher, and TCP/IP)
  - Numerous other enhancements and fixes affecting all aspects of system performance, reliability, and usability.

Functions delivered in DDCS Version 2 Release 4

Distributed Database Connection Services (DDCS) for Windows Single-User Version 2.4 introduced:

- A Data Source Setup tool to help you define connections to host and AS/400 servers quickly and easily.
- Wall Data Rumba, to provide you with the communications support required to make these connections.
- A DB2 Password Expiration Maintenance utility (DB2PEM), which enables you to change your DB2 for MVS password without logging on to TSO.
- Enhancements to improve the performance and flexibility of DB2 Connect:
  - Deferred Prepare, which improves the performance of ODBC and other dynamic SQL applications by attaching the PREPARE request to a subsequent request instead of sending it separately.
  - Asynchronous ODBC, which improves the availability of ODBC applications. Previously, these might have appeared to be delayed while processing long queries in some network situations.
  - On AIX and OS/2, support for multi-threaded applications, which gives non-ODBC applications the ability to maintain multiple database connections with their own contexts.

Functions delivered in DDCS Version 2 Release 3

New features in DDCS Version 2 Release 3.1 included:

- Two-phase commit for DRDA connections using the LU6.2 Syncpoint Manager (SPM) on OS/2 and AIX.

New features in DDCS Version 2 Release 3.0 included:

- Client application performance could be improved by running stored procedures on DB2 for MVS Version 4.1 and DB2 for AS/400 Version 3.1 servers.
- Able to work with multiple databases in a single transaction.
- Able to improve performance by concatenating SQL statements.
- Able to implement chargeback accounting by using accounting strings.
- Able to use many new bind options when binding applications to a DRDA application server.
- When using a DCE directory, the ability to consolidate the directory information needed by all your clients in a central repository.
- Greater flexibility in SQLCODE processing.
- Diagnostic information stored in a readable format and consolidated in a single location (the first failure service log).
- The DDCSSETP environment variable was replaced by BIND and PREPARE options such as `SQLERROR CONTINUE`, simplifying operations.
- Various other performance improvements were also implemented.

Related concepts:

- "DB2 Connect" on page 3

Host databases

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

OS/390 or z/OS

DB2 UDB for OS/390 (Version 5 and later), DB2 UDB for OS/390 and z/OS (Version 7 and later). A DB2 Universal Database for OS/390 and z/OS subsystem identified by its LOCATION NAME. The LOCATION NAME can be determined by logging into TSO and issuing the following SQL query using one of the available query tools:

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

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.

VSE DB2 for VSE running in a partition identified by its DBNAME

VM DB2 for VM running in a CMS virtual machine identified by its DBNAME

OS/400

DB2 for iSeries, an integral part of the OS/400 operating system. Only one database can exist on an iSeries server. If the database will be used by applications outside the iSeries system, the database must be given a name in the relational database directory. This name is known as the Relation Database Name (RDB Name).

To display the RDB Name of your iSeries system execute the command **WRKRDBDIRE** on your iSeries system. The RDB Name of your Local system has *LOCAL specified in the Remote Location column. To change the RDB Name use the command CHGRDBDIRE.

Related concepts:

- "DB2 Connect" on page 3
- "DB2 Connect and SQL statements" on page 11

Related reference:

- "DB2 Connect product offerings" on page 3
- "DB2 Connect administration utilities" on page 12

DB2 Connect and SQL statements

DB2 Connect forwards SQL statements submitted by application programs to host or iSeries™ database servers. DB2 Connect can forward almost any valid SQL statement, as well as supporting DB2® APIs (Application Programming Interfaces), ODBC (Open Database Connectivity), JDBC (Java Database Connectivity), SQLJ (Embedded SQLJ for Java), or DB2 CLI (Call Level Interface)

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 host or iSeries 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 are supported by DB2 Connect.

Different host or iSeries database servers implement SQL differently. DB2 Connect fully supports the common IBM® SQL, as well as the DB2 for

OS/390® and z/OS, DB2 for VSE & VM (formerly SQL/DS), and DB2 for iSeries implementations of SQL. IBM SQL is strongly recommended for maintaining database independence.

Related concepts:

- "DB2 Connect" on page 3

Related reference:

- "DB2 Connect product offerings" on page 3
- "Host databases" on page 10
- "DB2 Connect administration utilities" on page 12

DB2 Connect administration utilities

The following utilities are available to help a DB2 Connect administrator:

- The Command Line Processor lets you issue SQL statements against a host or iSeries database server database. It flows the SQL statements to the database that you specify.
- The DB2 Command Center provides a graphical interface to the Command Line Processor.
- Import and export utilities let you load, import, and export data to and from a file on a workstation and a host or iSeries database server database. These files can then be used for importing data into databases, spreadsheets, and other applications running on your workstation.
- Users of DB2 Connect Enterprise Edition running on Windows NT and Windows 2000 can use the Event Viewer and the Performance Monitor. Using the Event Viewer, you can view exception events logged by DB2 Connect. Using the Performance Monitor, you can monitor and manage the performance of DB2 Connect servers either locally or remotely.
- The DB2 Control Center lets you administer and monitor all aspects of DB2 Connect servers. It also allows administrators to work with DB2 for OS/390 or z/OS database objects, such as tables, views, buffer pools, and threads.
- The database system monitor utility lets the system administrator monitor system connections. This function is only available when DB2 Connect is acting as server. This utility also helps the system administrator determine the source of an error. The system administrator can correlate client applications with the corresponding jobs running on the host or iSeries database server.

Related concepts:

- "Database system monitor" in the *System Monitor Guide and Reference*
- "DB2 Connect" on page 3

- "DB2 Connect and SQL statements" on page 11

Related tasks:

- Appendix A, "Moving Data With DB2 Connect" on page 199

Related reference:

- "DB2 Connect product offerings" on page 3
- "Host databases" on page 10

Chapter 2. Distributed Relational Database Architecture (DRDA)

Distributed Relational Database Architecture

Distributed Relational Database Architecture™ (DRDA) is a set of protocols that permits multiple database systems, both IBM® and non-IBM, 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 may be multiple requests per unit of work.
- There is one database management server per request.
- Commitment is coordinated across multiple database servers.

Related concepts:

- "DB2 Connect and DRDA" on page 16
- "Remote unit of work" on page 18
- "Distributed requests" on page 19
- "Multisite Updates" on page 69
- "DRDA and data access" on page 16

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 Windows, and several UNIX® platforms. These tools are part of the DB2 Application Development Client. The DB2 Application Development Client supports several API types: embedded SQL, JDBC, SQLJ, and the DB2 Call Level Interface (DB2 CLI). These APIs can be used by programmers to build applications in a variety of programming languages.

Application developers can also use APIs provided by other companies. For example, Microsoft® ODBC and ADO are used by Windows® application programmers to develop database applications. DB2 Connect provides an ODBC driver and an OLE DB Provider that support applications developed using the ODBC and ADO APIs. IBM does not provide tools for developing ODBC applications; these tools are provided by the Microsoft Corporation.

Related concepts:

- "Distributed Relational Database Architecture" on page 15
- "DB2 Connect and DRDA" on page 16
- "Applications in Host or iSeries Environments" on page 39

DB2 Connect and DRDA

DB2 Connect implements the DRDA® architecture to reduce the cost and complexity of accessing data stored in DB2® UDB for iSeries, DB2 UDB for OS/390® and z/OS, DB2 for VSE & VM, 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; it is the application that is requesting data. An *application server (AS)* is the code that handles the database end of the connection. In the DB2 Connect environment, the DB2 Connect workstation can only function as an application requester on behalf of application programs.

Figure 1 shows the flow of data between the DB2 Connect server and the host or iSeries™ server in the case where there are local clients only.

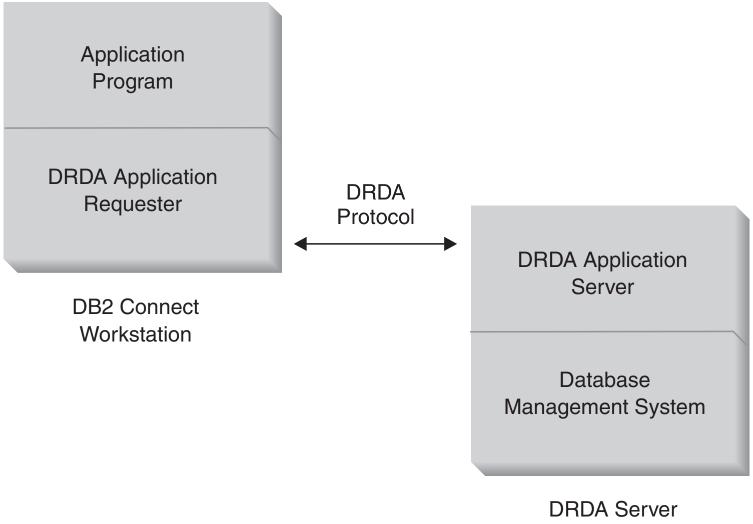


Figure 1. Data flow between a DB2 Connect™ server and a host or iSeries server

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

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

These architectures used as building blocks. The data streams which flow over the network are specified by 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.

Related concepts:

- "Distributed Relational Database Architecture" on page 15
- "Remote unit of work" on page 18
- "Distributed requests" on page 19

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 may roll back the unit of work.

For example, Figure 2 shows a database client running a funds transfer application that accesses a database containing checking and savings account tables, as well as a banking 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).

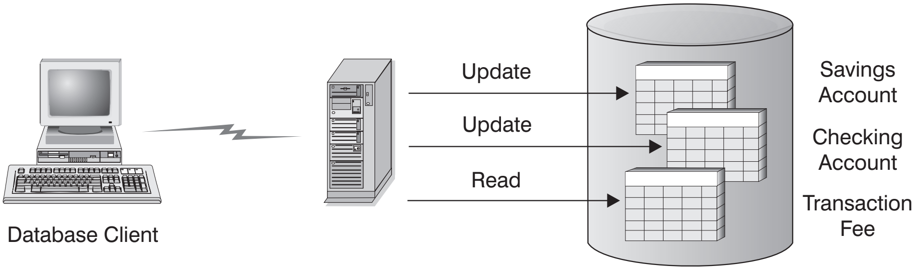


Figure 2. Using a Single Database in a Transaction

To set up such an application, you must:

1. Create the tables for the savings account, checking account and banking 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.

Related concepts:

- "Distributed Relational Database Architecture" on page 15
- "DB2 Connect and DRDA" on page 16
- "Distributed requests" on page 19
- "Remote Unit of Work" in the *Application Development Guide: Programming Client Applications*

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 OS/390® or 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 UDB for Windows, and UNIX, DB2 for OS/390 and z/OS, and DB2 UDB for iSeries), and Oracle.

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 consist 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.

Related concepts:

- "Distributed Relational Database Architecture" on page 15
- "DB2 Connect and DRDA" on page 16
- "Remote unit of work" on page 18

Chapter 3. DB2 Connect scenarios

DB2 Connect scenarios

DB2 Connect can provide a variety of solutions to your host or iSeries™ database access needs. This topic outlines several scenarios that may apply to your particular needs or environment.

Related concepts:

- "DB2 Connect" on page 3
- "Direct access to host databases" on page 21
- "DB2 Connect Enterprise Edition as a connectivity server" on page 23
- "DB2 Connect and web applications" on page 25
- "DB2 Connect and application servers" on page 31
- "DB2 Connect and transaction processing monitors" on page 34

Related reference:

- "DB2 Connect product offerings" on page 3

Scenarios

Direct access to host databases

DB2 Connect's basic feature is providing a direct connection to a host database from desktop applications running on Windows® 32–bit systems, or Linux workstations. DB2 Connect Personal Edition is the simplest way to provide this solution.

Each workstation that has DB2 Connect Personal Edition installed can establish a direct TCP/IP connection to DB2® UDB for OS/390® and z/OS, DB2 UDB for iSeries™ and DB2 UDB for Windows NT, Windows 2000, and UNIX® 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.

On Windows 32-bit systems, DB2 Connect Personal Edition has integrated APPC support as well, to communicate with DB2 databases on host systems that require APPC. However, the use of TCP/IP is highly recommended instead of SNA, when native TCP/IP support is available.

Figure 3 shows workstations directly connected to a host or iSeries database server. Each workstation has DB2 Connect Personal Edition installed.

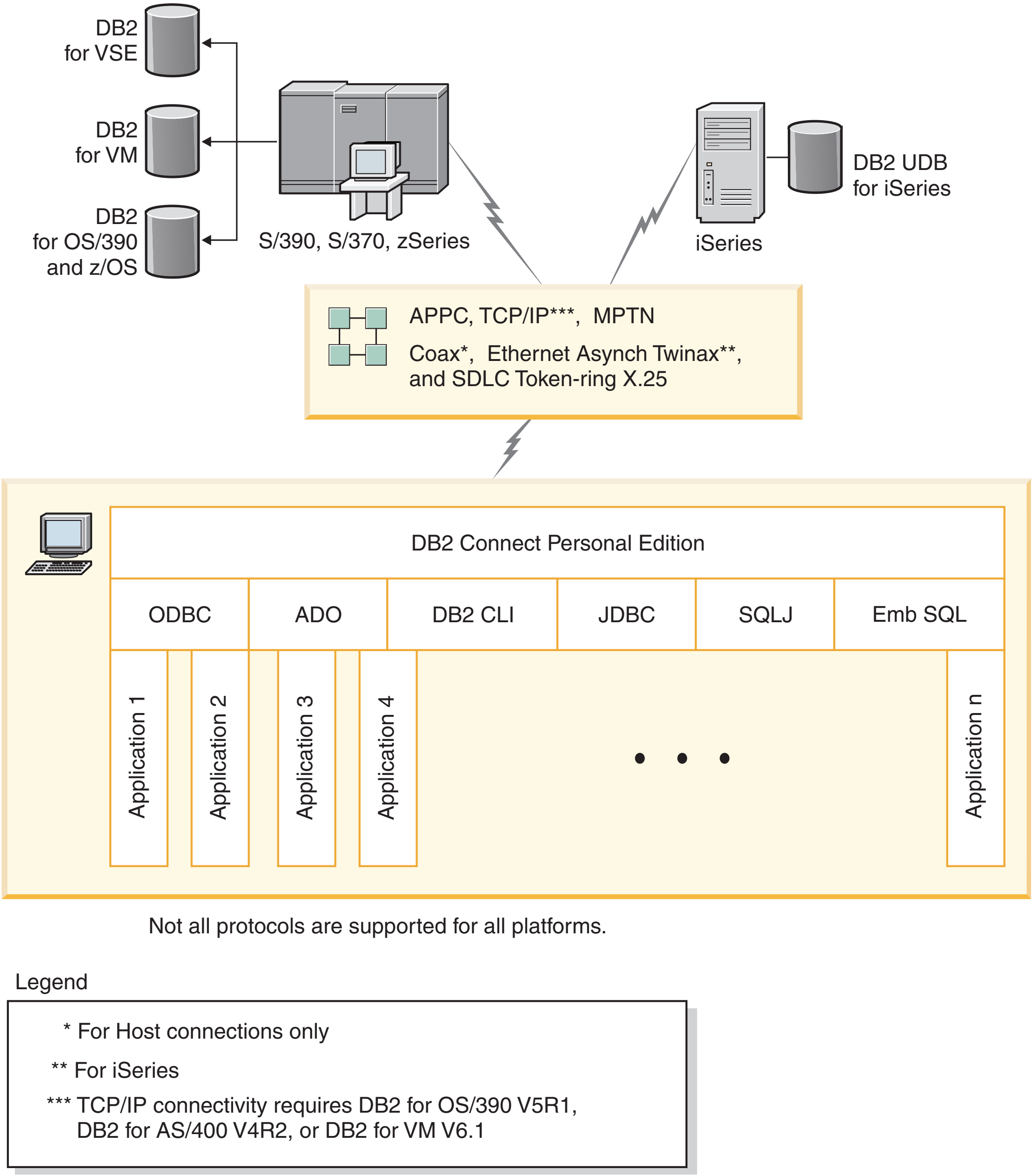


Figure 3. Direct Connection Between DB2 Connect and a host or iSeries database server

Notes:

1. You do not need to have DB2 Universal Database installed on the DB2 Connect workstation. If you want a complete relational database management system on the DB2 Connect workstation, order DB2 Universal Database.
2. The DB2 Application Development Client is now a part of the DB2 Connect package and can be installed if a customer wants to use it for application development. In addition, DB2 Connect now includes Stored Procedure Builder that can be used to build, test, and deploy stored procedures for DB2 for OS/390 and z/OS.
3. C programmers developing Windows applications that use Microsoft® ODBC, OLE DB, or ActiveX Data Objects (ADO) should use the *Microsoft Open Database Connectivity Software Development Kit*. Programmers who want to develop applications using the Java™ programming language can use any Java development environments, such as IBM's VisualAge® for Java.

Related concepts:

- "DB2 Connect scenarios" on page 21
- "DB2 Connect Enterprise Edition as a connectivity server" on page 23
- "DB2 Connect and web applications" on page 25
- "DB2 Connect and application servers" on page 31
- "DB2 Connect and transaction processing monitors" on page 34

Related reference:

- "DB2 Connect product offerings" on page 3

DB2 Connect Enterprise Edition as a connectivity server

A DB2 Connect server enables multiple clients to connect to host or iSeries™ data and can significantly reduce the effort that is required to establish and maintain access to enterprise data. Figure 4 on page 24 illustrates IBM's solution for environments in which you want a DB2® client to make an indirect connection to a host or iSeries database server through DB2 Connect Enterprise Edition.

In the following example, you could replace the DB2 Connect™ server with a DB2 UDB Enterprise Server Edition that has the DB2 Connect Server Support component installed.

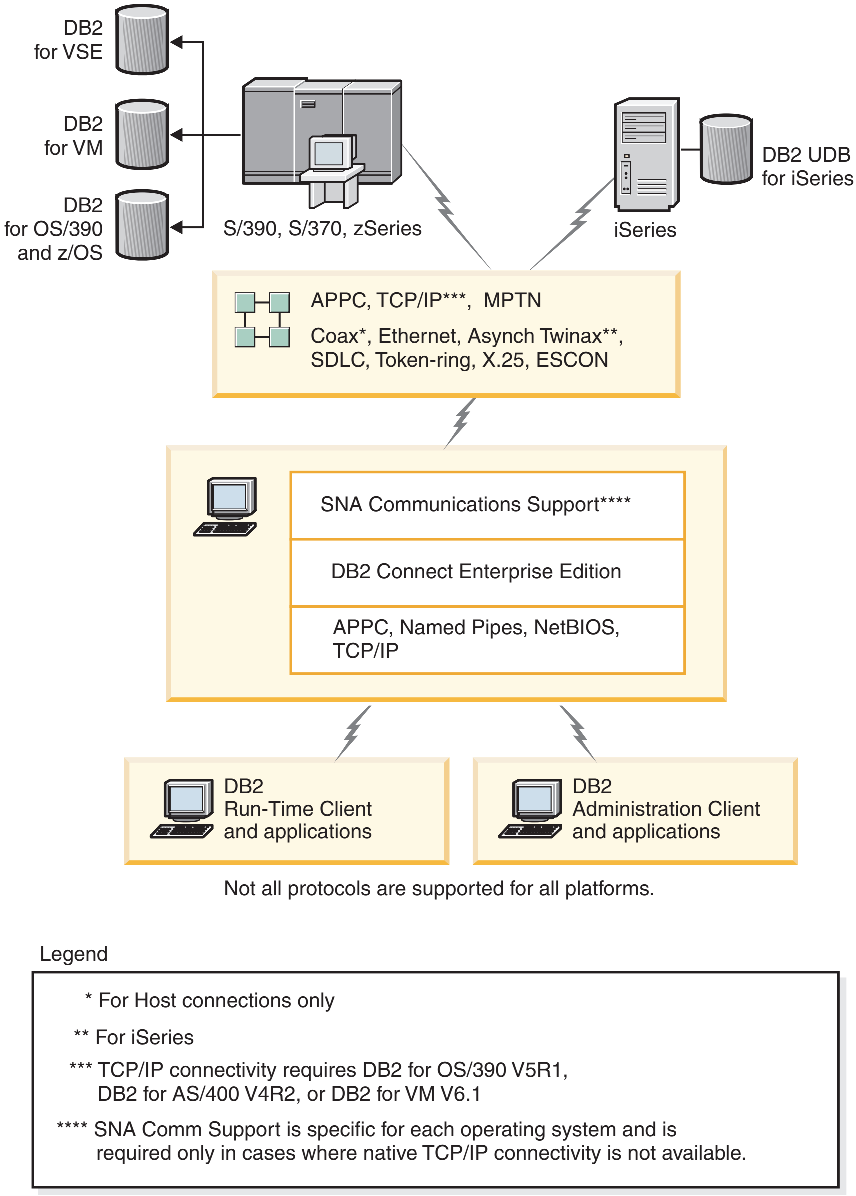

DB2
for VSE
DB2
for VM
DB2
for OS/390
and z/OS
S/390, S/370, zSeries
iSeries
DB2 UDB
for iSeries
APPC, TCP/IP***, MPTN
Coax*, Ethernet, Asynch Twinax**,
SDLC, Token-ring, X.25, ESCON
SNA Communications Support****
DB2 Connect Enterprise Edition
APPC, Named Pipes, NetBIOS,
TCP/IP
DB2
Run-Time Client
and applications
DB2
Administration Client
and applications
Not all protocols are supported for all platforms.
Legend
* For Host connections only
** For iSeries
*** TCP/IP connectivity requires DB2 for OS/390 V5R1,
DB2 for AS/400 V4R2, or DB2 for VM V6.1
**** SNA Comm Support is specific for each operating system and is
required only in cases where native TCP/IP connectivity is not available.

Figure 4. DB2 Connect Enterprise Edition

Related concepts:

- "DB2 Connect" on page 3
- "DB2 Connect scenarios" on page 21
- "DB2 Connect and web applications" on page 25

- “DB2 Connect and application servers” on page 31
- “DB2 Connect and transaction processing monitors” on page 34

Related reference:

- “DB2 Connect product offerings” on page 3

DB2 Connect and web applications

The web browser is rapidly becoming a standard interface for everything from online catalogs to intranet applications. For simple web applications, a web server alone may be sufficient. For high-volume applications that may require database access and transaction processing, IBM® offers solutions that use DB2® Connect to manage very high numbers of simultaneous transactions over the web.

Advantages and limitations of traditional CGI programming:

e-business applications on the World Wide Web typically use the Common Gateway Interface (CGI) to enable users to query back-end databases. Many companies also use web applications internally, and these usually have a database in the background as well.

Users fill out forms on a web page, and these forms are submitted via CGI to applications or scripts on the web server. The script will in turn use a provided database API to submit SQL queries to a host database. The same script can then build a web (HTML) page with results of the query and send it back to be displayed by the user’s web browser. For example, an online catalog, in which the user can query the availability and current price of particular goods or services.

CGI applications can be simple to design and easy to maintain. Since the CGI standard is both operating system- and language-independent, it is available on nearly all computing platforms. CGI programs can be written in C++, or in a scripting language such as Perl.

While CGI may seem like an ideal solution for web-based applications, it has significant shortcomings. The programming environment for CGI is not as sophisticated as other APIs. In addition, there is a scalability issue that will affect any large-scale e-commerce operation. Every time a CGI application is invoked, a new process is created on the web server. Each instance must make its own connection to the database, and each instance submits its own query. In high-volume transactional environments, this limitation can create significant performance issues.

You can use DB2 Connect with a web server to create robust, high-volume e-commerce applications. DB2 Connect provides several solutions that improve web-based application performance. Stored procedures allow DB2 Connect users to reduce the number of queries being sent to the database.

Connection pooling reduces the frequency of connections and disconnections to and from a database. For large operations where the limitations of CGI become important, see IBM Net.Data® and WebSphere® provide non-CGI connections to large enterprise applications.

Related concepts:

- "DB2 Connect" on page 3
- "DB2 Connect scenarios" on page 21
- "DB2 Connect Enterprise Edition as a connectivity server" on page 23
- "DB2 Connect and application servers" on page 31
- "DB2 Connect and transaction processing monitors" on page 34
- "Connection pooling" on page 153
- "DB2 Connect and IBM WebSphere" on page 26
- "DB2 Connect and Net.Data" on page 28
- "DB2 Connect on the web server" on page 30

DB2 Connect and IBM WebSphere

IBM® WebSphere® provides a more complete e-business solution than is possible with traditional CGI programming. WebSphere application servers not only perform the scripting possibilities of CGI, but also allow you to provide complex and high-end services through the web, using servlets, Active Server Pages, and enterprise JavaBeans™ and include support for Web based technologies such as Java, TCP/IP, HTTP, HTTPS, HTML, DHTML, XML, MIME, SMTP, IIOP, and X.509, among others. With WebSphere you can:

- Exploit industry standards to speed development and maximize inter-operability
- Plug in third-party tools technologies and application frameworks
- Analyze Web site content performance and usage
- Scale your site easily to accommodate more users and maintain throughput
- Deploy across a number of major operating environments (AIX, HP-UX, Linux, Novell NetWare, OS/390, z/OS, OS/400, Solaris Operating Environment, Microsoft® Windows® NT and Windows 2000)
- Use your existing web server, including those from Apache, IBM, Netscape, and Microsoft.

WebSphere is not one product, but a family of three products addressing three different target markets. The heart of the WebSphere solution is the WebSphere application server.

The WebSphere application server provides the environment for three types of objects. One is Java™ server pages, which are analogous to Active Server Pages. The second component consists of Java servlets, and the third is enterprise JavaBeans. Enterprise JavaBeans are the emerging standard for deploying very large-scale, robust enterprise-class applications.

Additionally, Data Access JavaBeans provide very sophisticated database functions specifically tailored to DB2. The DB2® Application Development Client provides support for Java embedded SQL (SQLJ). With DB2 JDBC and SQLJ support you can build and run SQLJ applications and applets. These contain static SQL and use embedded SQL statements that are bound to the DB2 database.

WebSphere applications can be deployed on the same platform as the web server and DB2 Universal Database. In the case of the DB2 UDB for OS/390® and z/OS, DB2 for VM, DB2 for VSE, and DB2 UDB for iSeries, WebSphere is deployed on the same platform as DB2 Connect Enterprise Edition.

There are several WebSphere solutions, as well as the Web Studio and WebSphere Performance Packs. The three WebSphere versions are:

Standard Edition
For Web site producers this server enables the use of Java servlets and JSP technology to quickly and easily transform Web sites and portals from static pages to vital sources of personalized dynamic Web content. It also includes industry-leading XML support for sharing information and data easily across groups or between enterprises and built-in site analysis technology that provides performance and usage information to help you maximize your company's return on its Web site investment.

Advanced Edition
For application programmers this high-performance EJB server enables deployment of business logic using EJB components. It provides scalability security connectivity and Java support and includes all of the functionality of the Standard Edition.

Enterprise Edition
For enterprise architects this server integrates disparate business systems across your organization to build robust e-business applications and maximize the reuse of resources. Enterprise Edition incorporates the capabilities of the award-wining IBM TXSeries™ and Component Broker technologies. It also includes all of the functionality of the Advanced and Standard Editions.

Related concepts:

- "DB2 Connect scenarios" on page 21

DB2 Connect and Net.Data

Net.Data, part of the DB2® Universal Database and DB2 Connect family, is a set of application development tools designed to help you create and maintain web-based transaction applications. You can use Net.Data® to access and alter data stored on a DB2 UDB for Windows® NT and Windows 2000, DB2 UDB for UNIX, DB2 UDB for OS/390® and z/OS, DB2 for VM, DB2 for VSE, and DB2 UDB for iSeries. The applications you create using Net.Data are stored on a web server and can be activated through a web browser.

Net.Data uses macros, or templates, to allow users with a basic understanding of HTML and SQL to build very sophisticated web applications. A macro is a text file that can be composed of Java, Java™ Scripts, HTML tags, and built-in functions. These macros can then be used to generate dynamic web pages with predefined layout, variables, and functions.

A basic Net.Data macro has seven distinct sections:

- Common sections, which basically serve as documentation aids for the programmer.
- Define section, which provides a place to specify variable definitions.
- Function section, which contains the main programming logic.
- Report section, which specifies the formatting logic for the Net.Data macro output.
- HTML section, which contains most of the HTML used in the web page.
- Include section, which is just a convenient way of including common parts of the macro that can be reused by other macros.
- Message section, where the error handling is provided.

The key feature of Net.Data, specifically for DB2, is that there is no client deployment required. The client in this implementation is simply a web browser.

The Net.Data processor is installed together with the DB2 Universal Database™ on a Windows NT, Windows 2000, or UNIX® workstation along with the Web server. When connecting to DB2 UDB for OS/390 and z/OS, DB2 for VSE and VM, and DB2 UDB for iSeries, all of the Net.Data infrastructure is deployed on a DB2 Connect™ server, along with a Web server.

Related concepts:

- "DB2 Connect scenarios" on page 21

DB2 Connect as a Java application server

Many of the shortcomings of CGI can be overcome by moving away from it and using Java™ instead. IBM® provides both applets and applications that allow you to substitute Java for CGI at every stage of a web transaction. The solutions IBM provides allow for a mix of techniques, which means you can use scripting solutions such as Net.Data® and Microsoft® Active Server Pages with DB2, or move towards a more robust implementation provided by a Java application server such as IBM WebSphere.

There are two Application Programming Interfaces (APIs) for Java programmers. The first, JDBC, is supported for using Java to develop data-aware Java Applets, Java Applications as well as Java servlets, Java server pages (JSP) and Enterprise Java Beans (EJB). JDBC is a call-level or method invocation API. The other Java API is SQLJ. SQLJ provides the ability to specify SQL in-line within a Java program. DB2® can use both APIs, on either the client or server side of a web transaction.

On the client side, applets, data-aware applets, and applications are supported. On the database side Java enablement consists of database objects, such as user-defined functions and stored procedures.

For DB2 for OS/390® and z/OS, DB2 for VSE and VM, and DB2 UDB for iSeries, there are two different ways to deploy a Java application. You can use the direct connectivity provided by DB2 Connect Personal Edition with TCP/IP or SNA, or you can choose to go through a DB2 Connect Enterprise Edition server that will provide connectivity to the mainframe or the iSeries™ back end.

In both cases, the user on the Web does not require any special software to access the database, only a standard web browser. The only thing that needs to be installed is a DB2 Connect server and any industry standard Web server. If the web server and DB2 Connect are not on the same physical machines, a DB2 client needs to be installed on the web server.

For DB2 for OS/390 and z/OS, the key component is DB2 Connect Enterprise Edition running on a mid-tier server. This component provides JDBC server enablement, in addition to connecting to the DB2 for OS/390 and z/OS, DB2 for VSE and VM, or DB2 UDB for iSeries server. Again, there is no need for any special software for the client's web browser.

IBM provides an extensive set of tools for developing Java applications and applets. For database connectivity, DB2 Developer's Edition provides a complete kit containing VisualAge® for Java Professional Edition, WebSphere® Application Server, Net.Data, as well as DB2 Universal Database™ and DB2 Connect for testing. IBM VisualAge for Java Enterprise Edition also contains development tools for large-scale enterprise applications. Third-party tools such as Borland JBuilder or Symantec Visual Cafe will also work with IBM's database solutions.

Related concepts:

- "DB2 Connect scenarios" on page 21

DB2 Connect on the web server

IBM® provides HTTP (Web) servers with all DB2 Connect products for UNIX, Windows® NT, and Windows 2000. DB2 Connect Enterprise Edition provides out-of-the-box support for Apache or Lotus® Domino™ Go web servers and can also work with any other web server such as Microsoft® Internet Information Server or Netscape Enterprise Server.

If you are working with the DB2® family of databases running on zSeries, iSeries, VM, and VSE systems, DB2 Connect Enterprise Edition is required on the Web server. DB2 Connect Enterprise Edition will provide the libraries and communication interfaces to enable Web servers to access these host and iSeries™ platforms. Either TCP/IP or SNA can be used to communicate between the Web server and a database running on zSeries, iSeries, VM or VSE.

Note: IBM web solutions provide the ability to work with multiple databases within the same CGI script or within the same transaction in a CGI script.

Stored procedures:

An important consideration for web applications, as in the client/server world, is to minimize the traffic that occurs between the HTTP server and the back end database. This consideration is particularly important in high-volume transactional processing, which is the heart of most e-business applications.

The recommended approach is to combine CGI application programming with the programming and business logic encapsulated in stored procedures. DB2 Universal Database on UNIX, and Windows, and DB2 UDB on OS/390® and z/OS, DB2 UDB for iSeries, and DB2 for VSE all share the same parameter convention for invoking stored procedures.

As with regular CGI, the web browser submits the form to the web server, where the CGI script is run. However, instead of each individual SQL statement being sent to the DB2 database, a request to execute a stored procedure is sent. This stored procedure encapsulates a number of SQL statements that would have otherwise been run individually. Stored procedures reduce the number of messages flowing back and forth between the CGI script and the back end database.

The key benefit of stored procedures is reduced network traffic between the HTTP server and the DB2 database back end.

Related concepts:

- "DB2 Connect scenarios" on page 21
- "DB2 Stored Procedures" in the *Application Development Guide: Programming Client Applications*
- "Stored Procedures in Host or iSeries Environments" on page 47

DB2 Connect and application servers

The rise of client-server applications allowed application designers to improve usability and decrease training costs by providing applications with graphical user interfaces on platforms such as Windows. At the same time, it allowed the flexibility of delegating database management function to robust database servers on a variety of operating systems and hardware platforms.

The client-server model, where application logic is distributed to client workstations, is commonly referred to as *2-tier client server*. In the 2-tier model, the application is deployed on the client tier and database server implements the server or the back-end tier. DB2® Connect provides complete support for 2-tier client-server applications, where database servers are DB2 UDB for OS/390® and z/OS, DB2 UDB for iSeries, or DB2 for VM and VSE.

With the increase in the size of the client-server applications, it became apparent that the 2-tier client-server model had significant limitations. Distributing large amounts of business logic to hundreds or even thousands of client workstations made change management a complex and costly undertaking. Any change in business rules required replacement of the client portion of the application. Often these application rollouts had to be on all client workstations in the enterprise at the same time to ensure that business rules are being applied consistently.

Another shortcoming of the 2-tier client-server model became apparent with scale is the amount of resources that are consumed by such applications. Deploying hundreds or thousands of *fat clients*, as 2-tier clients are often called, increased demands on processing power and capacity of each client workstation. Moreover, the demands on the database server are also greatly increased as each client required a dedicated database connection and the resources associated with maintaining such a connection. While the 2-tier client-server dependency of distributing business logic can be somewhat reduced by extensive use of stored procedures, the other shortcomings are not easily addressed without changes to the model.

An application server solution

As the cost and complexity of 2-tier client-server applications escalated, most of the largest applications embarked on the path to multi-tier client-server. Under the multi-tier model, the role of the database tier remains unchanged. However, the client tier is supplemented by a one or more middle tiers; typically one, therefore the name *3-tier*.

In the 3-tier model, the client is relegated to handling user interactions and does not contain any business logic. The middle-tier is comprised of one or more application servers. The goal of the application server is to provide robust, cost-efficient implementation of the logic behind the business processes and business rules. As with the 2-tier model, the business rules implementation is often supplemented by using stored procedures to improve performance.

Because client workstations no longer implement the bulk of the application logic and are only handling user interactions, the resource requirements for the client tier are greatly reduced. As a matter of fact, the client tier in the 3-tier model is often called *thin client*. In addition, because a centralized application server is handling requests from all of the clients, it has the ability to share resources, such as database connections between all of the clients. As a result, the database server no longer has to maintain dedicated connections for each application user.

Many examples of 3-tier applications servers exist in the industry today. Almost all Enterprise Resource Planning (ERP) vendors implement their applications using the 3-tier model, such as SAP R/3 and PeopleSoft V7 applications. Other examples include leading Enterprise Relationship Management vendors, such as Siebel and Vantive.

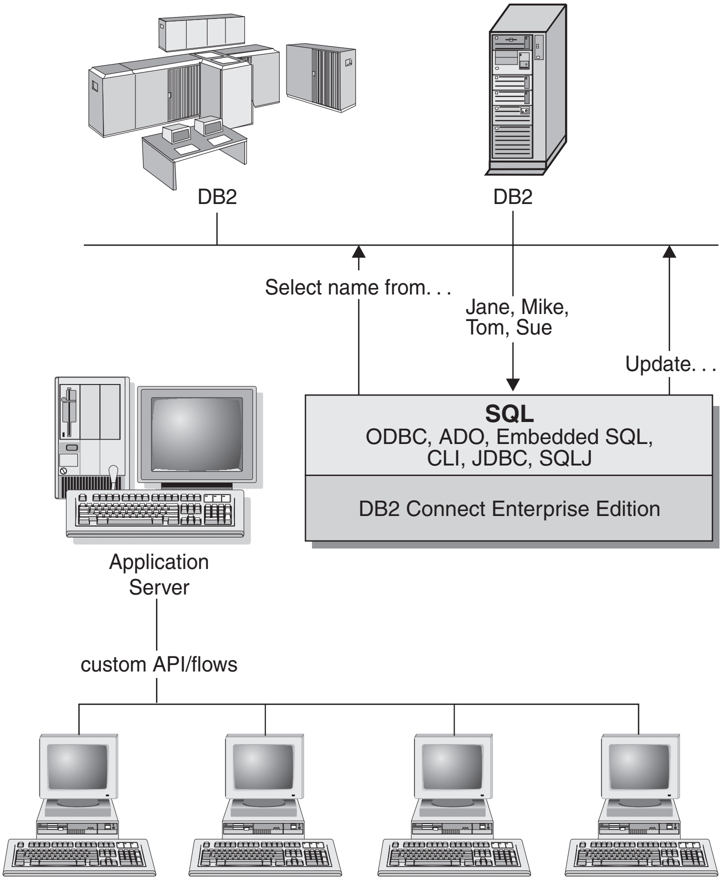
Application servers and DB2 Connect™

DB2 Connect Enterprise Edition servers provide comprehensive support for deploying multi-tier applications. The support provided by DB2 Connect includes a variety of APIs that can be used to develop application logic (ODBC, ADO, DB2 CLI, Embedded SQL, JDBC, and SQLJ), as well as a complete communication infrastructure for interacting with DB2 Family database servers.

DB2 Connect also supports implementations in which a database tier is comprised of multiple DB2 Family database servers. This allows application servers to implement transactions that update data residing on multiple database servers in a single transaction.

The two-phase commit protocol support provided by DB2 Connect assures the integrity of such distributed transactions. For example, an application can update data in a DB2 for OS/390 and z/OS™ database and DB2 UDB on Windows® 2000 in the same transaction. If distributed request support is installed and enabled, the application can read an Oracle database and update a DB2 family database in the same transaction.

In the following diagram, the APIs as well as the connectivity mechanism between the application server and the back-end database servers is provided by DB2 Connect Enterprise Edition.



Advanced features of DB2 Connect, such as connection pooling greatly reduce application resource requirements and simplify application server implementation.

DB2 Connect and application server configurations

DB2 Connect Enterprise Edition product (available on its own or as part of the DB2 Connect Unlimited Edition product package) is required for use with application servers. DB2 Connect Personal Edition is not supported and is not licensed for use with application servers. In addition, customers implementing application servers should review terms and conditions provided with their copy of DB2 Connect to understand the number of user licenses that need to be acquired.

There are two deployment methods for DB2 Connect in the application server environment. DB2 Connect Enterprise Edition installed on:

- The application server machine; or
- A separate communication server machine.

In most situations, installing a copy of DB2 Connect on the same server as the application server itself is the preferred solution. Installing DB2 Connect on the application server allows it to participate in any fail-over and load-balancing scheme that an application server may be implementing. This setup can potentially provide better performance since it eliminates an additional network hop that is required when DB2 Connect is installed on a separate server. Furthermore, the administration can be simplified since there is no need for an installing and maintaining an additional server.

Installing DB2 Connect on a separate server is a good option in situations where DB2 Connect Enterprise Edition is not available for the operating system or hardware platform where application server is running.

Related concepts:

- "DB2 Connect" on page 3
- "DB2 Connect Enterprise Edition as a connectivity server" on page 23
- "DB2 Connect and web applications" on page 25
- "DB2 Connect and transaction processing monitors" on page 34
- "Connection concentrator" on page 156
- "Connection pooling" on page 153

Related reference:

- "DB2 Connect security considerations for DB2 for OS/390 and z/OS" on page 189

DB2 Connect and transaction processing monitors

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.

A *transaction* can be thought of as a routine event, usually a request for service, in running the day-to-day operations of an organization. The orderly processing of transactions is the type of work for which TP monitors were designed.

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.

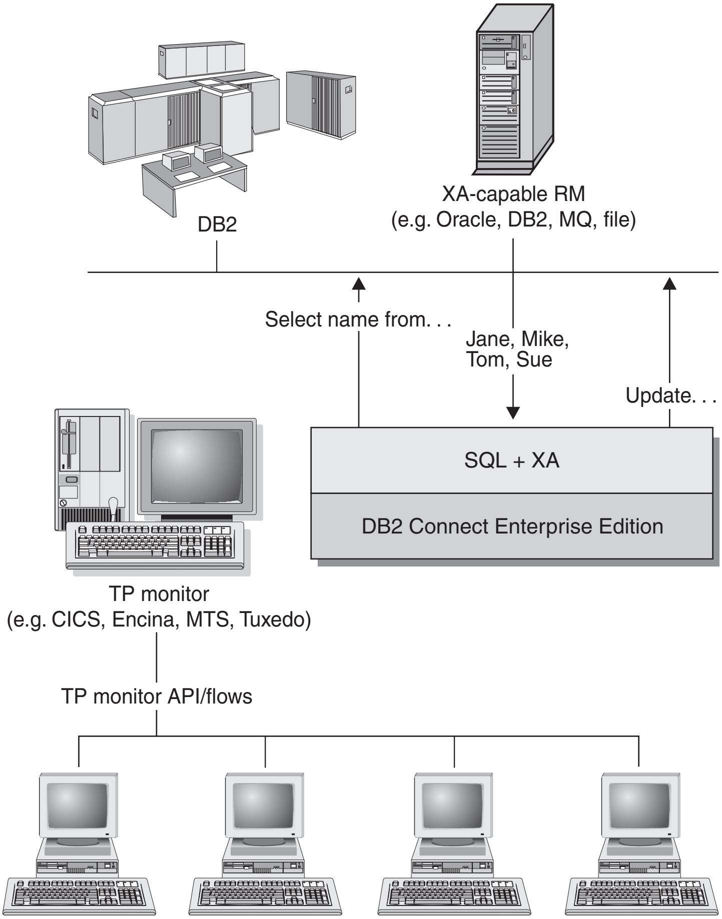
Repetitive
Most interactions with the computer tend to be the same process executed over and over again. For example, entering an order or processing payments are used many times every day.

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

Shared Data
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 a minimum. DB2® Connect allows applications under the control of an application server running on UNIX, Windows® NT, or Windows 2000 to execute transactions against remote LAN, host, and iSeries™ database servers and have these transactions coordinated by a TP monitor.



In this figure, the APIs, as well as the connectivity mechanism between the application server and the back-end database servers, are provided by DB2 Connect Enterprise Edition.

Examples of transaction processing monitors:

The most common TP monitors on the market today are:

- IBM® WebSphere® Application Server
- IBM TxSeries CICS®
- IBM TxSeries Encina® Monitor
- BEA Tuxedo
- BEA WebLogic
- Microsoft® Transaction Server

Remote iSeries, zSeries, and LAN database servers can be use within transactions coordinated by these TP monitors.

Tuxedo and DB2 Connect:

With DB2 Connect™ Version 6 and earlier versions, Tuxedo based applications were limited to read only access to host and iSeries database servers. This restriction has been removed. Tuxedo based applications can now update host and iSeries database servers within a Tuxedo coordinated transaction. Special configuration requirements and restrictions apply.

X/Open Distributed Transaction Processing (DTP) model:

An application executing business logic may 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 Universal Database™ for OS/390® and 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 above listed TP monitors are all XA compliant TMs. Remote host, iSeries, and DB2 UDB LAN-based database servers, when accessed via DB2 Connect, are XA compliant RMs. Therefore, any TP monitor which has an XA compliant TM can use host, iSeries, and LAN based DB2 UDB database servers within business applications executing transactions.

Related concepts:

- "X/Open distributed transaction processing model" in the *Administration Guide: Planning*
- "DB2 Connect" on page 3
- "DB2 Connect Enterprise Edition as a connectivity server" on page 23
- "DB2 Connect and web applications" on page 25
- "DB2 Connect and application servers" on page 31
- "Configuration considerations for XA transaction managers" in the *Administration Guide: Planning*

- "XA function supported by DB2 UDB" in the *Administration Guide: Planning*
- "Configuring DB2 Connect with an XA compliant transaction manager" on page 74

Related tasks:

- "Updating host or iSeries database servers with an XA-compliant transaction manager" in the *Administration Guide: Planning*

Related reference:

- "DB2 Connect security considerations for DB2 for OS/390 and z/OS" on page 189

Chapter 4. Programming in a DB2 Connect Environment

Programming in a Host or iSeries Environment

Applications in Host or iSeries Environments

DB2® Connect lets an application program access data in DB2 databases on System/390, zSeries, iSeries™ servers. For example, an application running on Windows® can access data in a DB2 Universal Database for OS/390 and z/OS database. You can create new applications, or modify existing applications to run in a host or iSeries environment. You can also develop applications in one environment and port them to another.

DB2 Connect™ enables you to use the following APIs with host database products such as DB2 Universal Database for OS/390 and z/OS, as long as the item is supported by the host database product:

- Embedded SQL, both static and dynamic
- The DB2 Call Level Interface
- The Microsoft® ODBC API
- JDBC

Some SQL statements differ among relational database products. You may encounter SQL statements that are:

- The same for all the database products that you use regardless of standards
- Available in all IBM® relational database products (see your SQL reference information for details)
- Unique to one database system that you access.

SQL statements in the first two categories are highly portable, but those in the third category will first require changes. In general, SQL statements in Data Definition Language (DDL) are not as portable as those in Data Manipulation Language (DML).

DB2 Connect accepts some SQL statements that are not supported by DB2 Universal Database. DB2 Connect passes these statements on to the host or iSeries server. For information on limits on different platforms, such as the maximum column length, see the topic on SQL limits.

If you move a CICS® application from OS/390® or VSE to run under another CICS product (for example, CICS for AIX), it can also access the OS/390 or

VSE database using DB2 Connect. Refer to the *CICS/6000 Application Programming Guide* and the *CICS Customization and Operation* manual for more details.

Note: You can use DB2 Connect with a DB2 Universal Database Version 8 database, although all you need is a DB2 client. Most of the incompatibility issues listed in the following topics will not apply if you are using DB2 Connect against a DB2 Universal Database Version 8 database, except in cases where a restriction is due to a limitation of DB2 Connect itself.

Related tasks:

- "Creating the sample Database on Host or AS/400 and iSeries Servers" in the *Application Development Guide: Building and Running Applications*

Related reference:

- "SQL limits" in the *SQL Reference, Volume 1*

Data Definition Language in Host and iSeries Environments

DDL statements differ among the IBM® database products because storage is handled differently on different systems. On host or iSeries™ server systems, there can be several steps between designing a database and issuing a CREATE TABLE statement. For example, a series of statements may translate the design of logical objects into the physical representation of those objects in storage.

The precompiler passes many such DDL statements to the host or iSeries server when you precompile to a host or iSeries server database. The same statements would not precompile against a database on the system where the application is running. For example, in an Windows® application the CREATE STORGROUP statement will precompile successfully to a DB2 Universal Database for OS/390 and z/OS database, but not to a DB2® for Windows database.

Data Manipulation Language in Host and iSeries Environments

In general, DML statements are highly portable. SELECT, INSERT, UPDATE, and DELETE statements are similar across the IBM® relational database products. Most applications primarily use DML SQL statements, which are supported by DB2® Connect.

Following are the considerations for using DML in host and iSeries™ environments:

- Numeric data types

When numeric data is transferred to DB2 Universal Database, the data type may change. Numeric and zoned decimal SQLTYPEs, supported by OS/400, are converted to fixed (packed) decimal SQLTYPEs.

- Mixed-byte data

  Mixed-byte data can consist of characters from an extended UNIX® code (EUC) character set, a double-byte character set (DBCS) and a single-byte character set (SBCS) in the same column. On systems that store data in EBCDIC (OS/390, z/OS, OS/400, VSE, and VM), shift-out and shift-in characters mark the start and end of double-byte data. On systems that store data in ASCII (such as UNIX), shift-in and shift-out characters are not required.

  If your application transfers mixed-byte data from an ASCII system to an EBCDIC system, be sure to allow enough room for the shift characters. For each switch from SBCS to DBCS data, add 2 bytes to your data length. For better portability, use variable-length strings in applications that use mixed-byte data.

- Long fields

  Long fields (strings longer than 254 characters) are handled differently on different systems. A host or iSeries server may support only a subset of scalar functions for long fields; for example, DB2 Universal Database for OS/390 and z/OS allows only the **LENGTH** and **SUBSTR** functions for long fields. Also, a host or iSeries server may require different handling for certain SQL statements; for example, DB2 for VSE & VM requires that with the INSERT statement, only a host variable, the SQLDA, or a NULL value be used.

- Large object data type

  The LOB data type is supported by DB2 Connect.

- User-defined types

  Only user-defined distinct types are supported by DB2 Connect. Structured types, also known as abstract data types, are not supported by DB2 Connect.

- ROWID data type

  The ROWID data type is handled by DB2 Connect as VARCHAR for bit data.

- BIGINT data type

  Eight byte (64-bit) integers are supported by DB2 Connect. The BIGINT internal data type is used to provide support for the cardinality of very large databases, while retaining data precision.

Data Control Language in Host and iSeries Environments

Each IBM® relational database management system provides different levels of granularity for the GRANT and REVOKE SQL statements. Check the product-specific publications to verify the appropriate SQL statements to use for each database management system.

Database Connection Management with DB2 Connect

DB2® Connect supports the CONNECT TO and CONNECT RESET versions of the CONNECT statement, as well as CONNECT with no parameters. If an application calls an SQL statement without first performing an explicit CONNECT TO statement, an *implicit* connect is performed to the default application server (if one is defined).

When you connect to a database, information identifying the relational database management system is returned in the SQLERRP field of the SQLCA. If the application server is an IBM® relational database, the first three bytes of SQLERRP contain one of the following:

DSN	DB2 Universal Database for OS/390 and z/OS
ARI	DB2 for VSE & VM
QSQ	DB2 UDB for iSeries™
SQL	DB2 Universal Database.

If you issue a CONNECT TO or null CONNECT statement while using DB2 Connect, the terrritory code or territory token in the SQLERRMC field of the SQLCA is returned as blanks; the CCSID of the application server is returned in the code page or code set token.

You can explicitly disconnect by using the CONNECT RESET statement (for type 1 connect), the RELEASE and COMMIT statements (for type 2 connect), or the DISCONNECT statement (either type of connect, but not in a TP monitor environment).

Note: An application can receive SQLCODEs indicating errors and still end normally; DB2 Connect™ commits the data in this case. If you do not want the data to be committed, you must issue a ROLLBACK command.

The FORCE command lets you disconnect selected users or all users from the database. This is supported for host and iSeries server databases; the user can be forced off the DB2 Connect workstation.

Related reference:

- “CONNECT (Type 1) statement” in the *SQL Reference, Volume 2*
- “CONNECT (Type 2) statement” in the *SQL Reference, Volume 2*

Processing of Interrrupt Requests

DB2® Connect handles an interrupt request from a DB2 client in one of two ways:

- If the keyword INTERRUPT_ENABLED exists in the PARMS field of the DCS catalog entry, DB2 Connect™ will drop the connection to the host or iSeries™ server on receipt of an interrupt request. The loss of connection, at least on DB2 UDB for OS/390® and z/OS™ servers, will cause the current request to be interrupted at the server.
- If the keyword INTERRUPT_ENABLED does not exist in the PARMS field of the DCS catalog entry, interrupt requests are ignored.

Package Attribute Differences among IBM Relational Database Systems

A package has the following attributes:

Collection ID
The ID of the package. It can be specified on the PREP command.

Owner
The authorization ID of the package owner. It can be specified on the PREP or BIND command.

Creator
The user name that binds the package.

Qualifier
The implicit qualifier for objects in the package. It can be specified on the PREP or BIND command.

Each host or iSeries™ server system has limitations on the use of these attributes:

DB2 Universal Database for OS/390 and z/OS
All four attributes can be different. The use of a different qualifier requires special administrative privileges. For more information on the conditions concerning the usage of these attributes, refer to the *Command Reference* for DB2 Universal Database for OS/390 and z/OS.

DB2 for VSE & VM
All of the attributes must be identical. If USER1 creates a bind file (with PREP), and USER2 performs the actual bind, USER2 needs DBA authority to bind for USER1. Only USER1’s user name is used for the attributes.

DB2® UDB for iSeries
The qualifier indicates the collection name. The relationship between

qualifiers and ownership affects the granting and revoking of privileges on the object. The user name that is logged on is the creator and owner unless it is qualified by a collection ID, in which case the collection ID is the owner. The collection ID must already exist before it is used as a qualifier.

DB2 Universal Database

All four attributes can be different. The use of a different owner requires administrative authority and the binder must have CREATEIN privilege on the schema (if it already exists).

CNULREQD BIND Option for C Null-Terminated Strings

The CNULREQD bind option overrides the handling of null-terminated strings that are specified using the LANGLEVEL option.

By default, CNULREQD is set to YES. This causes null-terminated strings to be interpreted according to MIA standards. If connecting to a DB2 Universal Database for OS/390 and z/OS server, it is strongly recommended that you set CNULREQD to YES. You need to bind applications coded to SAA1 standards (with respect to null-terminated strings) with the CNULREQD option set to NO. Otherwise, null-terminated strings will be interpreted according to MIA standards, even if they are prepared using LANGLEVEL set to SAA1.

Related concepts:

- "Null-Terminated Strings in C and C++" in the *Application Development Guide: Programming Client Applications*

Standalone SQLCODE and SQLSTATE Variables

Standalone SQLCODE and SQLSTATE variables, as defined in ISO/ANS SQL92, are supported through the LANGLEVEL SQL92E precompile option. An SQL0020W warning will be issued at precompile time, indicating that LANGLEVEL is not supported. This warning applies only to the features listed under LANGLEVEL MIA, which is a subset of LANGLEVEL SQL92E.

Related reference:

- "PRECOMPILE" in the *Command Reference*

User-Defined Sort Orders

The differences between EBCDIC and ASCII cause differences in sort orders in the various database products, and also affect ORDER BY and GROUP BY clauses. One way to minimize these differences is to create a user-defined collating sequence that mimics the EBCDIC sort order. You can specify a collating sequence only when you create a new database.

Note: Database tables can now be stored on DB2 Universal Database for OS/390 and z/OS in ASCII format. This permits faster exchange of data between DB2 Connect and DB2 Universal Database for OS/390 and z/OS, and removes the need to provide field procedures which must otherwise be used to convert data and resequence it.

Referential Integrity Differences among IBM Relational Database Systems

Different systems handle referential constraints differently:

DB2 Universal Database for OS/390 and z/OS
An index must be created on a primary key before a foreign key can be created using the primary key. Tables can reference themselves.

DB2 for VSE & VM
An index is automatically created for a foreign key. Tables cannot reference themselves.

DB2® UDB for iSeries™
An index is automatically created for a foreign key. Tables can reference themselves.

DB2 Universal Database
For DB2 Universal Database databases, an index is automatically created for a unique constraint, including a primary key. Tables can reference themselves.

Other rules vary concerning levels of cascade.

Locking and Application Portability

The way in which the database server performs locking can affect some applications. For example, applications designed around row-level locking and the isolation level of cursor stability are not directly portable to systems that perform page-level locking. Because of these underlying differences, applications may need to be adjusted.

The DB2 Universal Database for OS/390 and z/OS and DB2 Universal Database products have the ability to time-out a lock and send an error return code to waiting applications.

SQLCODE and SQLSTATE Differences among IBM Relational Database Systems

Different IBM® relational database products do not always produce the same SQLCODEs for similar errors. You can handle this problem in either of two ways:

- Use the SQLSTATE instead of the SQLCODE for a particular error.

SQLSTATEs have approximately the same meaning across the database products, and the products produce SQLSTATEs that correspond to the SQLCODEs.

- Map the SQLCODEs from one system to another system.

  By default, DB2® Connect maps SQLCODEs and tokens from each IBM host or iSeries™ server system to your DB2 Universal Database system. You can specify your own SQLCODE mapping file if you want to override the default mapping or you are using a database server that does not have SQLCODE mapping (a non-IBM database server). You can also turn off SQLCODE mapping.

Related concepts:

- "SQLCODE mapping" on page 77

System Catalog Differences among IBM Relational Database Systems

The system catalogs vary across the IBM® database products. Many differences can be masked by the use of views. For information, see the documentation for the database server that you are using.

The catalog functions in CLI avoid this problem by presenting support of the same API and result sets for catalog queries across the DB2® family.

Related concepts:

- "Catalog Functions for Querying System Catalog Information in CLI Applications" in the *CLI Guide and Reference, Volume 1*

Numeric Conversion Overflows on Retrieval Assignments

Numeric conversion overflows on retrieval assignments may be handled differently by different IBM® relational database products. For example, consider fetching a float column into an integer host variable from DB2 Universal Database for OS/390 and z/OS and from DB2 Universal Database. When converting the float value to an integer value, a conversion overflow may occur. By default, DB2 Universal Database for OS/390 and z/OS will return a warning SQLCODE and a null value to the application. In contrast, DB2 Universal Database will return a conversion overflow error. It is recommended that applications avoid numeric conversion overflows on retrieval assignments by fetching into appropriately sized host variables.

Isolation Levels Supported by DB2 Connect

DB2 Connect accepts the following isolation levels when you prep or bind an application:

RR Repeatable Read

RS Read Stability

CS Cursor Stability

UR Uncommitted Read

NC No Commit

The isolation levels are listed in order from most protection to least protection. If the host or iSeries™ server does not support the isolation level that you specify, the next higher supported level is used.

The following table shows the result of each isolation level on each host or iSeries application server.

Table 1. Isolation Levels

DB2 Connect	DB2 Universal Database for OS/390 and z/OS	DB2 for VSE & VM	DB2® UDB for iSeries	DB2 Universal Database
RR	RR	RR	note 1	RR
RS	note 2	RR	COMMIT(*ALL)	RS
CS	CS	CS	COMMIT(*CS)	CS
UR	note 3	CS	COMMIT(*CHG)	UR
NC	note 4	note 5	COMMIT(*NONE)	UR

Notes:

1. There is no equivalent COMMIT option on DB2 UDB for iSeries that matches RR. DB2 UDB for iSeries support RR by locking the whole table.
2. Results in RR for Version 3.1, and results in RS for Version 4.1 with APAR PN75407 or Version 5.1.
3. Results in CS for Version 3.1, and results in UR for Version 4.1 or Version 5.1.
4. Results in CS for Version 3.1, and results in UR for Version 4.1 with APAR PN60988 or Version 5.1.
5. Isolation level NC is not supported with DB2 for VSE & VM.

With DB2 UDB for iSeries, you can access an unjournalled table if an application is bound with an isolation level of UR and blocking set to ALL, or if the isolation level is set to NC.

Stored Procedures in Host or iSeries Environments

The considerations for stored procedures in host and iSeries™ environments are as follows:

- Invocation

  A client program can invoke a server program by issuing an SQL CALL statement. Each server works a little differently to the other servers in this case.

z/OS™ and OS/390®

The schema name must be no more than 8 bytes long, the procedure name must be no more than 18 bytes long, and the stored procedure must be defined in the SYSIBM.SYSPROCEDURES catalog on the server.

VSE or VM

The procedure name must not be more than 18 bytes long and must be defined in the SYSTEM.SYSROUTINES catalog on the server.

OS/400®

The procedure name must be an SQL identifier. You can also use the DECLARE PROCEDURE or CREATE PROCEDURE statements to specify the actual path name (the schema-name or collection-name) to locate the stored procedure.

All CALL statements to DB2® UDB for iSeries from REXX/SQL must be dynamically prepared and executed by the application, as the CALL statement implemented in REXX/SQL maps to CALL USING DESCRIPTOR.

You can invoke the server program on DB2 Universal Database with the same parameter convention that server programs use on DB2 Universal Database for OS/390 and z/OS, DB2 UDB for iSeries or DB2 for VSE & VM. For more information on the parameter convention on other platforms, refer to the DB2 product documentation for that platform.

All the SQL statements in a stored procedure are executed as part of the SQL unit of work started by the client SQL program.

- Do not pass indicator values with special meaning to or from stored procedures.

  Between DB2 Universal Database, the systems pass whatever you put into the indicator variables. However, when using DB2 Connect, you can only pass 0, -1, and -128 in the indicator variables.
- You should define a parameter to return any error or warning encountered by the server application.

  A server program on DB2 Universal Database can update the SQLCA to return any error or warning, but a stored procedure on DB2 Universal Database for OS/390 and z/OS or DB2 UDB for iSeries has no such support. If you want to return an error code from your stored procedure, you must pass it as a parameter. The SQLCODE and SQLCA is only set by the server for system detected errors.
- DB2 for VSE & VM Version 7 or higher, DB2 Universal Database for OS/390 and z/OS Version 5.1 or higher, DB2 for AS/400® V5R1, and DB2 for iSeries Version 7 or higher are the only host or iSeries application servers that can return the result sets of stored procedures at this time.

Related concepts:

- "DB2 Stored Procedures" in the *Application Development Guide: Programming Client Applications*

Related reference:

- "CALL statement" in the *SQL Reference, Volume 2*

DB2 Connect Support for Compound SQL

Compound SQL allows multiple SQL statements to be grouped into a single executable block. This may reduce network overhead and improve response time.

With NOT ATOMIC compound SQL, processing of compound SQL continues following an error. With ATOMIC compound SQL, an error rolls back the entire group of compound SQL.

Statements will continue execution until terminated by the application server. In general, execution of the compound SQL statement will be stopped only in the case of serious errors.

NOT ATOMIC compound SQL can be used with all of the supported host or iSeries™ application servers. ATOMIC compound SQL can be used with supported host application servers.

If multiple SQL errors occur, the SQLSTATEs of the first seven failing statements are returned in the SQLERRMC field of the SQLCA with a message that multiple errors occurred.

Related reference:

- "SQLCA" in the *Administrative API Reference*

Multisite Update with DB2 Connect

DB2® Connect allows you to perform a multisite update, also known as two-phase commit. A multisite update is an update of multiple databases within a single distributed unit of work (DUOW). Whether you can use this capability depends on several factors:

- Your application program must be precompiled with the CONNECT 2 and SYNCPOINT TWOPHASE options.
- If you have SNA network connections, you can use two-phase commit support provided by the sync point manager (SPM) function of DB2 Connect™ Enterprise Edition on AIX, and Windows® NT. This feature enables the following host database servers to participate in a distributed unit of work:

- DB2 for AS/400® Version 3.1 or later
- DB2 UDB for iSeries™ Version 5.1 or later
- DB2 for OS/390® Version 5.1 or later
- DB2 UDB for OS/390 and z/OS™ Version 7 or later
- DB2 for VM & VSE Version V5.1 or later.

The above is true for native DB2 UDB applications and applications coordinated by an external TP monitor such as IBM® TXSeries, CICS® for Open Systems, BEA Tuxedo, Encina® Monitor, and Microsoft® Transaction Server.

- If you have TCP/IP network connections, then a DB2 for OS/390 V5.1 or later server can participate in a distributed unit of work. If the application is controlled by a Transaction Processing Monitor such as IBM TXSeries, CICS for Open Systems, Encina Monitor, or Microsoft Transaction Server, then you must use SPM.

  If a common DB2 Connect Enterprise Edition server is used by both native DB2 applications and TP monitor applications to access host data over TCP/IP connections, the sync point manager must be used.

  If a single DB2 Connect Enterprise Edition server is used to access host data using both SNA and TCP/IP network protocols and two-phase commit is required, you must use SPM. This is true for both DB2 applications and TP monitor applications.

Related concepts:

- "XA function supported by DB2 UDB" in the *Administration Guide: Planning*
- "Configuring DB2 Connect with an XA compliant transaction manager" on page 74

Related tasks:

- "Configuring BEA Tuxedo" in the *Administration Guide: Planning*
- "Updating host or iSeries database servers with an XA-compliant transaction manager" in the *Administration Guide: Planning*

Host and iSeries Server SQL Statements Supported by DB2 Connect

The following statements compile successfully for host and iSeries™ server processing, but not for processing with DB2 Universal Database systems:

- ACQUIRE
- DECLARE (modifier.(qualifier.)table_name TABLE ...
- LABEL ON

These statements are also supported by the command line processor.

The following statements are supported for host and iSeries server processing but are not added to the bind file or the package and are not supported by the command line processor:

- DESCRIBE statement_name INTO descriptor_name USING NAMES
- PREPARE statement_name INTO descriptor_name USING NAMES FROM ...

The precompiler makes the following assumptions:

- Host variables are input variables
- The statement is assigned a unique section number.

Host and iSeries Server SQL Statements Rejected by DB2 Connect

The following SQL statements are not supported by DB2® Connect and not supported by the command line processor:

- COMMIT WORK RELEASE
- DECLARE state_name, statement_name STATEMENT
- DESCRIBE statement_name INTO descriptor_name USING xxxx (where xxxx is ANY, BOTH, or LABELS)
- PREPARE statement_name INTO descriptor_name USING xxxx FROM :host_variable (where xxxx is ANY, BOTH, or LABELS)
- PUT ...
- ROLLBACK WORK RELEASE
- SET :host_variable = CURRENT ...

DB2 for VSE & VM extended dynamic SQL statements are rejected with -104 and syntax error SQLCODEs.

Part 2. Reference

Chapter 5. Updating database directories

Updating database directories

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

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

Notes:

1. Before updating these directories, you should configure communications on the host or iSeries database server and workstations.
2. Database directories can be updated using the Configuration Assistant (CA).
3. This topic assumes that you are *not* using DCE Directory Services.

Procedure:

To update database directories:

1. Collect database directory information using the directory customization worksheet
2. Update the directories with information about remote database server machines

Related tasks:

- "Updating the directories with information about remote database server machines" in the *Administration Guide: Implementation*

Related reference:

- "LIST DATABASE DIRECTORY" in the *Command Reference*
- "LIST NODE DIRECTORY" in the *Command Reference*
- "LIST DCS DIRECTORY" in the *Command Reference*
- "Directory customization worksheet" on page 63

System database directory values

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 host or iSeries™ 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 done for connections originating from the DB2® Connect server. The valid options are: `SERVER`, `SERVER_ENCRYPT`, `CLIENT`, and `DCE`.

Related concepts:

- "Updating database directories" on page 55

Node directory values

You can specify the following information in the node directory:

Node name
A nickname for the host or iSeries™ 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
Can be `APPC` or `TCPIP`.

Symbolic destination name
When defining an APPC node, use the symbolic destination name that was specified in the CPI Communications Side Information Table (for example, the name of the CPI-C Symbolic Destination Properties when

using Microsoft® SNA Server). You should get this value from the person who either installed and/or configured SNA. The symbolic destination name is case sensitive (you may encounter an SQL1338 return code if there is a mismatch between upper and lower case names).

Security type

The type of security checking that will be done. For APPC nodes, the valid options are `SAME`, `PROGRAM`, and `NONE`. 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 OS/390® and z/OS™ remote hosts, the hostname appears in the DSNL004I message (DOMAIN=hostname) when the Distributed Data Facility (DDF) is started.

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 OS/390 and 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.

Note: A second port used for two-phase commit resynchronization operations over TCP/IP connections is assigned by the server. For example, the DB2 Universal Database for OS/390 and z/OS bootstrap dataset assigns a port number (RESPORT) to be used for resynchronization for inbound connections to DB2 Universal Database for OS/390 and z/OS only. No service name need be defined for this.

Related concepts:

- "Updating database directories" on page 55
- "Security types supported with DB2 Connect" on page 192

DCS directory values

You can specify the following information in the DCS directory:

Database name

A user-defined nickname for the host or iSeries™ 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 host or iSeries database server system, as follows:

OS/390® and z/OS™

A DB2 Universal Database for OS/390 and z/OS subsystem identified by its LOCATION NAME.

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
```

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.

VSE or VM

The database name (DBNAME)

OS/400® and z/OS

The relational database name (RDBNAME)

Other For Windows® NT, Windows 2000, and UNIX-based 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 (e.g. 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 host or iSeries database server database when one of the following SQLCODES is returned:

```
SQL30000N
SQL30040N
SQL30050N
SQL30051N
SQL30053N
SQL30060N
SQL30070N
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:

```
SQL30020N
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 host or iSeries 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, Windows NT, and Windows 2000.

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 host or iSeries database server, in order to process additional database requests. On platforms other than AIX® V4.1 and later, SNA Server V3.1 and later, Windows NT® and Windows 2000, 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 may kill the socket, but - depending on the server implementation - there may or may not be an outstanding receive. DB2 Universal Database for OS/390 and 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.

A new profile (environment or registry) variable has also been introduced, called DB2SYSPLEX_SERVER, and it can be used to disable the SYSPLEX support at the workstation level.

,,,,,,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 appc node nynode remote nycpic security program
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 quotes, the LOCALDATE value has to be specified inside two pairs of double quotes. Note the use of the operating system escape character ″\″ (backslash) to ensure that the double quotes 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 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 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 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 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.

Related concepts:

- "Updating database directories" on page 55

Directory customization worksheet

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

Node Directory Parameters:

Table 2. Node Directory Parameters

Parameter	Example	Your value
Node name	DB2NODE	
Symbolic destination name (APPC node)	DB2CPIC	
Remote hostname (TCP/IP node)	ZOSHOST	
Server (TCP/IP service name or port number)	db2inst1c (or 446)	
Security type	PROGRAM for APPC Nodes; NONE for TCP/IP nodes.	

Notes:

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

DCS Directory Parameters:

Table 3. 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 4. System Database Directory Parameters

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

Related concepts:

- "Updating database directories" on page 55
- "System database directory values" on page 56
- "Node directory values" on page 56
- "DCS directory values" on page 58

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 host or iSeries™ 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.

Related concepts:

- "Updating database directories" on page 55

Related reference:

- "Directory customization worksheet" on page 63

Handling BiDi data

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

- Numeral shape (ARABIC vs HINDI)
- Orientation (RIGHT-TO-LEFT vs LEFT-TO-RIGHT)
- Shaping (SHAPED vs UNSHAPED)
- Symmetric swapping (YES or NO)
- Text type (LOGICAL vs 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 OS/390® or z/OS™ data is usually in SHAPED VISUAL format. Therefore, without any support for BiDi attributes, data sent from DB2 for OS/390 and 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 or iSeries™ DB2 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 DB2 client running CCSID 62213 (BiDi string type 5) and you would like to access a DB2 host or iSeries database running CCSID 424 (BiDi string type 4). However, you know that the data contained in the DB2 host or iSeries database is instead based on CCSID 8616 (BiDi string type 6).

There are two problems in this situation. The first is that the DB2 host or iSeries database does not know the difference between the BiDi string types with CCSIDs 424 and 8616. The second problem is that the DB2 host or iSeries database does not recognize the DB2 client CCSID of 62213. It only supports CCSID 862, which is based on the same code page as CCSID 62213.

You will need to make sure that data sent to the DB2 host or iSeries 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 or iSeries database. You will use the following cataloging for the DB2 host or iSeries database:

```
catalog dcs database nydb1 as TELAVIV parms ",,,,,,,,BIDI=8616"
```

This tells DB2 Connect to override the DB2 host or iSeries database CCSID of 424 with 8616. This override includes the following processing:

1. DB2 Connect will connect to the DB2 host or iSeries database using CCSID 862.
2. DB2 Connect will perform BiDi layout transformation on data it is about to send to the DB2 host or iSeries database from CCSID 62213 (BiDi string type 5) to CCSID 62221 (BiDi string type 6).
3. DB2 Connect will perform BiDi layout transformation on data it receives from the DB2 host or iSeries database from CCSID 8616 (BiDi string type 6) to CCSID 62213 (BiDi string type 5).

Notes:

1. The environment variable or registry value DB2BIDI has to be set to YES in order for the BIDI parameter to take effect.
2. If you would like DB2 Connect to perform layout transformation on data it is about to send to the DB2 host or iSeries 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 or iSeries database CCSID.
3. In some cases, use of a bidirectional CCSID may 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 or iSeries 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:

Here are examples of some parameter strings you could specify.

For example, you could specify any of the following where ″\″ (backslash) is the operating system escape character:

On AIX:

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

On Windows NT, or Windows 2000:

```
NOMAP
d:\sqllib\map\dcs1new.map,D
,,INTERRUPT_ENABLED
NOMAP,D,INTERRUPT_ENABLED,,,SYSPLEX,LOCALDATE=\"\"YYMMDD\"\",,
```

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

Note: Because of the need to specify two pairs of double quotes when specifying the LOCALDATE mask in the parameter string, you must use the operating system escape character ″\″ (backslash), 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
```

Related concepts:

- “Bidirectional support with DB2 Connect” in the *Administration Guide: Planning*

Related tasks:

- “Enabling bidirectional support” in the *Administration Guide: Planning*

Related reference:

- “Bidirectional-specific CCSIDs” in the *Administration Guide: Planning*

Chapter 6. Multisite updates

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. 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.

DB2® products provide comprehensive support for multisite updates. This support is available for applications developed using regular SQL as well as 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 and Encina), 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.

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`, `Encina® Abort()`, `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 and non-DB2 resource managers 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 distributed unit of work. Currently, the following DB2 servers provided DUOW support that enabled them to participate in distributed transactions:

- DB2 UDB for UNIX® and Windows® Version 5 or later
- DB2 for OS/390® Version 5.1
- DB2 UDB for OS/390 Version 6.1 or later
- DB2 for z/OS™ Version 7
- DB2 UDB for iSeries™ Version 4 or later
- DB2 Server for VM and VSE V5.1 or later (SNA only)

A distributed transaction can update any mix of supported database servers. For example, your application can update several tables in DB2 UDB on Windows NT or Windows 2000, a DB2 for OS/390 and z/OS database, and a DB2 UDB for iSeries database, all within a single transaction.

Related concepts:

- "Remote unit of work" on page 18
- "Distributed requests" on page 19
- "Multisite update and sync point manager" on page 72

Related tasks:

- "Enabling Multisite Updates using the Control Center" on page 70
- "Testing Multisite Update using the Control Center" on page 71

Enabling Multisite Updates using the Control Center

You can use the Control Center to provide multisite updates.

Procedure:

To enable multistie updates:

1. Launch the Multisite Update Wizard. From the Control Center.
2. Click the [+] sign to expand the tree view.
3. With the right mouse button, select the instance that you wish to configure. A pop-up menu opens.
4. Select **Multisite Update —> Configure** menu item.
5. The Multisite Update Wizard provides a notebook-type interface. Each page of the wizard will prompt you for certain information about your configuration.

a. Specify a Transaction Processor (TP) monitor. This field will show the defaults for the TP monitor you have enabled. If you do not want to use a TP monitor, select **Do Not Use a TP Monitor**. Click **Next**.
b. Specify the communications protocols you will use. Click **Next**.
c. Specify a Transaction Manager database. This panel defaults to the first database you connect to (1ST_CONN). You can leave this default or select another catalogued database. Click **Next**.
d. Specify the types of database servers involved in the update, and also whether or not TCP/IP is to be used exclusively.
e. Specify the sync point manager settings. This page will only appear if the settings on the previous page indicate that you need to use DB2's sync point manager in a multisite update scenario.

Related concepts:

- "Multisite Updates" on page 69

Related tasks:

- "Testing Multisite Update using the Control Center" on page 71

Testing Multisite Update using the Control Center

You can test your multisite update setup using the Control center.

Procedure:

To test multisite update:

1. Select the instance with the right mouse button and choose the **Multisite Update —> Test** menu option from the pop-up menu. The Test Multisite Update window opens.
2. Select the databases you want to test from the available databases in the **Available Databases** list box. You can use the arrow buttons (> and >>) in the middle to move selections to and from the **Selected databases** list box. You can also change the selected userid and password by directly editing them in the **Selected databases** list box.
3. When you have finished your selection, click **OK**. The Multisite Update Test Result window opens.
4. The Multisite Update Test Result window shows which of the databases you selected succeeded or failed the update test. The window will show SQL codes and error messages for those that failed. Click **Close** to close the window.
5. Click **Close** to close the Test Multisite Update window.

Related concepts:

- "Multisite Updates" on page 69

Related tasks:

- "Enabling Multisite Updates using the Control Center" on page 70

Multisite update and sync point manager

Host and iSeries™ database servers require DB2® Connect to participate in a distributed transaction originating from Windows, UNIX, and web applications. In addition, many of the multisite update scenarios that involve host and iSeries 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 (SNA or 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 host or iSeries from Intel or UNIX® machines. For multisite updates, the SPM component of DB2 Connect is required if the access is via SNA or if you are using a TP monitor.

Table 5. Multisite update scenarios that require SPM – TCP/IP

Transaction Processor Monitor Used?	Sync Point Manager Needed?	Product Required (Choose One)	Host and iSeries Database Supported
Yes	Yes	• DB2 Connect EE • DB2 UDB ESE	• DB2 for OS/390® V5.1 • DB2 UDB for OS/390 V6.1 or later • DB2 UDB for z/OS™ V7 or later
No	No	• DB2 Connect PE • DB2 Connect EE • DB2 UDB ESE	• DB2 for OS/390 V5.1 • DB2 UDB for OS/390 V6.1 or later • DB2 UDB for z/OS V7 or later

Table 6. Multisite update scenarios that require SPM – SNA

Transaction Processor Monitor Used?	Sync Point Manager Needed?	Product Required (Choose One)	Host and iSeries Database Supported
Yes	Yes	• DB2 Connect EE* • DB2 UDB ESE* **Note:** *AIX, Windows® NT, and Windows 2000 platforms only.	• DB2 for OS/390 V5.1 • DB2 UDB for OS/390 V6.1 or later • DB2 UDB for z/OS V7 or later • DB2 for AS/400® V3.1 or later • DB2 UDB for iSeries V4 or later • DB2 Server for VM or VSE V5.1 or later
No	Yes	• DB2 Connect EE* • DB2 UDB ESE* **Note:** *AIX, Windows NT, and Windows 2000 platforms only.	• DB2 for OS/390 V5.1 • DB2 UDB for OS/390 V6.1 or later • DB2 UDB for z/OS V7 • DB2 for AS/400 V3.1 or later • DB2 UDB for iSeries V4 or later • DB2 Server for VM and VSE V5.1 or later

Note: A distributed transaction can update any mix of supported database servers. For example, your application can update several tables in DB2 UDB on Windows, a DB2 for OS/390 database and a DB2 UDB for iSeries database all within a single transaction.

Related concepts:

- "Multisite Updates" on page 69

Configuring DB2 Connect with an XA compliant transaction manager

This topic describes the configuration steps necessary to use S/390, iSeries, and zSeries™ database servers within your TP monitor.

Prerequisites:

You have an operational TP monitor and have installed DB2® Connect, as well as have configured and tested a connection to the host or iSeries™ database server.

Procedure:

There is no distinguishing between configuring for access to a LAN-based DB2 UDB database server versus a host or iSeries database server. The following instructions outline the general configuration steps for TP monitors not listed in the *Administration Guide*.

To configure DB2 Connect™ to use S/390, iSeries, and zSeries 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 DB2's XA_OPEN string. Each TP monitor has its own way to do this. For information on how to configure DB2's XA OPEN string 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. Host and iSeries database servers do not yet support the XA interface.

   The SPM is a component of DB2 Connect which maps the XA two phase commit protocol into the two phase commit protocol used by host and iSeries database servers. By default, the DB2 instance has pre-defined 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.

   If you are using TCP/IP to connect to DB2 for OS/390® and z/OS, then you should not have to change any of the default settings. In this case, there is no SPM configuration required since it is already operational. If you are using SNA to access host or iSeries database servers, then you have to ensure the SPM_NAME value represents a valid SNA LU in your network. If the default SPM_NAME value is not acceptable then you should use the Multisite Update Wizard to modify this value.

Related concepts:

- "DB2 Connect and transaction processing monitors" on page 34

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 DB2 for OS/390® Version 6 or later, or 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 OS/390 and z/OS.

This feature reduces the window where one branch of a distributed transaction encounters lock timeout or deadlock as a result of another branch within the same global transaction. DB2 for OS/390 and z/OS shares the lock space in this situation provided DB2 Connect™ sends the XID on each connection serving different branches of the same global transaction.

Chapter 7. SQLCODE mapping

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 may 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 host or iSeries™ database server to the appropriate DB2 Universal Database SQLCODEs.

If you want to turn off SQLCODE mapping, specify NOMAP in the parameter string of the DCS directory or the DCE routing information object.

If you port an application directly from a host or iSeries database server, such as DB2® UDB for OS/390® and z/OS, you might want to turn off SQLCODE mapping. This would let you use the application without changing the SQLCODEs that it references.

Related tasks:

- "Tailoring the SQLCODE mapping" on page 78

Turning off SQLCODE mapping

If you want to turn off SQLCODE mapping, specify NOMAP in the parameter string of the DCS directory or the DCE routing information object.

If you port an application directly from a host or iSeries database server, such as DB2 UDB for OS/390 and z/OS, you might want to turn off SQLCODE mapping. This would let you use the application without changing the SQLCODEs that it references.

Related tasks:

- "Tailoring the SQLCODE mapping" on page 78

Tailoring the SQLCODE mapping

By default, DB2 Connect maps SQLCODEs and tokens from each host or iSeries database server to the appropriate DB2 UDB SQLCODEs. The following files are copies of the default SQLCODE mapping:

- dcs1dsn.map maps DB2 UDB for OS/390 and z/OS SQLCODEs.
- dcs1ari.map maps DB2 for VSE & VM SQLCODEs.
- dcs1qsq.map maps DB2 UDB for iSeries SQLCODEs.

No mapping is required for UNIX-based DB2 systems.

Procedure:

If you want to override the default SQLCODE mapping or you are using a host or iSeries database server that does not have SQLCODE mapping (a non-IBM database server), you can copy one of these 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.

Specify the file name of your new SQLCODE mapping file in the parameter string of the DCS Directory or the DCE routing information object.

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.

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:

sqlcode The SQLCODE from the host or iSeries 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.

cc*nn* The SQLSTATE class code from the host or iSeries database server. *nn* is one of the following:

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:

(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 datatype 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 datatype 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 host or iSeries database server products support only CHAR tokens. The character **i** must be lowercase.

n A number or numbers indicating which host or iSeries database server tokens are used. They are arranged in the order desired for placement in the output SQLCA. The number indicates the host or iSeries database server token; the arrangement indicates the order in which the tokens will be placed in the SQLCA.

For example, the host or iSeries 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.

Figure 5 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 5. An SQLCODE Mapping File

Each mapping statement in the file is described as follows:

1. The SQLCODE is mapped from -007 to -007. The first input token received from the host or iSeries 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 host or iSeries 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 host or iSeries 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 host or iSeries 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 host or iSeries 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.

Related concepts:

- "SQLCODE mapping" on page 77

Related tasks:

- "Turning off SQLCODE mapping" on page 77

Chapter 8. Tuning CLI/ODBC application performance with the CLISCHEMA keyword

CLI/ODBC

CLI/ODBC is an SQL application programming interface that can be called by your database applications. It passes dynamic SQL statements as database function calls. Unlike embedded SQL it does not require host variables or a precompiler.

When an application program calls CLI/ODBC, the first thing that it must do is to make SQL calls to some of the system catalog tables on the target database in order to obtain information about other database contents. CLI/ODBC applications always access the system catalog tables in this way. There are ten API calls that may be made in order to gather information about the database that is being connected to. These API calls include:

- `SQLTables`
- `SQLColumns`
- `SQLSpecialcolumns`
- `SQLStatistics`
- `SQLPrimarykeys`
- `SQLForeignkeys`
- `SQLTablePrivileges`
- `SQLColumnPrivileges`
- `SQLProcedures`
- `SQLProcedureColumns.`

By default, when you connect to a database, your CLI/ODBC application will query the system catalog tables for information about *all* the database tables in that database. Especially on a large system this can result in a lot of network traffic and considerable delays when starting an application.

Related concepts:

- "Additional CLISCHEMA keyword hints and tips" on page 84
- "CLI/ODBC application performance tuning with the CLISCHEMA keyword" on page 88
- "The CLISCHEMA keyword" on page 85

Related tasks:

- "Calling Stored Procedures in CLI Applications" in the *CLI Guide and Reference, Volume 1*

Related reference:

- “SQLTables Function (CLI) - Get Table Information” in the *CLI Guide and Reference, Volume 2*

Additional CLISCHEMA keyword hints and tips

The CLISCHEMA keyword has to be added to the db2cli.ini file within a section for the DSN name or the common section. A section is text in square brackets. The COMMON section is indicated by the text ″COMMON″ in square brackets. Note that keywords and section names are not case-sensitive.

At connect, each possible keyword is checked for first under the DSN name, then if not found, under the COMMON section. This allows for both DSN specific keywords and global (client) keywords.

Also, the DBALIAS keyword can be used to create different DSN (ODBC Data Sources) that map to the same database. (A DSN name can be up to 255 characters in length, and it is mapped to the 8 char dbname).

In the example below, any time a user connects to TESTDB or any DSN that is not listed in the file, they will use clischema=ODBCCAT. If they connect to TestDBcat2, they will use clischema=odbccat2, but still connect to the testdb database.

Example db2cli.ini file:

```
[TESTDB]

[COMMON]
clischema=odbccat

[TestDBcat1]
DBALIAS=testdb
clischema=odbccat1

[TestDBcat2]
DBALIAS=testdb
clischema=odbccat2
```

Related concepts:

- “CLI/ODBC” on page 83
- “CLI/ODBC application performance tuning with the CLISCHEMA keyword” on page 88
- “The CLISCHEMA keyword” on page 85

The CLISCHEMA keyword

DB2® Universal Database provides several CLI/ODBC initialization keywords that 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. Changing ODBC/CLI settings for the database using the Client Configuration Assistant (on those platforms which support it).
3. Updating the database CLI configuration using the DBA Command Line Interface.

The keywords are:

```
- DBNAME
- TABLETYPE
- SCHEMALIST
- SYSSCHEMA
- CLISCHEMA
```

Usage notes:

The CLISCHEMA option indicates an alternative schema, tables, and index set to be searched instead of the SYSIBM (or SYSTEM, QSYS2) schemas when the DB2 CLI and ODBC Catalog Function calls are issued to obtain catalog information.

For example, if you specify CLISCHEMA='SERGE', the internal CLI/ODBC API calls that normally reference the system tables will reference the following user tables instead:

```
- SERGE.TABLES
- SERGE.COLUMNS
- SERGE.SPECIALCOLUMNS
- SERGE.TSTATISTICS
- SERGE.PRIMARYKEYS
- SERGE.FOREIGNKEYS
- SERGE.TABLEPRIVILEGES
- SERGE.COLUMNTABLES
- SERGE.PROCEDURES
- SERGE.PROCEDURESCOLUMNS.
```

These user tables must be built by the database administrator before CLISCHEMA can be used.

Note: DataPropagator™ provides support for CLISCHEMA, so that the Database Administrator can perform this task in three possible ways:

1. Using db2cli.exe on the client.
2. Automatically on the server using DataPropagator.

3. Manually on the server.

The information which follows explains how this task can be performed on the client.

Related concepts:

- "Additional CLISCHEMA keyword hints and tips" on page 84
- "CLI/ODBC" on page 83
- "CLI/ODBC application performance tuning with the CLISCHEMA keyword" on page 88

db2ocat catalog optimizer tool

A new tool `db2ocat` is provided on Windows® 32-bit operating systems to help you optimize system catalog searches for ODBC and JDBC applications.

You can obtain the `db2ocat` point-and-click catalog optimizer utility by downloading `db2ocat.exe` from:

```
ftp://ftp.software.ibm.com/ps/products/db2/tools.
```

Related concepts:

- "CLI/ODBC" on page 83
- "CLI/ODBC application performance tuning with the CLISCHEMA keyword" on page 88

db2cli and bldschem utilities

A utility to set up the user tables required by CLISCHEMA is provided in the form of the previously undocumented bldschem support command of the CLI Command Line Interface, which can be found as: /samples/cli/db2cli.exe.

For example, to build the set of user tables that is required to work with CLISCHEMA='SERGE' for the table name STAFF owned by the schema owner (creator) USERID, in the database SAMPLE, you would run the following command after issuing db2start and after registering the database to ODBC/CLI:

```
db2cli < addstaff.txt
```

Where "addstaff.txt" contains the following script:

```
opt callerror on
opt echo on
quickc 1 1 sample userid password
#
# Repeat next line for each table to add.
```

```
#
bldschem 1 SERGE USERID STAFF
#
# Exit
#
killenv 1
```

This script will result in the creation of the set of tables SERGE.* as listed above, with indexes, populated using the system catalog table data for the table USERID.STAFF. For example, SERGE.TABLES will be populated with a new row for each entry that is matched. Additional bldschem calls result in appends to the existing SERGE.* tables, with replacement of existing rows.

In summary, the syntax of the bldschem support command is:

```
bldschem <handle_number> <value_of_CLISCHEMA> <schema_owner> <table_name>
```

Where:

```
- <handle_number> should be 1

- <value_of_CLISCHEMA> should be the same as the schema name specified with
     the CLISCHEMA keyword

- <schema_owner> is the creator of the table

- <table_name> can be the name of a user table, a view, an alias, a synonym,
     or a system table name.
   (Wildcard characters are allowed).
```

If you subsequently run the following example through db2cli.exe, then you will append to the user tables SERGE.* created in the previous example, adding rows which reflect the data in the system catalog tables for each table for which FRED and BERT are the schema owners.

```
bldschem 1 SERGE FRED %
bldschem 1 SERGE BERT %
```

When the CLISCHEMA CLI/ODBC keyword is subsequently set to SERGE, processing by ODBC/CLI applications against the SAMPLE database will reference the SERGE.* set of tables instead of the system catalog tables.

Related concepts:

- “CLI/ODBC” on page 83
- “CLI/ODBC application performance tuning with the CLISCHEMA keyword” on page 88

CLI/ODBC application performance tuning with the CLISCHEMA keyword

This section will help you tune the performance of your ODBC/CLI applications using the CLISCHEMA initialization keyword. It does not contain general information about tuning network or database performance.

The information presented here is intended primarily for users of DB2® UDB for OS/390® and z/OS, and the target environment comprises:

- A CLI/ODBC application running with a DB2 Universal Database client
- DB2 Connect™ Version 6 or higher (Personal Edition or Enterprise Edition)
- DB2 Universal Database for OS/390 and z/OS Version 5.1 or higher.

In most production environments, the default search of the system catalog tables can return a very large amount of data, so that each time a CLI/ODBC application opens a database there can be a considerable delay. Even on a typical test database, the delay can easily be of the order of 25 seconds or so.

Measure this delay initially without having any of the above CLI keywords set, while remembering to discount connect time and especially the long delays that can occur when a DB2 client issues its first ever connection to a new database - autobinding can often take several minutes.

How you proceed next depends on the structure of your data and your organization. In some cases you can use DBNAME, SCHEMALIST, and TABLETYPE in combination to limit the search for use by a particular application or group of applications. For example, if production DBA clients usually access tables under a given DBNAME and schema then this is easy to specify.

CLISCHEMA provides the best performance advantages for most users. For this reason we would generally recommend using CLISCHEMA in a production environment, since it is much easier to set up and modify CLISCHEMA's user tables through the CLI command Line Interface (db2cli.exe) and its bldschem support command.

Related concepts:

- "Additional CLISCHEMA keyword hints and tips" on page 84
- "CLI/ODBC" on page 83
- "The CLISCHEMA keyword" on page 85

Related reference:

- "CLISCHEMA CLI/ODBC Configuration Keyword" in the *CLI Guide and Reference, Volume 1*

Chapter 9. Binding applications and utilities

Binding applications and utilities (DB2 Connect)

Application programs developed using embedded SQL must be bound to each database with which they will operate. On platforms where these functions are available, you can do this using the Command Center and the Configuration Assistant.

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 simply a process of processing these bind files by a host or iSeries™ database server. For more information about binding, refer to the *Application Development Guide*.

Because several of the utilities supplied with DB2 Connect are developed using embedded SQL, they must be bound to a host or iSeries 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 host or iSeries database servers. The lists of bind files required by these utilities are contained in the following files:

- ddcsmvs.lst for OS/390® or z/OS™
- ddcsvse.lst for VSE
- ddcsvm.lst for VM
- ddcs400.lst for OS/400®

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

If DB2 Connect Enterprise Edition is installed, the DB2 Connect utilities must be bound to each host or iSeries database server; once from each type of client platform, before they can be used with that system.

For example, if you have 10 Windows® clients, and 10 AIX® clients connecting to DB2® UDB for OS/390 and z/OS via a DB2 Connect Enterprise Edition for Window NT server, do the following:

1. Bind ddcsmvs.lst from one of the Windows clients.
2. Bind ddcsmvs.lst from one of the AIX clients.
3. Bind ddcsmvs.lst from the DB2 Connect server.

Note: This assumes all the clients are at the same service level. If they are not then, in addition, you may need to bind from each client of a particular service level.

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 host or iSeries database server that you are binding to, do the following:

1. Make sure that you have sufficient authority for your host or iSeries database server management system:

   OS/390 or z/OS

   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 above 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

On the VSE or VM system, you can issue:

```
grant select on table to nullid with grant option
```

OS/400

*CHANGE authority or higher on the NULLID collection.

2. Issue commands similar to the following:

```
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 host or iSeries database server, ddcsmvs.lst is the bind list file for MVS, 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 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
```

For your reference, Table 7 shows the bind files and package names that are used by different components of DB2 Connect. In some cases, different bind files and packages are used on different operating systems.

Table 7. Bind Files and Packages

Component	**Bind file**	**Package**	**OS/390 or z/OS**	**VSE**	**VM**	**OS/400**
Binder (used by the GRANT bind option)	db2ajgrt.bnd	sqlab*xxx*	yes	yes	yes	yes
DB2 Call Level Interface						
Isolation level CS	db2clics.bnd	sqll1*xxx*	yes	yes	yes	yes
Isolation level RR	db2clirr.bnd	sqll2*xxx*	yes	yes	yes	yes
Isolation level UR	db2cliur.bnd	sqll3*xxx*	yes	yes	yes	yes
Isolation level RS	db2clirs.bnd	sqll4*xxx*	yes	yes	yes	yes
Isolation level NC	db2clinc.bnd	sqll5*xxx*	no	no	no	yes
Using OS/400 table names (OS/400 3.1 or later)	db2clias.bnd	sqlla*xxx*	no	no	no	yes
Using VSE/VM table names	db2clivm.bnd	sqll8*xxx*	no	yes	yes	no
Command Line Processor						
Isolation level CS	db2clpcs.bnd	sqlc2*xxx*	yes	yes	yes	yes
Isolation level RR	db2clprr.bnd	sqlc3*xxx*	yes	yes	yes	yes
Isolation level UR	db2clpur.bnd	sqlc4*xxx*	yes	yes	yes	yes
Isolation level RS	db2clprs.bnd	sqlc5*xxx*	yes	yes	yes	yes
Isolation level NC	db2clpnc.bnd	sqlc6*xxx*	no	no	no	yes
REXX						
Isolation level CS	db2arxcs.bnd	sqla1*xxx*	yes	yes	yes	yes
Isolation level RR	db2arxrr.bnd	sqla2*xxx*	yes	yes	yes	yes
Isolation level UR	db2arxur.bnd	sqla3*xxx*	yes	yes	yes	yes
Isolation level RS	db2arxrs.bnd	sqla4*xxx*	yes	yes	yes	yes
Isolation level NC	db2arxnc.bnd	sqla5*xxx*	no	no	no	yes
Utilities						

Table 7. Bind Files and Packages (continued)

Component	Bind file	Package	OS/390 or z/OS	VSE	VM	OS/400
Export	db2uexpm.bnd	sqlub*xxx*	yes	yes	yes	yes
Import	db2uimpm.bnd	sqluf*xxx*	yes	yes	yes	yes

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
```

Notes:

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. 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 may flag the SQL syntax of another to be invalid. For this reason, binding any of the list files `ddcsxxx.lst` against any particular host or iSeries database server should be expected to produce some warnings. For example, when binding against DB2 for VM, numerous warning messages may result since DB2 for VM does not permit cursors to be declared as `"WITH HOLD"`.

b. If you are connecting to a DB2 Universal Database database through DB2 Connect, use the bind list `db2ubind.lst` and do not specify `sqlerror continue`, which is only valid when connecting to a host or iSeries database server. Also, to connect to a DB2 Universal Database database, we recommend 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 may need to bind the utilities on these clients to DB2 Connect.

Related reference:

- "BIND" in the *Command Reference*
- "REBIND" in the *Command Reference*
- "db2rbind - Rebind all Packages" in the *Command Reference*

Chapter 10. Database system monitor

Monitoring connections for remote clients

You can use the database system monitor with 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 host or iSeries™ 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 logical unit of work identifier (LUWID), for two-phase conversations protected by an SNA sync point manager (SPM).
- 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.

Related concepts:

- "Monitoring performance using the Windows Performance Monitor" on page 96
- "System monitor switches" in the *System Monitor Guide and Reference*

Related tasks:

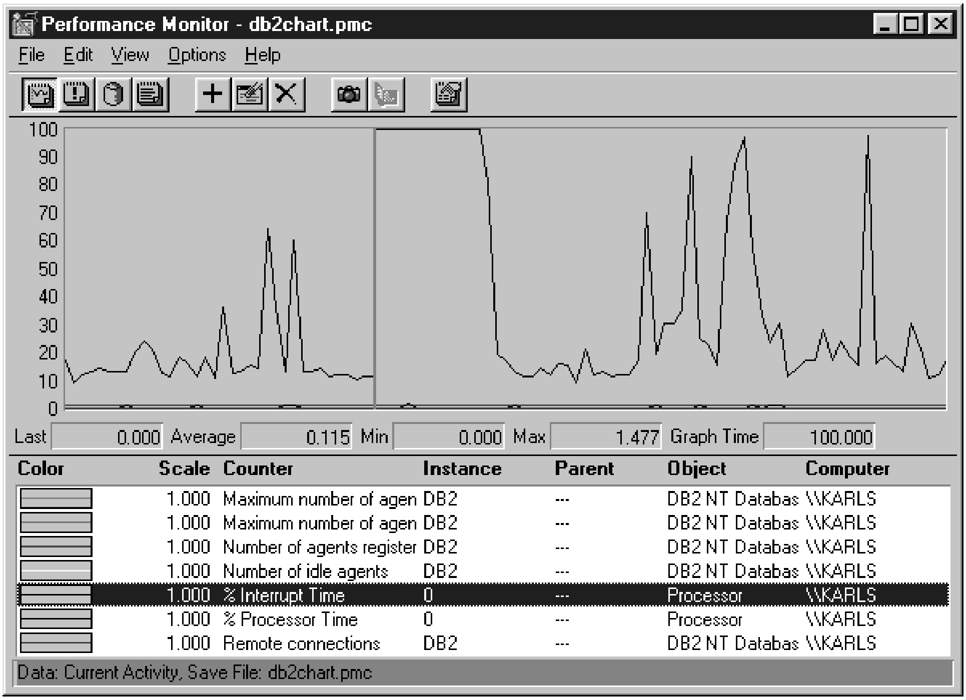
- "Setting monitor switches from a client application" in the *System Monitor Guide and Reference*
- "Setting monitor switches from the CLP" in the *System Monitor Guide and Reference*

Monitoring performance using the Windows Performance Monitor

Windows® NT and Windows 2000 provides 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 figure below, several DB2 measures are being graphed against CPU usage. The collection of values being charted was saved in the file `db2chart.pmc`. You may save as many PMC files as you like, each reflecting a different cross-section of system performance.



Last 0.000 Average 0.115 Min 0.000 Max 1.477 Graph Time 100.000

Color	Scale	Counter	Instance	Parent	Object	Computer
▭	1.000	Maximum number of agen	DB2	---	DB2 NT Databas	\\KARLS
▭	1.000	Maximum number of agen	DB2	---	DB2 NT Databas	\\KARLS
▭	1.000	Number of agents register	DB2	---	DB2 NT Databas	\\KARLS
▭	1.000	Number of idle agents	DB2	---	DB2 NT Databas	\\KARLS
▭	1.000	% Interrupt Time	0	---	Processor	\\KARLS
▭	1.000	% Processor Time	0	---	Processor	\\KARLS
▭	1.000	Remote connections	DB2	---	DB2 NT Databas	\\KARLS

Data: Current Activity, Save File: db2chart.pmc

Figure 6. Performance Monitor

To enable monitoring of local applications you will need to turn the DB2CONNECT_IN_APP_PROCESS environment variable off.

Related concepts:

- "Monitoring connections for remote clients" on page 95

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 wish 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 on 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:

```
           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)
```

Related concepts:

- "Monitoring connections for remote clients" on page 95

Related reference:

- "GET SNAPSHOT" in the *Command Reference*

DCS application status

As of DB2® Connect Version 5.2, 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.

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 an APPC connection (DB2 Connect Enterprise Edition to DB2 Universal Database for OS/390 and z/OS):

```
Auth Id  Application Name     Appl.      Host Application ID
                              Handle
-------- -------------------- ---------- --------------------------------
USERID   db2bp_41             0          CAIBMOML.OMXT4H0A.A79EAA3C6E29
```

It returns the following information for a TCP/IP connection (DB2 Connect Enterprise Edition to DB2 Universal Database for OS/390 and z/OS):

```
Auth Id  Application Name     Appl.      Host Application ID
                              Handle
-------- -------------------- ---------- --------------------------------
USERID   db2bp_41             2          0915155C.9704.1517172201BE
```

Auth.Id
The authorization ID that was used to log on to the host or iSeries™ 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:

- The DRDA® correlation token (CRRTKN), for unprotected conversations.
- The logical unit of work identifier (LUWID), for two-phase conversations protected by an SNA Syncpoint Manager (SPM).
- 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 host or iSeries 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:

Table 8. DB2 LIST DCS APPLICATIONS SHOW DETAIL

```
Auth Id  Application Name     Appl.      Client Application Id            Seq# Client
                              Handle                                           DB Alias
-------- -------------------- ---------- -------------------------------- ---- --------
NEWTON   db2bp                0          09151251.07D3.980925183850       0001 MVSDB

Client   Client   Client     Host Application Id              Seq# Host DB Name
Node     Release  Codepage
-------- -------- ---------- -------------------------------- ---- --------------------
antman   SQL05020 819        G9151251.G7D4.980925183851       0000 GILROY

Host
Release
--------
DSN05011
```

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 host or iSeries 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 example, for a client connected via NetBIOS, this is the value of the NNAME database manager configuration parameter. 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:

- SQL01010 for Version 1 of DB2 UDB for OS/2
- SQL01011 for Version 1 of UNIX-based DB2 products and Client Application Enablers.
- SQL02010 for Version 2 of DB2 products and Client Application Enablers.
- SQL02020 for Version 2.1.2 of DB2 products and Client Application Enablers.
- SQL05000 for Version 5.0 of DB2 Universal Database and DB2 Connect products and their clients.
- SQL05020 for Version 5.2 of DB2 Universal Database and DB2 Connect products and their clients.
- SQL06010 for Version 6.1 of DB2 Universal Database and DB2 Connect products and their clients.
- 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.

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 host or iSeries database server databases, the host or iSeries 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 host or iSeries database server product (for example, DSN for DB2 Universal Database for OS/390 and z/OS, ARI for DB2 for VSE & VM, or QSQ for DB2 UDB for iSeries)

VV Represents a two-digit version number, such as `01`.

RR Represents a two-digit release number.

M Represents a one-digit modification level.

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                     = 09151251.0AA7.981015204853
  Sequence number                         = 0001
  Authorization ID                        = NEWTON
  Application name                        = db2bp
  Application handle                      = 1
  Application status                      = waiting for request
  Status change time                      = 10-15-1998 16:50:29.489160
  Client node                             = antman
  Client release level                    = SQL05020
  Client platform                         = AIX
  Client protocol                         = TCP/IP
  Client codepage                         = 819
  Process ID of client application        = 39324
  Client login ID                         = smith
  Host application ID                     = G9151251.GAA8.981015204854
  Sequence number                         = 0000
  Database alias at the gateway           = MVSDB
  DCS database name                       = DCSDB
  Host database name                      = GILROY
  Host release level                      = DSN05011
  Host CCSID                              = 500
```

The application status field contains one of the following three values:

1. `connect pending - outbound`. This means that the request to connect to a host or iSeries 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 host or iSeries 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 host or iSeries 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.

Using the DB2 Control Center to List Extended DCS Applications Information:

You can use the DB2 Control Center to perform DB2 Connect gateway monitoring. This section shows how you can use the Control Center to display the same report as provided by the **list dcs applications extended** command.

To view the extended report for any application:

1. Expand the tree under the **systems** icon of the Control Center to display **System —> Instances —> Gateway Connections**. If you right-click the mouse on any instance under the **Gateway Connections** folder, a pop-up menu will appear. Select the Applications... item from this menu. The Applications window appears. This window has a tabbed notebook appearance, with one tab titled **Applications**. If there are gateway applications in your instance, there will be a second tab titled **Gateway Applications**.
2. The main window of each page contains columns of information corresponding to the fields of the `LIST DCS APPLICATIONS EXTENDED` report. The first six columns, visible in the window, provide the following data:

   ```
   Client Node
   Application Name
   Client Application ID
   Host Application ID
   Database Alias at Gateway
   Status
   ```

   The rest of the fields of the report can be viewed by moving the horizontal scroll bar at the bottom of the window.

All of the fields listed by the `LIST DCS APPLICATIONS EXTENDED` command are present in this view.

Chapter 11. Troubleshooting

Problem determination

The DB2 Connect environment involves multiple software, hardware and communications products. Problem determination 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.

Related concepts:

- “Diagnostic tools” on page 108
- “Gathering relevant information” on page 107
- “Initial connection is not successful” on page 108
- “Problems encountered after an initial connection” on page 109
- “Trace utility” on page 111
- “Generating a CS AIX CPIC APPC API trace” on page 121

Problem determination concepts

Gathering relevant information

Problem determination 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 client and DB2 Connect workstation, DB2 Connect workstation and host or iSeries™ database server, all clients or one client?

- What can you determine by the content of the message and the tokens returned in the message?
- Will using diagnostic tools 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?

Related concepts:

- “Problem determination” on page 107

Diagnostic tools

When you encounter a problem, you can use the following:

- The first failure service log, where diagnostic information is consolidated and stored in a readable format, is stored in the administration notification log.
- Both logs are found in the path specified:

  This file is located in `/u/db2/sqllib/db2dump/notifyloglevel.nfy` on UNIX® systems, where `db2` represents the instance name.

  This file is located in `x:\sqllib\db2\db2diag.log` on Windows® systems, where `x:` represents the logical drive, and `db2` represents the instance name.
- For Windows NT® and Windows 2000 systems, you can use the Event Viewer to view the administration notification log.
- The trace utility
- For UNIX-based systems, the **ps** command, which returns process status information about active processes to standard output.
- For UNIX-based 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.

Related concepts:

- “DB2 Connect performance troubleshooting” on page 168
- “Trace utility” on page 111

Initial connection is not successful

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-based 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-based systems, did you edit the nodelock file and enter the password that IBM® supplied?
4. *Were the host or iSeries™ database server and workstation communications configured properly?*
   - There are three configurations that must be considered:
     a. The host or iSeries database server configuration identifies the application requester to the server. The host or iSeries server database management system will have system catalog entries that will define the requestor in terms of location, network protocol and security.
     b. The DB2 Connect workstation configuration defines the client population to the server and the host or iSeries 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 for SNA connections that all the LU (logical unit) and 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 host or iSeries server database administrator and the Network administrators have utilities available to diagnose problems.
5. *Do you have the level of authority required by the host or iSeries server database management system to use the host or iSeries 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 to issue SQL statements against a host or iSeries database server, are you unsuccessful?*
   - Did you follow the procedure to bind the command line processor to the host or iSeries database server?

Related concepts:

- "Problem determination" on page 107

Problems encountered after an initial connection

The following questions are offered as a starting point to assist in narrowing the scope of the 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 PTF 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 PTF level and load that level *after* installing the feature.
2. *Has this error occurred before?*
   - Is there any documented resolution to previous error conditions?
   - Who were the participants and can they provide insight into possible course of action?
3. *Have you explored using communications software commands that return information about the network?*
   - Is there a verification tool available for your SNA software?
   - If you are using TCP/IP there may be 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 may provide a common base.
5. *Was DB2START 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 may have been reached. If another client disconnects from the server, is the client previously not able 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 may be successful, but authorization not 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 may block communication between the client and the server. For example, when using APPC, ensure that a session can be established. When using TCP/IP, ensure that you can PING the remote host.

Related concepts:

- "Problem determination" on page 107

Trace utility

The **db2drdat** utility records the data interchanged between the DB2 Connect server (on behalf of the database client) and the host or iSeries™ database server.

As a database administrator (or application developer), you may find it useful to understand how this flow of data works, because this knowledge can help you determine the origin of a particular problem. For example, if you issue a `CONNECT TO` database statement for a host or iSeries database server, but the command fails and you receive an unsuccessful return code. If you understand exactly what information was conveyed to the host or iSeries database server management system, you may be able to determine the cause of the failure even if the return code information is general. Many failures are caused by simple user errors.

Output from `db2drdat` lists the data streams exchanged between the DB2 Connect workstation and the host or iSeries database server management system. Data sent to the host or iSeries database server is labeled SEND BUFFER and data received from the host or iSeries database server is labeled RECEIVE BUFFER.

If a receive buffer contains SQLCA information, it will be followed by a formatted interpretation of this data and labeled SQLCA. The SQLCODE field of an SQLCA is the *unmapped* value as returned by the host or iSeries database server. The send and receive buffers are arranged from the oldest to the most recent within the file. Each buffer has:

- The process ID
- A SEND BUFFER, RECEIVE BUFFER, or SQLCA label. The first DDM command or object in a buffer is labeled DSS TYPE.

The remaining data in send and receive buffers is divided into five columns, consisting of:

- A byte count.
- Columns 2 and 3 represent the DRDA® data stream exchanged between the two systems, in ASCII or EBCDIC.
- An ASCII representation of columns 2 and 3.
- An EBCDIC representation of columns 2 and 3.

For more information, see the *DB2 for OS/390 Reference for Remote DRDA Requesters and Servers*, the *Distributed Relational Database Reference*, and the *Distributed Data Management Architecture Level 3: Reference*.

Related concepts:

- "Trace output file analysis" on page 113
- "Trace output" on page 112
- "Trace parameters" on page 120
- "Trace syntax" on page 120

Trace utility details

Trace output

The **db2drdat** utility writes the following information to *tracefile*:

- -r
  - Type of DRDA® reply/object
  - Receive buffer
- -s
  - Type of DRDA request
  - Send buffer
- -c
  - SQLCA
- CPI-C error information
  - Receive function return code
  - Severity
  - Protocol used
  - API used
  - Function
  - CPI-C return code
  - Error number

  - Internal return code.
- SNA error information
  - Receive function return code
  - Severity
  - Protocol used
  - Function
  - Partner LU name
  - Error number.
- TCP/IP error information
  - Receive function return code
  - Severity
  - Protocol used
  - API used
  - Function
  - Error number.

Notes:

1. A value of zero for the exit code indicates that the command completed successfully, and a non-zero value indicates that it did not.
2. The fields returned vary based on the API used. The SNA API is used only for 2PC SPM connections.
3. The fields returned vary based on the platform on which DB2 Connect is running, even for the same API.
4. If the **db2drdat** command sends the output to a file that already exists, the old file will be erased unless the permissions on the file do not allow it to be erased.

Related concepts:

- "Trace output file analysis" on page 113
- "Trace parameters" on page 120
- "Trace syntax" on page 120
- "Trace utility" on page 111

Trace output file analysis

The following figures show sample output illustrating some DRDA® data streams exchanged between DB2 Connect workstations and a host or iSeries™ database server. From the user's viewpoint, a `CONNECT TO` database command has been issued using the command line processor.

Figure 7 uses DB2 Connect Enterprise Edition Version 8 and DB2 UDB for OS/390; Version 5.1 over an APPC connection.

```
1       DB2® fnc_data      gateway_drda_ar      sqljcsend (1.35.10.80)
        pid 95212; tid 537115484; node 0; cpid 0; sec 0; nsec 0; tpoint 177

        SEND BUFFER:  EXCSAT RQSDSS        (ASCII)           (EBCDIC)
         0 1 2 3 4 5 6 7  8 9 A B C D E F   0123456789ABCDEF  0123456789ABCDEF
  0000  006AD04100010064 10410020115E8482   .j.A...d.A. .^..  .|}..........;db
  0010  F282974040404040 4040404040404040   ...@@@@@@@@@@@@@  2bp
  0020  4040F0F0F0F1F7F3 C5C3000C116DA685   @@...........m..    000173EC..._we
  0030  81A2859340400013 115AC4C2F240C396   ....@@...Z...@..  asel  ...]DB2 Co
  0040  95958583A340F54B F200141404140300   .....@.K........  nnect 5.2.......
  0050  0414440003240700 05240F0003000D11   ..D..$...$......  ................
  0060  47D8C4C2F261F6F0 F0F00085D0010002   G....a..........  .QDB2/6000.e}...
  0070  007F200100162110 E2C1D56DC6D9C1D5   .. ...!....m....  ."......SAN_FRAN
  0080  C3C9E2C3D6404040 40400006210F2407   .....@@@@@..!.$.  CISCO     ......
  0090  000D002FD8E3C4E2 D8D3C1E2C3000C11   .../............  ....QTDSQLASC...
  00A0  2EE2D8D3F0F5F0F2 F0003C210437E2D8   ..........
```

Figure 7. Example of Trace Output (APPC connection) (Part 1 of 2)

```
3       DB2 fnc_data      gateway_drda_ar      sqljcsend (1.35.10.80)
        pid 95212; tid 537115484; node 0; cpid 0; sec 0; nsec 0; tpoint 177

        SEND BUFFER:  RDBCMM  RQSDSS        (ASCII)           (EBCDIC)
         0 1 2 3 4 5 6 7  8 9 A B C D E F   0123456789ABCDEF  0123456789ABCDEF
  0000  000AD00100010004 200E               ........ .        ..}.......

4       DB2 fnc_data      gateway_drda_ar      sqljcrecv (1.35.10.81)
        pid 95212; tid 537115484; node 0; cpid 0; sec 0; nsec 0; tpoint 178

        RECEIVE BUFFER:  ENDUOWRM RPYDSS    (ASCII)           (EBCDIC)
         0 1 2 3 4 5 6 7  8 9 A B C D E F   0123456789ABCDEF  0123456789ABCDEF
  0000  002BD05200010025 220C000611490004   .+.R...%"....I..  ..}.............
  0010  00162110E2C1D56D C6D9C1D5C3C9E2C3   ..!....m........  ....SAN_FRANCISC
  0020  D640404040400005 211501000BD00300   .@@@@@..!.......  O     .......}..
  0030  0100052408FF                        ...$..            ......

5       DB2 fnc_data      gateway_drda_ar      sqljmsca (1.35.10.108)
        pid 95212; tid 537115484; node 0; cpid 0; sec 0; nsec 0; tpoint 179
        SQLCA

        SQLCAID:  SQLCA
        SQLCABC:  136
        SQLCODE:  0
        SQLERRML: 0
        SQLERRMC:
        SQLERRP:  DSN
        SQLERRD[0->5]: 00000000, 00000000, 00000000, 00000000, 00000000, 00000000
        SQLWARN(0->A):  ,  ,  ,  ,  ,  ,  ,  ,  ,  ,
        SQLSTATE: 00000
```

Figure 7. Example of Trace Output (APPC connection) (Part 2 of 2)

Figure 8 on page 116 uses DB2 Connect Enterprise Edition Version 8 and DB2 Universal Database for OS/390 and z/OS Version 5.1 over a TCP/IP connection.

```
1       DB2 fnc_data      gateway_drda_ar      sqljcsend (1.35.10.80)
        pid 80286; tid 537125164; node 0; cpid 0; sec 0; nsec 0; tpoint 177

        SEND BUFFER:  EXCSAT RQSDSS         (ASCII)           (EBCDIC)
         0 1 2 3 4 5 6 7  8 9 A B C D E F   0123456789ABCDEF  0123456789ABCDEF
  0000  006ED04100010068 10410020115E8482   .n.A...h.A. .^..  .>}..........;db
  0010  F282974040404040 4040404040404040   ...@@@@@@@@@@@@@  2bp
  0020  4040F0F0F0F1F3F9 F9C5000C116DA685   @@...........m..    0001399E..._we
  0030  81A2859340400013 115AC4C2F240C396   ....@@...Z...@..  asel  ...]DB2 Co
  0040  95958583A340F54B F200181404140300   .....@.K........  nnect 5.2.......
  0050  0514740005240700 05240F0003144000   ..t..$...$....@.  .............. .
  0060  05000D1147D8C4C2 F261F6F0F0F00010   ....G....a......  .....QDB2/6000..
  0070  D0410002000A106D 000611A20003003C   .A.....m.......<  }......_...s....
  0080  D04100030036106E 000611A200030016   .A...6.n........  }......>...s....
  0090  2110E2C1D56DC6D9 C1D5C3C9E2C3D640   !....m.........@  ..SAN_FRANCISCO
  00A0  40404040000C11A1 9781A2A2A6969984   @@@@............      ....password
  00B0  000A11A0A4A28599 8984009CD0010004   ................  ....userid..}...
  00C0  0096200100162110 E2C1D56DC6D9C1D5   .. ...!....m....  .o......SAN_FRAN
  00D0  C3C9E2C3D6404040 40400006210F2407   .....@@@@@..!.$.  CISCO     ......
  00E0  000D002FD8E3C4E2 D8D3C1E2C3000C11   .../............  ....QTDSQLASC...
  00F0  2EE2D8D3F0F5F0F2 F0003C210437E2D8   ..........
3       DB2 fnc_data      gateway_drda_ar      sqljcsend (1.35.10.80)
        pid 80286; tid 537125164; node 0; cpid 0; sec 0; nsec 0; tpoint 177

        SEND BUFFER:  RDBCMM  RQSDSS        (ASCII)           (EBCDIC)
         0 1 2 3 4 5 6 7  8 9 A B C D E F   0123456789ABCDEF  0123456789ABCDEF
  0000  000AD00100010004 200E               ........ .        ..}.......

4       DB2 fnc_data      gateway_drda_ar      sqljcrecv (1.35.10.81)
        pid 80286; tid 537125164; node 0; cpid 0; sec 0; nsec 0; tpoint 178

        RECEIVE BUFFER:  ENDUOWRM RPYDSS    (ASCII)           (EBCDIC)
         0 1 2 3 4 5 6 7  8 9 A B C D E F   0123456789ABCDEF  0123456789ABCDEF
  0000  002BD05200010025 220C000611490004   .+.R...%"....I..  ..}.............
  0010  00162110E2C1D56D C6D9C1D5C3C9E2C3   ..!....m........  ....SAN_FRANCISC
  0020  D640404040400005 211501000BD00300   .@@@@@..!.......  O     .......}..
  0030  0100052408FF                        ...$..            ......

5       DB2 fnc_data      gateway_drda_ar      sqljmsca (1.35.10.108)
        pid 80286; tid 537125164; node 0; cpid 0; sec 0; nsec 0; tpoint 179
        SQLCA

        SQLCAID:  SQLCA
        SQLCABC:  136
        SQLCODE:  0
        SQLERRML: 0
        SQLERRMC:
        SQLERRP:  DSN
        SQLERRD[0–>5]: 00000000, 00000000, 00000000, 00000000, 00000000, 00000000
        SQLWARN(0–>A):  ,  ,  ,  ,  ,  ,  ,  ,  ,  ,
        SQLSTATE: 00000
```

Figure 8. Example of Trace Output (TCP/IP connection)

The following information is captured in the traces:

- The process ID (PID) of the client application
- The RDB_NAME cataloged in the database connection services (DCS) directory
- The DB2 Connect CCSID(s)
- The host or iSeries database server CCSID(s)
- The host or iSeries database server management system with which the DB2 Connect system is communicating.

The first buffer contains the Exchange Server Attributes (EXCSAT) and Access RDB (ACCRDB) commands sent to the host or iSeries database server management system. It sends them as a result of a `CONNECT TO` database command.

The next buffer contains the reply that DB2 Connect received from the host or iSeries database server management system. It contains an Exchange Server Attributes Reply Data (EXCSATRD) and an Access RDB Reply Message (ACCRDBRM).

EXCSAT and ACCRDB:

The EXCSAT command contains the workstation name of the client specified by the Server Name (SRVNAM) object, which is code point X'116D', according to DDM specification. The EXCSAT command is found in the first buffer. Within the EXCSAT command, the values X'116DA68581A28593' (coded in CCSID 500) are translated to *weasel* once the X'116D' is removed.

The EXCSAT command also contains the EXTNAM (External Name) object, which is often placed in diagnostic information on the host or iSeries database management system. It consists of a 20-byte application ID followed by an 8-byte process ID (or 4-byte process ID and 4-byte thread ID). It is represented by code point X'115E', and in this example its value is `db2bp_32` padded with blanks followed by `0000BE5C`. On a UNIX-based database client, this value can be correlated with the **ps** command, which returns process status information about active processes to standard output.

The ACCRDB command contains the RDB_NAME in the RDBNAM object, which is code point X'2110'. The ACCRDB command follows the EXCSAT command in the first buffer. Within the ACCRDB command, the values X'2110E2C1D56DC6D9C1D5C3C9E2C3D6' are translated to SAN_FRANCISCO once the X'2110' is removed. This corresponds to the target database name field in the DCS directory.

The accounting string has code point X'2104'.

The code set configured for the DB2 Connect workstation is shown by locating the CCSID object CCSIDSBC (CCSID for single-byte characters) with code point X'119C' in the ACCRDB command. In this example, the CCSIDSBC is X'0352', which is 850.

If the additional objects CCSIDDBC (CCSID for double-byte characters) and CCSIDMBC (CCSID for mixed-byte characters), with code points X'119D' and X'119E' respectively, are present, the DB2 Connect workstation is configured for DBCS code page support. Since the example output file does not include the two additional code points, the workstation is not configured for DBCS.

Note: The TCP/IP flows contains two new commands: ACCSEC used to access the security manager and exchange supported security mechanisms, and SECCHK, which contains the authentication tokens used to authenticate the end user of the connection. ACCSEC and SECCHK only appear for TCP/IP connections, and they do so between EXCSAT and ACCRDB.

EXCSATRD and ACCRDBRM:

CCSID values are also returned from the host or iSeries database server in the Access RDB Reply Message (ACCRDBRM) within the second buffer. This buffer contains the EXCSATRD followed by the ACCRDBRM. The example output file contains CCSID values for the host or iSeries database server system of 500 (X'01F4', SBCS CCSID).

If DB2 Connect does not recognize the code page coming back from the host or iSeries database server, SQLCODE -332 will be returned to the user with the source and target code pages. If the host or iSeries database server doesn't recognize the code set sent from DB2 Connect, it will return VALNSPRM (Parameter Value Not Supported, with DDM code point X'1252'), which gets translated into SQLCODE -30073 for the user.

The ACCRDBRM also contains the parameter PRDID (Product-specific Identifier, with code point X'112E'). The value is X'C4E2D5F0F5F0F1F0'. This hex string corresponds to DSN05010 in EBCDIC. According to standards, DSN is DB2 Universal Database for OS/390 and z/OS. The version, 5.1, is also indicated. ARI is DB2 for VSE & VM, SQL is DB2 Common Server, and QSQ is DB2 UDB for iSeries.

Subsequent buffers:

You can analyze subsequent send and receive buffers for additional information. The third buffer contains a commit. The **commit** command instructs the host or iSeries database server management system to commit the current unit of work. The fourth buffer is received from the host or iSeries database server database management system as a result of a commit or rollback. It contains the End Unit of Work Reply Message (ENDUOWRM), which indicates that the current unit of work has ended.

In this example, it contains a null SQLCA, indicated by DDM code point X'2408' followed by X'FF'. A null SQLCA (X'2408FF') indicates success (SQLCODE 0). When a receive buffer contains an SQLCA (possibly a null SQLCA), ddcstrc will follow this receive buffer with a formatted interpretation of the SQLCA information.

Figure 9 shows an example of a receive buffer containing an error SQLCA, and the formatted display of the SQLCA. This SQLCA is the result of an attempt to delete rows from a nonexistent table.

```
1       DB2 fnc_data      gateway_drda_ar      sqljcrecv (1.35.10.81)
        pid 48732; tid 1; node 0; cpid 0; sec 0; nsec 0; tpoint 178

        RECEIVE BUFFER:  SQLCARD OBJDSS     (ASCII)           (EBCDIC)
         0 1 2 3 4 5 6 7  8 9 A B C D E F   0123456789ABCDEF  0123456789ABCDEF
  0000  0065D0030001005F 240800FFFFFF34F4   .e....._$.....4.  ..}....^.......4
  0010  F2F7F0F4C4E2D5E7 D6E3D34000E2C1D5   ...........@....  2704DSNXOTL .SAN
  0020  6DC6D9C1D5C3C9E2 C3D64040404040FF   m.........@@@@@.  _FRANCISCO     .
  0030  FFFE0C0000000000 000000FFFFFFFF00   ................  ................
  0040  0000000000000040 4040404040404040   .......@@@@@@@@@  .......
  0050  40400000000FC4C4 C3E2E4E2F14BD4E8   @@...........K..    ....DDCSUS1.MY
  0060  E3C1C2D3C5                          .....             TABLE

2       DB2 fnc_data      gateway_drda_ar      sqljmsca (1.35.10.108)
        pid 48732; tid 1; node 0; cpid 0; sec 0; nsec 0; tpoint 179
        SQLCA

        SQLCAID:  SQLCA
        SQLCABC:  136
        SQLCODE:  -204
        SQLERRML: 15
        SQLERRMC: DDCSUS1.MYTABLE
        SQLERRP:  DSNXOTL
        SQLERRD[0—>5]: FFFFFE0C, 00000000, 00000000, FFFFFFFF, 00000000, 00000000
        SQLWARN(0—>A):  ,  ,  ,  ,  ,  ,  ,  ,  ,  ,
        SQLSTATE: 42704
```

Figure 9. Receive Buffer Example

Related concepts:

- "Trace output" on page 112
- "Trace parameters" on page 120
- "Trace syntax" on page 120
- "Trace utility" on page 111

Trace parameters

The trace utility has the following parameters:

on Turns on DB2® Connect tracing of DRDA® flows with the host or iSeries™ database server.

off Turns off DB2 Connect™ tracing of DRDA flows with the host or iSeries database server.

-i Timestamps will be included in the trace information.

-r Traces DRDA data streams received from the host or iSeries server system.

-s Traces DRDA data streams sent to the host or iSeries database server.

-c Traces the SQLCA received from the host or iSeries database server. The default is -r, -s, and -c.

-l=*length*
Specifies the size of the buffer used to store the trace information. The default is 1M, and the minimum is 64K.

-t=*tracefile*
Specifies the destination for the trace; *tracefile* may be the name of a file or a standard device. If a file name is specified without a complete path, the current path is used for the missing parts. The default file name is `ddcstrc.dmp`.

-p=*pid* Traces events only for this process. If -p is not specified, all processes for the user's instance are written to the output file.

Note: For a remote client, the *pid* can be found in the Agent ID field returned by the database system monitor.

Related concepts:

- "Trace output file analysis" on page 113
- "Trace output" on page 112
- "Trace syntax" on page 120
- "Trace utility" on page 111

Trace syntax

You can invoke the **db2drdat** command from a command prompt with the following syntax:

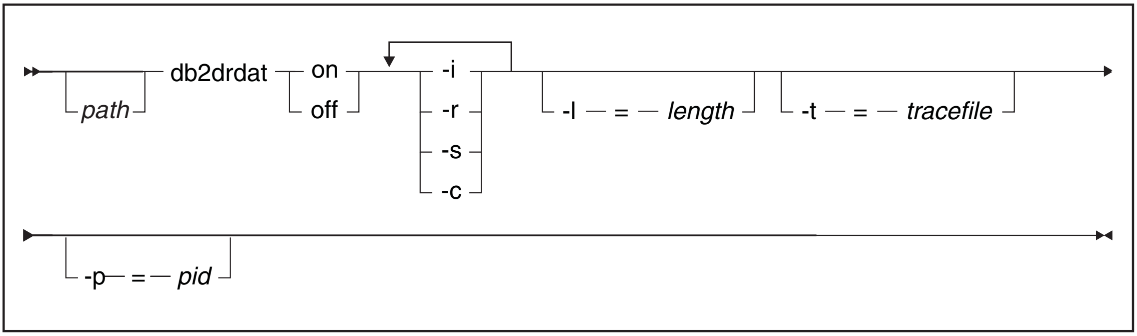


Figure 10. Syntax of the db2drdat command

The syntax of this command may vary slightly depending on the operating system that you are using.

Related concepts:

- “Trace output file analysis” on page 113
- “Trace output” on page 112
- “Trace parameters” on page 120
- “Trace utility” on page 111

Generating a CS AIX CPIC APPC API trace

There are two methods of generating an CS/AIX CPIC/APPC API trace.

Method 1:

Add:

```
SNATRC=/tmp/file1:/tmp/file2:
```

to `$HOME/sqllib/cfg/vendor.cfg` file.

Method 2:

```
export SNATRC=/tmp/file1:/tmp/file2:
db2set db2envlist=SNATRC
```

For both methods, you also need to run the following commands:

```
db2stop
db2start
```

in order for the DB2® gateway to pick up the new environment variable.

Related concepts:

- “Problem determination” on page 107

Common DB2 Connect problems

This topic lists the most common symptoms of connection problems encountered 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 may be provided.

Note: For message and return code combinations specific to APPC communications, an SNA sense code may also be indicated. At present, any SNA sense code information associated with a particular message must be obtained from the SNA subsystem.

SNA sense codes can be viewed by looking through system logs. Whether this is the case or not depends on the SNA subsystem being used, and in some situations you may have to recreate the problem with an SNA trace active to obtain the sense code information.

SQL0965 or SQL0969:

Symptom

Messages SQL0965 and SQL0969 can be issued with a number of different return codes from DB2 Universal Database (UDB) for iSeries, DB2 UDB for OS/390 and z/OS, and DB2 for VM & 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 host or iSeries database cannot be translated. Correct the problem, based on the error code, then resubmit the failing command.

SQL1338 During CONNECT:

Symptom / Cause

The symbolic destination name was not defined, or it is not properly defined.

For example, this can happen when an APPC node is used and the symbolic destination name specified in the DB2 node directory does not match a CPI-C entry in the local APPC communications subsystem configuration.

Another cause can be that there is more than one SNA stack installed on your machine. You may need to check PATH and LIBPATH to ensure that the stack you want to use is referenced first.

Solutions

1. Ensure the CPIC Side Information profile name specified in the DB2 Node directory entry matches the SNA configuration (it is case sensitive).
2. You may need to check PATH and LIBPATH to ensure that the SNA stack that you want to use is referenced first.

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 may have been a successful client connection previously.

If `diaglevel = 4`, then `db2diag.log` may contain a similar entry, for example:

```
2001-05-30-14.09.55.321092   Instance:svtdbm5   Node:000
PID:10296(db2tcpcm)   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, APPC and others, and usually the message indicates that one of the communications protocols defined to DB2 Connect is not configured properly.

Often the cause may 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, svcename, nname, or tpname).

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 may still appear to exist on the client, and it may become possible to restore the connection without further intervention by issuing the commands shown below.

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.

Check the db2dump directory for an ffdc dump (pid.000). Then, format this dump file with `db2fdump` and look in the result file for ″ERROR″.

SQL30060:

Symptom

SQL30060N ″<authorization-ID>″ does not have the privilege to perform operation ″<operation>″.

Solution

When connecting to DB2 for OS/390® and z/OS, the Communications Database (CDB) tables have not been updated properly.

SQL30061:

Symptom

Connecting to the wrong host or iSeries™ database server location - no target database can be found.

Solution

The wrong server database name may 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 Universal Database for OS/390 and 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 an APPC or APPN® node are:

```
db2 catalog appc node <node_name> remote <sym_dest_name>
             security program
db2 catalog dcs database <local_name> as <real_db_name>
db2 catalog database <local_name> as <alias> at node <node_name>
             authentication server
```

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>
```

SQL30073 with Return Code 119C During CONNECT:

Symptom

Message SQL30073 is issued with return code 119C. This happens when the target server database does not support the code page used by the DB2 client (going through DB2 Connect). The code page is derived from the configuration of the operating environment in which the DB2 client is running.

Solution

This problem can often be resolved by installing a fix at the target server database system. Contact the appropriate service organization and obtain and apply any fix which might be recommend for this symptom.

As a temporary workaround, the user can override the default code page by setting the DB2CODEPAGE environment variable. Check the locale or set DB2CODEPAGE=850.

On UNIX® platforms, the user may be able to switch to a different code page by setting the LANG environment variable to a different value.

SQL30081N with Return Code 1:

Symptom

Symptom is the following message plus an SNA sense code:

```
db2 connect to <database name> user <userid>
Enter password for <userid>:
SQL30081N  A communication error has been detected.
Communication protocol
being used: "APPC".  Communication API being used: "CPI-C".
Location where
the error was detected: "".  Communication function detecting
the error:
"cmallc".  Protocol specific error code(s): "1", "*",
"0x10030021".
SQLSTATE=08001
```

Solution(s)

In this sample the sense code is 10030021.

The most common sense codes associated with this error message, and the suggested solution in each case, are as follows:

SQL30081N with Return Code 1 and sna sense code 0877002C
Wrong network name has been specified.

SQL30081N with Return Code 1 and SNA sense code ffff0003
The wrong MAC address has been specified or the SNA link is not active.

SQL30081N with Return Code 1 and SNA sense code 10030021
There is an LU type mismatch.

SQL30081N with Return Code 1 and SNA sense code 084B6031
The MAXDBAT in DSNZPARM (at a DB2 for OS/390 and z/OS™ host) is set to 0.

Other suggestions:

- When creating the Local LU profile, define the LU as the default LU. For example, in the SNA Feature list panel in CM/2, either:
  1. Place a checkmark in the checkbox ’Use this local LU as your default local LU alias’, or
  2. Set the profile or environment variable APPCLLU on the DB2 Connect Enterprise Edition server system to the Local LU name. You can do this on Windows® systems using the Control Panel.
- Check that SNA is started on the DB2 Connect server.
- If you are using DB2 for OS/390 and z/OS, check that the Distributed Data Facility (DDF) address space is started and that DB2 is running.

SQL30081N with Return Code 2:

Symptom

Message SQL30081N is received with Return Code 2 and SNA Sense Code 08120022.

Solution

The NUMILU parameter at the NCP (host or iSeries end of the link) may be set to the default (0). Check this parameter. Modify the NCP definition if necessary before retrying, after putting the change into effect.

SQL30081N with Return Code 9:

Symptom

Symptom is the following message (the SNA sense code is not required in this case):

```
db2 connect to <database> user <userid>
SQL30081N  A communication error has been detected.
Communication protocol
being used: "APPC".  Communication API being used: "CPI-C".
Location where
the error was detected: "".  Communication function detecting
the error:
"cmsend".  Protocol specific error code(s): "9", "*",
"0x10086021".
SQLSTATE=08001
```

Solution

The problem is that the Transaction Program name (TPNAME) is not defined correctly on the DB2 Connect system. For example, you may have updated your SNA configuration, but not yet verified it at the DB2 Connect server.

SQL30081N with Return Code 10:

Symptom

The symptom is the following message (the SNA sense code is not required):

```
SQL30081N  A communication error has been detected.
Communication protocol
being used: "APPC".  Communication API being used: "CPI-C".
Location where
the error was detected: "".  Communication function detecting
the error:
"cmrcv".  Protocol specific error code(s): "10", "*", "*".
SQLSTATE=08001
```

Solution

Ensure that DB2 is correctly installed.

SQL30081N with Return Code 20:

Symptom

```
SQL30081N  A communication error has been detected.
Communication protocol
being used: "APPC".  Communication API being used: "CPI-C".
Location where
the error was detected: "".  Communication function detecting
the error:
"xcstp".  Protocol specific error code(s): "20", "*", "*".
SQLSTATE=08001
```

Solution

Ensure that the SNA subsystem is started on the DB2 Connect system.

SQL30081N with Return Code 27:

Symptom

Message SQL30081N is received with Return Code 27 and SNA Sense Code 800Axxxx.

Solution

The VTAM® Path Information Unit (PIU) is too large.

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 host or iSeries database server.

1. The `DB2COMM` profile variable may set incorrectly on the DB2 Connect server. Check this. For example, the command `db2set db2comm=tcpip` should appear in `sqllib/db2profile` when running DB2 Extended Enterprise Edition on AIX.
2. There may be a mismatch between the TCP/IP service name and/or port number specifications at the DB2 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 file to check that DB2 TCP/IP communications have been started. You should see output similar to the following:

```
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 may 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 may be required.

SQL30082 RC=24 During CONNECT:

Symptom

SQL1403N The username and/or 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 DB2 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.

Related concepts:

- “Common DB2 DRDA AS problems” on page 130
- “Problem determination” on page 107
- “Trace utility” on page 111

Common DB2 DRDA AS problems

This topic lists the most common problem scenarios found when using a DB2® DRDA® application server.

Communication errors during CONNECT:

Ensure the following are set properly at the DB2 UDB end.

APPC/SNA LU 6.2

1. SNA configuration

   Make sure the TP name is configured if necessary.

   Also, if security SAME is to be used from the DRDA AR, ensure that it is enabled for the DRDA AR LU.
2. Database manager configuration TPNAME parameter
3. Environment variable DB2COMM set to include APPC

Ensure that **db2start** completes without any warning.

TCP/IP

1. Services file
2. Database manager configuration SVCENAME parameter
3. Environment variable DB2COMM set to include `TCPIP`. Make sure that **db2start** completes without any warning.

DRDA Error during CONNECT:

APPC/SNA LU 6.2

If SNA Server for AIX® is in use, make sure that the group name for the ~/sqllib/adm/db2sysc executable is in the ″Trusted group names″ field in the ″SNA System Defaults″ profile in the SNA configuration.

TCP/IP

If the DRDA AR is DB2 for OS/390® and z/OS, make sure that the following fixes have been applied: APAR PQ05771/PTF UQ06843.

Database not found error during CONNECT:

Ensure that the DRDA AR is configured with the database alias for the target DB2 UDB database.

Security error during CONNECT over APPC/SNA LU 6.2:

There are special considerations with regard to the AUTHENTICATION setting in the DB2 UDB database manager configuration if the connection from a DRDA AR is over APPC/SNA LU 6.2. If you encounter a security error, please make sure that the database manager configuration AUTHENTICATION setting is set correctly as follows:

Client
With this setting, both security SAME and PROGRAM connections will work.

Server With this setting, only security PROGRAM connections going to the DB2 UDB DRDA AS on AIX with SNA Server will work.

DCS AUTHENTICATION SERVER can be used with DB2 UDB DRDA AS to permit APPC connections from DRDA clients using security SAME (no password required), while at the same time enforcing SERVER authentication (which requires a password) for all other client requests. This feature is enabled by setting the USE_SNA_AUTH configuration parameter to `YES` in the dbm cfg.

With this setting, the following will work:

1. DB2 UDB DRDA AS on AIX with SNA Server:
   ```
   Security SAME
   ```
2. DB2 UDB DRDA AS on Windows, and Solaris Operating Environments:
   ```
   Security SAME or PROGRAM
   ```

These differences exist because some communications subsystems do not expose an incoming password to DB2 UDB.

Errors during BIND:

An SQLCA with SQLCODE -4930 may be returned if a bind option specified by the DRDA AS is not supported. The SQLERRMC field contains information on the bind option causing the error.

Related concepts:

- “Common DB2 Connect problems” on page 122
- “Problem determination” on page 107
- “Trace utility” on page 111

Chapter 12. DB2 Connect Custom Advisor

Before you install DB2 Connect Custom Advisor, the Network Dispatcher machine and the cluster of DB2 Connect servers must be properly installed and configured. For more information on setting up the Network Dispatcher machine and server cluster, see the *IBM Network Dispatcher User's Guide*.

For each DB2 Connect server, DB2 Connect Enterprise Edition Version 6.1 or higher must be installed and all host and iSeries database connections configured. For the Network Dispatcher machine, IBM Network Dispatcher V2.1.1 or above (included as a component of IBM WebSphere Edge Server) and DB2 Connect Enterprise Edition Version 6.1 or above must be installed.

DB2 Connect Custom Advisor concepts

The DB2® Connect Custom Advisor is a Java™ based extension to IBM® Network Dispatcher. During each advisor cycle, DB2 Connect™ Custom Advisor attaches to one of the DB2 Connect servers and takes a snapshot of its System Monitor. From the snapshot, DB2 Connect Custom Advisor calculates an accurate load value that reflects closely to the actual workload of the server. Once the load values from all servers are retrieved, the Dispatcher refreshes the server weights with the new load information and distributes the work accordingly.

Calculating load values:

During each advisor cycle, DB2 Connect Custom Advisor returns to the Dispatcher a load value for each of the servers. This load value must be between 10 and 1000 with 10 indicating a fast server and 1000 being a heavily loaded server. A returned load value of –1 indicates an unavailable server. If a server is flagged as unavailable, no more new connections will be sent to it.

DB2 Connect Custom Advisor calculates the server load value with information provided by the System Monitor Snapshot and information retrieved from the DB2 Database Manager Configuration. The following information is extracted from the System Monitor Snapshot:

- Current® number of connections to DB2 Connect
- Number of connections waiting for clients to send request
- Number of inactive DRDA® agents
- Number of idle agents
- Number of agents registered

- Number of communication errors

The following information is extracted from the Database Manager Configuration:

- Maximum number of agents for server
- Maximum number of coordinating agents for server
- CPU speed

Calculation of load value is controlled by the following determining factors:

1. **Connections:** The number of connections for a server is the primary determinant in calculating the workload for a server. The more connection a server has, the more likely that the server is under a greater workload. During each advisor interval, a percentage value is derived by dividing the current number of connections by the maximum number of connections (maximum number of coordinating agents).

   This percentage is mapped to a value between 10 and 1000. A busy connection is given a double weighting compared to an idle one. A busy connection is defined as a connection that is not waiting for client input. For example, if a server has 10 connections used out of a maximum of 100 with 4 being busy, the returned load from the Connections factor is calculated as follows:

   ```
   Percentage Used
   = [Idle Connections + (Busy Connections x 2)]/Max Connections
   = [6 + (4 x 2)] / 100
   = 0.14

   Connections Factor Load = Load Range x Percentage Used + Offset
   = (1000 x 10) x 0.14 + 10
   = 149
   ```

   If you want a server to receive more connection requests, you may increase its maximum number of coordinating agents parameter in the Database Manager Configuration[1].

2. **Communication errors:** DB2 System Monitor reports the number of communication errors that occur between each DCS database and the DB2 Connect server. By keeping track of the number of communication errors that occur within each advisor interval, current connectivity status of each server can be determined. For each communication error that occurs within an advisor interval, the Communication Errors factor adds a value equivalent to 5% of the load range to the overall load value. The load

[1] In non-partitioned database environments, maximum number of coordination agents (max_coordagents) always equals to maximum number of agents (maxagents), unless the system is configured for intra-partition parallelism.

range is 1000 – 10 = 990 (as mentioned in the previous section, load of 10 indicates a fast server, 1000 indicates a heavily loaded server).

3. **Inactive DRDA agents:** Creating a new DRDA agent is an expensive process. If two servers have roughly the equivalent workload and one of them has inactive DRDA agents in its pool, it is better to send more new requests to this server than having to create new DRDA agents on the other server. For each inactive DRDA agent in a server's pool, the Inactive DRDA Agents factor lowers the overall load by a value equivalent to 5% of the load range. It is possible to increase the agent pool size (num_poolagents) to allow for more inactive DRDA agents to remain in a server's agent pool.
4. **CPU speed:** Database Manager calculates the CPU speed (in milliseconds per instruction) of a server machine during installation of DB2 Connect Enterprise Edition. DB2 Connect Custom Advisor determines the average CPU speed of all servers during initialization. For every time a server is faster than the average, a value equivalent to 5% of the load range is subtracted from the overall load. Inversely, for every time a server is slower than the average, the overall load is raised by a value equivalent to 5% of the load range.

Table 9. Cluster characteristics

Server	CPU Speed (ms/instruction)	Max # of Connections
SERVER1	1.00×10^{-6}	1000
SERVER2	4.00×10^{-6}	1000
SERVER3	1.00×10^{-6}	500

For example, a cluster has the following characteristics:

Average CPU Speed
= $(1.00 \times 10^{-6} + 4.00 \times 10^{-6} + 1.00 \times 10^{-6})/3$
= 2.00×10^{-6} ms/instruction

Since both SERVER1 and SERVER3 have a CPU speed that is one time faster than the average, therefore the overall loads of both servers will be lowered by 5% of the load range.

SERVER1 Load = SERVER1 Load – (990 x 5%)
= SERVER1 Load – 49.5 (same for SERVER3)

SERVER2 has a CPU speed that is one time slower than the average, thus its overall load is raised by 5% of the load range.

SERVER2 Load = SERVER2 Load + (990 x 5%)
= SERVER2 Load + 49.5

The results from all four factors are combined to form a server's load. This resultant load value is returned to the Dispatcher for calculating the server's weight.

Failover protection:

DB2 Connect Custom Advisor enhances IBM Network Dispatcher's failover protection by adding sensors to detect critical conditions within the DB2 Connect engine. During each advisor interval, DB2 Connect Custom Advisor collects the number of agents registered and the number of idle agents from System Monitor Snapshot.

If the number of agents registered minus the number of idle agents is greater than or equal to the maximum number of agents for a server, a load of –1 is returned. All new connections will be routed to the other healthy servers until the number of agents lowers or agents become idle. In order to access System Monitor information of each DB2 Connect server, DB2 Connect Custom Advisor must attach to the server. If a critical communication error is detected during the attach and get snapshot process, a load value of –1 is returned to the Dispatcher.

Calculating server weights:

The Dispatcher sets the server weights based on internal counters, load returned by the advisor and feedback from a system-monitoring program, such as ISS (Interaction Session Support). The administrator can change the proportion of importance given to each input. The proportions must add up to 100. When using DB2 Connect Custom Advisor, the following proportions should work well for most systems:

Table 10. DB2 Connect Custom Advisor proportions

Input	Proportion
Number of active connections on each server (proportion given to active connections)	20
Number of new connections on each server (proportion given to new connections)	20
Input from advisor (proportion given specific to the port)	60
Input from system monitoring tool (proportion given to system metrics)	0

According to the IBM Network Dispatcher User's Guide, it is not recommended to set the first two values below 20. Doing so will disable Dispatcher's load balancing and smoothing. IBM Network Dispatcher balances servers on a per port basis. All incoming requests for a port are distributed between servers based on their weights relative to each other. For example, if a server has a weight of 10 and the other server has a weight of 5, the server set to 10 gets twice as many requests as the server set to 5.

Manual Load Balancing:

DB2 Connect Custom Advisor operates in two modes: normal and manual. In normal mode, DB2 Connect Custom Advisor calculates workload of the servers as described in the previous sections. In manual mode, the system administrator gives each server a relative weight. Load value for each server is directly calculated from this set of weights.

An administrator may want a larger portion of connections directed to a particular server because it has more resources than the rest. This server may have faster processors, more memory, a faster network card or other superior characteristics. An administrator may give each server a weight of 1 to 10. A weight of 1 is given to the server handling the least amount of connections. The weights of the other servers are set relative to this ″least resourceful″ server. If another server is to receive three times the amount of connections as the ″least resourceful″ server, it should be given a weight of 3.

The following example illustrates how manual mode works:

Table 11. Load balancing weight

Server	Assigned Weight
SERVER1	1.5
SERVER2	1.0
SERVER3	1.0
SERVER4	3.0
SERVER5	1.0

SERVER1 Load = 500 / 1.5 = 333
SERVER2 Load = 500 / 1.0 = 500 (same for SERVER3 and SERVER5)
SERVER4 Load = 500 / 3.0 = 167

SERVER2, SERVER3, and SERVER5 (load of 500) will appear to have a workload 3 times heavier than SERVER4 (load of 167) and 1.5 times heavier than SERVER1 (load of 333). If 15 new requests come into the dispatcher, each of SERVER2, SERVER3, and SERVER5 should get approximately 2 connections. SERVER1 should get approximately 3 connections and SERVER4 should receive 6.

Failover protection remains activated in manual mode. If a critical condition appears in one of the servers, all new requests will be directed away from it.

Related tasks:

- "Activating the DB2 Connnect Custom Advisor" on page 141
- "Configuring the DB2 Connect Custom Advisor" on page 139
- "Installing the DB2 Connect Custom Advisor" on page 138
- "Setting up DB2 Connect Custom Advisor" on page 141
- "Troubleshooting DB2 Connect Custom Advisor" on page 142

Related reference:

- "Sample Dispatcher startup script file" on page 142

Installing the DB2 Connect Custom Advisor

Procedure (Windows):

To install the DB2 Connect Custom Advisor on Windows:

1. Unzip the db2cadv.zip into a temporary directory. The unzipped content contains:
   - UsersGuide.pdf – a standalone version of this topic
   - db2cadv.dll – DB2 Connect Custom Advisor native link file
   - db2cadv.ini – DB2 Connect Custom Advisor initialization file
   - ADV_db2cadv.class – DB2 Connect Custom Advisor Java byte code
   - ADV_db2cadv$jNode.class - DB2 Connect Custom Advisor Java byte code
2. Copy the files to their appropriate directories.

Table 12. File destination directories

Files	Directory
ADV_db2cadv.class ADV_db2cadv$jNode.classdb2cadv.ini	%Dispatcher Install Path%\dispatcher\lib\CustomAdvisors\
db2cadv.dll	%Dispatcher Install Path%\dispatcher\lib\

Procedure (AIX):

To install the DB2 Connect Custom Advisor on AIX:

1. Unzip the db2cadv.tar into a temporary directory. The unzipped content contains:
   - UsersGuide.pdf – a standalone version of this topic
   - libdb2cadv.so – DB2 Connect Custom Advisor native shared library file
   - db2cadv.ini – DB2 Connect Custom Advisor initialization file
   - ADV_db2cadv.class – DB2 Connect Custom Advisor Java byte code
   - ADV_db2cadv$jNode.class - DB2 Connect Custom Advisor Java byte code
2. Copy the files to their appropriate directories.

Table 13. File destination directories

Files	Directory
ADV_db2cadv.class ADV_db2cadv$jNode.classdb2cadv.ini	/usr/lpp/nd/dispatcher/lib /CustomAdvisors/
libdb2cadv.so	/usr/lpp/nd/dispatcher/lib/

Related concepts:

- "DB2 Connect Custom Advisor concepts" on page 133

Configuring the DB2 Connect Custom Advisor

The initialization file db2cadv.ini must be configured for DB2 Connect Custom Advisor to start.

Procedure:

DB2 Connect Custom Advisor can be configured to start in one of the two modes: normal or manual.

Configuring for Normal Mode:

Port Tag

The first line (excluding comments and white spaces) of the initialization file is the port tag. The port tag is the port number surrounded by square brackets. In the example below, the port tag is [50000], indicating that the cluster of DB2 Connect servers runs on port number 50000. (DB2 Connect usually runs on port 50000.)

Server Line

Each line following the port tag is called a server line. Each server line represents a server in the cluster. Server line contains information necessary for DB2 Connect Custom Advisor to attach to the server. Each server line has the following format:

```
server = server address DB2 node name logon id password weight
```

The server address is the IP address or server name of the server. DB2 node name is the name given to the server in DB2 Node Directory. Logon id is the logon name of the user account set up in Step 3. Password is the logon password of the user account. Weight should always be set to 1 in normal mode. Setting this value to any number other than 1 puts DB2 Connect Custom Advisor in manual mode. In the example below, two servers are set up. The first server has an address of 192.168.0.1. Its node name is SERVER1 and logon id is DB2NDADV.

Here's an initialization file example for normal mode setup:

```
; DB2CADV Custom Adviser Configuration File
; Created: 2002/03/29

[50000]
server = 192.168.0.1  SERVER1 DB2NDADV xxxxxxxx 1
server = 192.168.0.11 SERVER2 DB2NDADV xxxxxxxx 1
```

Configuring for Manual Mode:

Configuring for manual mode operation is identical to configuring for normal mode (as described in Configuring for Normal Mode) except for the weight values in the server line entries. Here, the desired weight value of each server should be entered. Weight value is a decimal number between 1.0 and 10.0. A weight of 1.0 should be assigned to the server receiving the least amount of connections. Each of the other servers is assigned a weight that is relative to the server weighing 1.0. Weight value is described in detail in Manual Load Balancing. In the example below, there are five servers in the cluster. SERVER1 has a weight of 1.5. SERVER4 has a weight of 3.0. SERVER2, SERVER3 and SERVER5 each has a weight of 1.0

Here's an initialization file example for manual mode setup:

```
; DB2CADV Custom Adviser Configuration File
; Created: 2002/03/29

[50000]
server = 192.168.0.1 SERVER1 DB2NDADV xxxxxxxx 1.5
server = 192.168.0.2 SERVER2 DB2NDADV xxxxxxxx 1.0
server = 192.168.0.3 SERVER3 DB2NDADV xxxxxxxx 1.0
server = 192.168.0.4 SERVER4 DB2NDADV xxxxxxxx 3.0
server = 192.168.0.5 SERVER5 DB2NDADV xxxxxxxx 1.0
```

Related concepts:

- "DB2 Connect Custom Advisor concepts" on page 133

Setting up DB2 Connect Custom Advisor

Procedure (Windows):

To set up the DB2 Connect Custom Advisor:

1. Create a user's account for the Dispatcher to start in. In order for DB2 Connect Custom Advisor to access its dynamic link library file, IBM Network Dispatcher service must start in an account with User authority.
2. Change the Startup profile for IBM Network Dispatcher Service. Set the IBM Network Dispatcher service to log on as the user created in Step 1. To change the Service Startup profile, click on **Start** and select **Settings –> Control Panel –> Services**.
3. Set up user accounts in DB2 Connect servers. A user's account with Administrator authority must be set up for each DB2 Connect server. The default DB2ADMIN accounts may be used or new accounts with Administrator authority may be set up. These accounts are necessary for DB2 Connect Custom Advisor to access the System Monitor.

Procedure (AIX):

To set up the DB2 Connect Custom Advisor:

1. Logon to the Dispatcher machine with an account with root authority. Dispatcher and Advisor operations may only be done from a root account.
2. Set up user accounts in DB2 Connect servers. A user's account with root authority must be set up for each DB2 Connect server. The default DB2ADMIN accounts may be used or new accounts with root authority may be set up. These accounts are necessary for DB2 Connect Custom Advisorto access the System Monitor.

Related concepts:

- "DB2 Connect Custom Advisor concepts" on page 133

Activating the DB2 Connnect Custom Advisor

Procedure:

After the Network Dispatcher and Manager are started, issue the advisor start command to start DB2 Connect Custom Advisor:

```
ndcontrol advisor start db2cadv 50000
```

This command starts the DB2 Connect Custom Advisor in port 50000. The advisor update interval, manager update interval and weight refresh cycle should be adjusted after starting DB2 Connect Custom Advisor. Update and refresh intervals should be set short enough to provide Dispatcher with up-to-date information for effective load balancing.

More information on configuring custom advisors can be found in *IBM Network Dispatcher User's Guide* – Chapter 8. Advanced Dispatcher and CBR Functions. DB2 Connect Custom Advisor and the Dispatcher can be started simultaneously with a script file.

For more information on using script files for IBM Network Dispatcher, consult *IBM Network Dispatcher User's Guide* – Chapter 5. Configuring the Dispatcher Component.

Related concepts:

- "DB2 Connect Custom Advisor concepts" on page 133

Troubleshooting DB2 Connect Custom Advisor

After starting DB2 Connect Custom Advisor, it is a good practice to check the log file for possible error messages.

Procedure:

The log file db2cadv_50000.log[2] can be found in C:\Program Files\ibm\nd\dispatcher\logs\ for Windows (or where the dispatcher\logs\ is located) and in /usr/lpp/nd/dispatcher/logs for AIX.

For non-critical error conditions, DB2 Connect Custom Advisor will print an error message to the log file and return a neutral load of 500 to Dispatcher. A non-critical error will not make a server unavailable. Its load will remain neutral until the non-critical error is fixed and the Dispatcher is restarted.

Related concepts:

- "DB2 Connect Custom Advisor concepts" on page 133

Sample Dispatcher startup script file

```
@echo off
rem
rem START UP FILE FOR IBM NETWORK DISPATCHER
rem AND DB2 CONNECT CUSTOM ADVISOR FOR LOAD-
```

[2] If you choose to run your DB2 Connect servers on another port, the log file name will be db2cadv_YourPort.log

```
rem BALANCING A CLUSTER OF TWO DB2 CONNECT SERVERS
rem
rem filename = ndstart.cmd
rem created  = 2000/04/13

call ndcontrol executor start

set NFA=192.168.0.10
set CLUSTER=192.168.0.6

echo "Loading the non-forwarding address..."
call ndcontrol executor set nfa %NFA%

call ndcontrol executor set fintimeout 30
call ndcontrol executor set fincount 4000

echo "Loading Cluster Address..."
call ndcontrol cluster add %CLUSTER%

echo "Assigning Port to CLUSTER: %CLUSTER%..."
call ndcontrol port add %CLUSTER%:50000

set SERVER1=192.168.0.1
set SERVER2=192.168.0.11

echo "Adding server machines..."
call ndcontrol server add %CLUSTER%:50000:%SERVER1%+%SERVER2%

echo "Starting the manager..."
call ndcontrol manager start

echo "Start DB2 Connect Custom Advisor on port 50000..."
call ndcontrol advisor start db2cadv 50000

echo "Setting the manager proportions..."
call ndcontrol manager proportions 20 20 60 0

echo "Setting alias for cluster..."
call ndcontrol cluster configure %CLUSTER% en0 255.255.255.0

echo "Configuring Dispatcher Manager..."
call ndcontrol manager loglevel 1
call ndcontrol manager logsize 500000
call ndcontrol manager sensitivity 5.000000
call ndcontrol manager interval 3
call ndcontrol manager refresh 3

echo "Configuring DB2 Connect Custom Advisor..."
call ndcontrol advisor interval db2cadv 50000 3
  call ndcontrol advisor loglevel db2cadv 50000 1
call ndcontrol advisor logsize db2cadv 50000 500000
call ndcontrol advisor timeout db2cadv 50000 unlimited
```

Related concepts:

- “DB2 Connect Custom Advisor concepts” on page 133

Chapter 13. Performance

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 host or iSeries® 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, and `link` for SNA are examples of software resources.

Data Flows:

Figure 11 on page 146 shows the path of data flowing between the host or iSeries database server and the workstation through DB2 Connect.

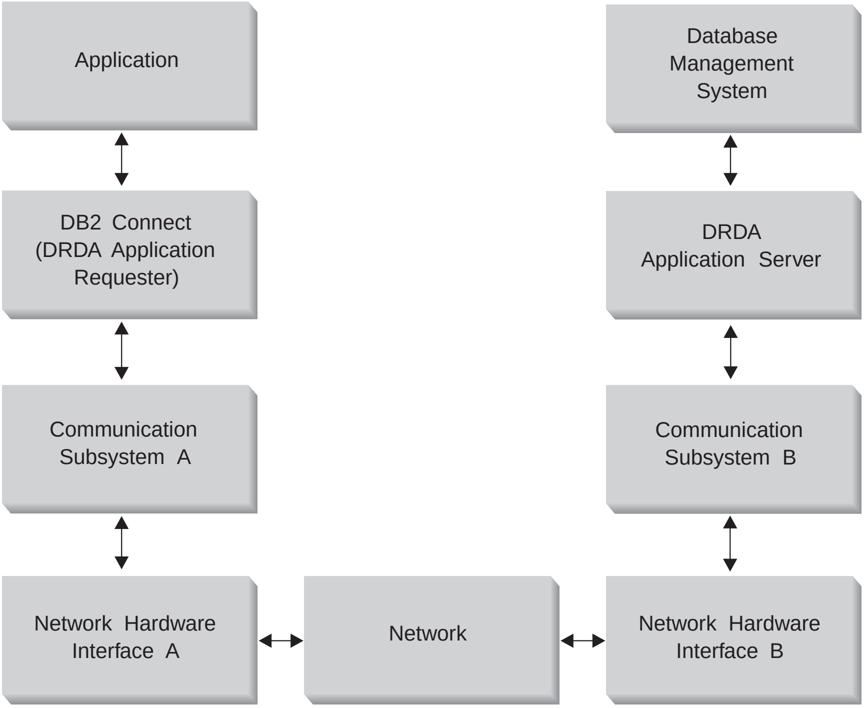

Application
Database Management System
DB2 Connect (DRDA Application Requester)
DRDA Application Server
Communication Subsystem A
Communication Subsystem B
Network Hardware Interface A
Network
Network Hardware Interface B

Figure 11. Data Flows in DB2 Connect

- The host or iSeries 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 may 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 iSeries 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. Communication subsystem A might be a product such as System Network Architecture (SNA), IBM® SNA Server for AIX, or SNAplus2 for HP-UX.
- DB2 Connect and the communication subsystem A are usually located on the same system. For the scope of this discussion, we assume 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 off-load 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 that those being measured. Unless the scenario includes some amount of 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 table lists 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 14. Performance Tools

System	Tool	Description
	CPU and memory usage	
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® NT and Windows 2000	Microsoft® Performance Monitor	
	Database activity	
All	Database monitor	Determines if the problem originates from the database.
OS/390® or zSeries™	DB2PM (IBM), OMEGAMON/DB2 (Candle), TMON (Landmark), INSIGHT (Goal Systems) and DB2AM (BMC)	
Windows NT® and Windows 2000	Microsoft Performance Monitor	
	Network activity	
AIX	netpmon	Reports low level network statistics, including TCP/IP and SNA statistics such as the number of packet or frames received per second.
DOS	Token-Ring Network 16/4 Trace and Performance Program	Most network monitors are platform dependent; this tool works for token-ring only.
Network controller such as 3745	NetView® Performance Monitor	Reports utilization of communication control and VTAM.
UNIX-based	netstat	Handles TCP/IP traffic.

Related concepts:

- "Application design" on page 150
- "DB2 Connect tuning" on page 163

- “Host database tuning” on page 165
- “Network tuning considerations” on page 165
- “DB2 Connect performance troubleshooting” on page 168
- “System resources contention” on page 167
- “SNA performance tuning hints and tips” on page 173
- “CLI/ODBC application performance tuning with the CLISCHEMA keyword” on page 88
- “db2ocat catalog optimizer tool” on page 86
- “Connection concentrator” on page 156
- “Connection pooling” on page 153
- “OSA-2 enhancements” on page 175
- “Multi Path Channel support for SNA over ESCON” on page 186
- “Network hardware” on page 187
- “Selection and tuning of the network attachment” on page 174
- “Tuning DB2 Connect connections using NCP” on page 169
- “Tuning DB2 for OS/390 and z/OS” on page 172

Related tasks:

- “Optimizing ODBC access” on page 149

Optimizing ODBC access

DB2 Universal Database provides special optimization designed to improve communication performance through ODBC. These enhancements are available to Microsoft Access, Lotus Approach, or Visual Basic. You can gain the benefit of faster ODBC throughput using DB2’s Configuration Assistant (CA).

Procedure:

To activate the optimized ODBC:

- If you are defining a new connection:
  1. Start the DB2 CA.
  2. Select the database alias that you want to optimize.
  3. Click **Properties**.
  4. Ensure that the **Register this Database for ODBC** check box is selected.
  5. Select the radio button that describes how you would like to register this database.
  6. Click **Settings**.

7. Click **Optimize** and select the application whose OBDC settings you wish to optimize.
8. Click **OK** and exit the CA.

- If you are updating an existing connection:
  1. Start the DB2 CA.
  2. Select the database alias that you want to optimize.
  3. Click **Properties**.
  4. Click **Settings**.
  5. Click **Optimize** from the CLI/ODBC Settings Window and select the application for which you want to optimize.
  6. Click **OK** and exit the CA.

Related concepts:

- "DB2 Connect performance considerations" on page 145

Application design

When you create an application, you can improve performance in several ways.

Compound SQL and stored procedures

For applications that send and receive many commands and replies, network overhead can be significant. Compound SQL and stored procedures are two ways to reduce this overhead.

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 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 overhead 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 overhead of both the host or iSeries® database server and the DB2 Connect server. There is fixed amount of CPU and network overhead 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:

Updateable SELECT statements (using `UPDATE/DELETE WHERE CURRENT OF` statements) are non-blocking queries, so you should use them only when absolutely necessary.

An updateable 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.

For a complete discussion of blocking when using SNA connections, refer to the *DRDA Connectivity Guide*.

Static and dynamic SQL

Use static SQL as much as possible. It avoids run-time 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 may 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 may 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 overhead of loading libraries and setting up internal data structures. For this reason, using type 2 connect may improve performance for applications that access more than one database.

Related concepts:

- "DB2 Connect performance considerations" on page 145

Connection Management

Connection pooling

DB2® Connect Enterprise Edition servers 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.

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. To reduce this overhead, DB2 Connect Enterprise Edition uses connection pooling to maintain open connections to the database in a readily accessible pool.

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 database connection and 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 of 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. In DB2 Connect Version 6 and later connection pooling is activated by default. 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 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 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 overhead 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_POOLAGENTSconfiguration parameter. This parameter equals the maximum number of idle agents you wish the system to maintain. Setting this parameter to zero is equivalent to turning off the connection pooling feature.

DB2 Connect does not establish connections to the database before receiving its first client request. If you wish, however, you may fill the pool of idle agents before any clients make a request. The pool can be filled on start-up using the NUM_INITAGENTSconfiguration parameter. This parameter determines how many idle agents should be created at start up time. These idle agents will not initially 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 db2 registry variable DB2CONNECT_IN_APP_PROCESS allows applications running on the same machine as DB2 Connect EE to either have DB2 Connect run within the applications process, default behavior, or to have the application connect to the DB2 Connect EE Server 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 EE Server agents and thus DB2CONNECT_IN_APP_PROCESS must be set to NO.

S/390 APARs required for Connection Pooling:

- DB2 for OS/390® Version 5.1 requires that fixes for APAR PQ24584 be applied.
- DB2 for OS/390 Version 6.1 requires that APARs PQ26179 and PQ33473 be applied.

For more information on these APARs go to IBM® server support web site at http://techsupport.services.ibm.com/server/support and search on these APAR numbers.

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 etc. 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 etc.

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.

Related concepts:

- "DB2 Connect performance considerations" on page 145
- "Connection concentrator" on page 156
- "Connection pooling and connection concentrator" on page 162

Connection concentrator

The connection concentrator reduces the resources required on DB2® for OS/390® and z/OS™ database servers to support large numbers of workstation and web users. This function can dramatically increase the scalability of your DB2 for OS/390 and z/OS and DB2 Connect™ solution while also providing for fail-safe operation and transaction level load balancing in DB2 for OS/390 and 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 Enterprise Edition servers to provide support to thousands of users simultaneously executing business transactions, while drastically reducing resources required on the S/390® host or iSeries™ database servers. It accomplishes this goal by concentrating the workload from all applications in a much smaller number of S/390 host or iSeries database server connections. While this may seem similar to the connection pooling function described above it is in fact a more sophisticated approach to reducing resource consumption for very high volume OLTP (On-line Transaction Processing) applications.

Connection concentrator introduces a concept of Logical Agent (LA) which handles user context while coordinating agent (CA) continues to own DB2 connection and thread. When a new application user connects he/she is assigned an LA. CA is needed to pass SQL to DB2 so one is assigned as soon as new transaction is initiated. To key to this architecture is the fact that CA is disassociated from the LA and is returned to the pool when transaction completes (commit/rollback). Another key feature is the method of assigning CAs to new transactions in a data sharing environments. DB2 Connect implements a sophisticated scheduling algorithm that uses OS/390 and z/OS Work Load Manager (WLM) information 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 those that are up and underutilized. DB2 Connect connection concentrator is activated when you set the number of maximum logical agents higher than the number of coordinating agents.

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.

Connection concentrator, on the other hand, 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 increase, and context switching between agents results in additional overhead.

In the above architecture, 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 >= 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 transaction boundary end the association and return to the available pool.

An entity known as the *logical agent scheduler* assigns worker agents to logical agents. Limitations in the number of open file handles on certain computing platforms may result in more than one scheduler instance when the number of logical agents exceeds the file handle limit.

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.

- The connection concentrator can only be used by DB2 Version 7 or later clients.
- Only DB2 for OS/390 and z/OS or DB2 for iSeries hosts are supported by the concentrator.
- 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 SNA, but you will not be able to use its XA concentration features with that connection.

- You should not use the static SET statement in your client applications if the concentrator is enabled on the gateway. DB2 will not return an error if you use static SET, but your application and any other applications that share the same outbound connection could be adversely affected.
- For SET statements, only immediate execution is supported.
- If you declare global temp tables, they must be closed explicitly at transaction or branch boundary. Failure to close the tables may result in an error during later transactions.
- For XA tightly coupled transaction support, all applications that participate in the same XA transaction must use the same gateway to connect to the host.
- Only applications that close any withhold cursors transaction boundary 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.
- All applications participating in the same XA transaction must have the same CCSID and use the same userid 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 only supports dynamic SQL from the command line interface. 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.

Activating the connection concentrator:

To use the connection concentrator, the following APAR must be applied to DB2 for OS/390 and z/OS Version 6.1:

```
APAR PQ33473
```

The database manager configuration parameter MAX_CONNECTIONS 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.

Several existing configuration parameters are used to configure agents. These parameters are as follows:

MAXAGENTS
Maximum number of worker agents.

MAX_COORDAGENTS
Maximum number of active coordinator agents.

NUM_POOLAGENTS
Agents pool size. The agent pool includes inactive agents and idle agents.

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 OS/390 and z/OS and DB2 for iSeries. 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 may 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_AGENTS      =  1,000
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 above example, worker agents will constantly form and break associations to logical agents. Those agents that are not idle may 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, we may assume that a TP Monitor is being used with a DB2 Connect gateway and an zSeries™ or iSeries 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. Let us 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 may 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 may 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.

Related concepts:

- "DB2 Connect performance considerations" on page 145
- "Connection pooling" on page 153
- "Connection pooling and connection concentrator" on page 162

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 overhead of database connections and handle connection volume. Connection concentrator helps increase the scalability of your DB2® for OS/390® and 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 then 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 then 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. Connection pooling, on the other hand, 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.

Related concepts:

- "DB2 Connect performance considerations" on page 145
- "Connection concentrator" on page 156
- "Connection pooling" on page 153

DB2 Connect tuning

DB2 Connect 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 may 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 may 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 monitor tools (such as the vmstat tool for UNIX-based 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 overhead 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.

Other DB2 Connect parameters:

MAXDARI and NUMDB should be set to their minimum values if there is no local database on the DB2 Connect workstation. These settings will minimize resource consumption.

AGENTPRI applies only with remote clients. AGENTPRI controls the priority given by the operating system scheduler to agents of a DB2 Connect instance. The DB2 Connect instance is granted more CPU cycles if it has a higher priority (lower number). This reduces the number of CPU cycles left for other processes executing on the DB2 Connect workstation. For example, you could have a high-priority DB2 Connect instance and a low-priority DB2 Connect instance running on the same workstation with different AGENTPRI values.

Every connection from a client machine to a host or iSeries® database server through DB2 Connect requires an agent running on the DB2 Connect workstation. Set MAXAGENTS to a value greater than or equal to the peak number of remote client connections accessing a host or iSeries database server through the DB2 Connect workstation.

If you decide to use accounting strings, using the sqlesact() API has performance advantages over the DB2ACCOUNT environment variable method.

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.

Related concepts:

- "DB2 Connect performance considerations" on page 145

Host database tuning

System performance will be affected by the performance of the host or iSeries™database server database. Different database management systems have different performance features. SQL optimizers of different systems, for example, could behave differently with the same application. Check your host or iSeries database server system performance documentation for more information.

For DB2® UDB for OS/390® and z/OS, you may be able to improve performance by using the uncommitted read (UR) or no commit (NC) bind options to avoid journalling.

Note: When using UR, unjournalled 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 may 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.

Related concepts:

- "DB2 Connect performance considerations" on page 145

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.

For SNA connections, you should set the Mode Profile of any workstation software to 63. In general, receive pacing values throughout the network should be set to their highest value, so the VPACING and PACING parameters on the DB2® APPL statement, and the PU/LU for the workstation in a switched major mode should also be set to 63. What this will do is allow the amount of message flows before the sender must wait for a response to increase progressively.

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 the following table.

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.
	RUSIZE	Optimum size is 4096.	Setting RUSIZE and RQRIOBLK to same size may give best performance.
	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.
	DLC Windowing on SNA	Set transmit window size high (>7). Set receive window size low (for example, at 1), test and increment repeatedly to find ideal value.	Every logical device adds delays. Simply network topology as much as possible.
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.
	Send/Receive Space Sizes	Should be 64K for both.	Default is only 8192 for Windows. Can be set in the Windows® registry.

Related concepts:

- "DB2 Connect performance considerations" on page 145

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 overhead.
- 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?

Related concepts:

- "DB2 Connect performance considerations" on page 145
- "DB2 Connect performance troubleshooting" on page 168

DB2 Connect performance troubleshooting

If DB2 Connect users are experiencing long response times during large queries from host or iSeries® servers, 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 host or iSeries 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. If VTAM® is used in the connection to the host or iSeries server, look under ″switched major node″ configuration for the value of the PACING parameter. On the DB2 Connect workstation, examine the communication setup of the ″LU 6.2 Mode Profile″ for IBMRDB mode definition. In this definition, ensure the value for the ″Receive pacing window″ parameter is less than or equal to the PACING value defined on VTAM. A common value for ″Receive pacing window″ on the DB2 Connect workstation and ″PACING″ on VTAM is 8.
3. Ensure the maximum RU size defined in the IBMRDB mode definition is set to a suitable value. We recommend 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 may be a limiting factor.
4. Consult with the VTAM administrator in your environment to ensure that VTAM is using ″adaptive pacing″ in LU-LU sessions with your DB2 Connect workstation.

Related concepts:

- "DB2 Connect performance considerations" on page 145

Tuning DB2 Connect connections using NCP

A typical network configuration might be:

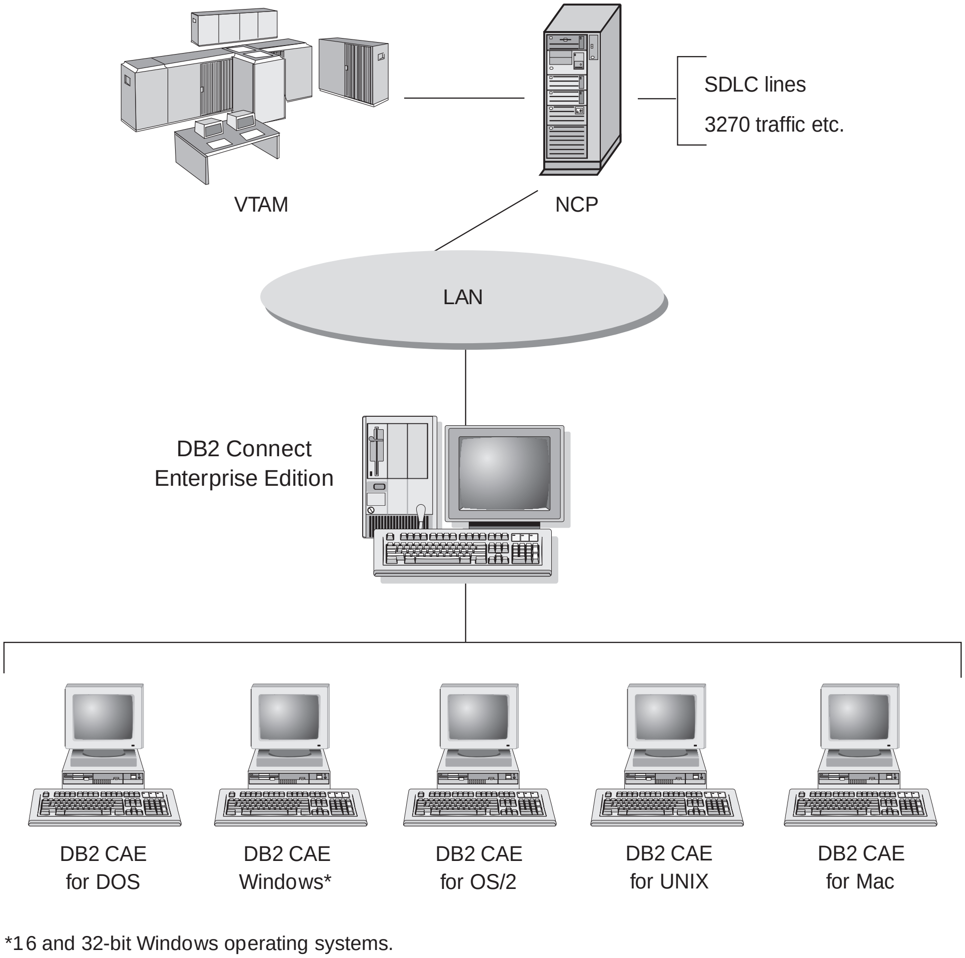


Figure 12. DB2 Connect Enterprise Edition server SNA network scenario

This scenario focuses on the throughput and response time between the host or iSeries™ database server to the DB2 Connect Enterprise Edition server and various parameters that could affect this.

Tuning criteria:

The suggested order in which to make these changes is:

```
1 - DELAY on PCCU macro*
2 - DLC/LLC Tuning*
3 - PIU size*
4 - Pacing window changes*
5 - DELAY on LINE macro*
6 - MAXBFRU changes
7 - LAN Frame sizes

* Major improvement in throughput is possible
```

PIU size (RU + 29 bytes)

The RU size at the host and the DB2® Connect server should be maximized. This implies that the RU size should be large enough to contain the API crossing (both SEND and RECEIVE data for the transaction where possible) in order to minimize the number of times VTAM® program stack must be traversed. Also, the network frame size may limit the maximum RU size if RU segmentation is not desired.

It is a good idea to set the DB2 Connect block size (RQRIOBLK), RU and pacing values such that RU * pacing >= RQRIOBLK. For example, the default RQRIOBLK size of 32K is a good value for most situations, and to exploit this you would set RU = 4K and receive window pacing to 8.

- RU size and pacing are set by the mode table which is defined on both DB2 Connect™ workstation and in VTAM. The mode table definitions should be the same in both places.
- RQRIOBLK is set using the DB2 UPDATE DBM CFG command.
- Network frame size I-frame is set in DLC configuration on the DB2 Connect workstation and in NCP.

Pacing window sizes

The session and VR pacing windows should be maximized: the largest value that does not cause network congestion or VR-held conditions, and so on, should be used. For a test environment set pacing to 0 (no pacing) or set it to the maximum value X'3F'.

Coat-tailing values (DELAY)

Coat-tailing is controlled by the DELAY parameter. The DELAY Parameter in the PCCU macro controls outbound coat-tailing (outbound with reference to the host). The DELAY value in the LINE definition statement for the NCP controls inbound coat-tailing (inbound with reference to the host).

The DELAY value determines how long a PIU is held in the queue (NCP or VTAM) before it is transmitted. The purpose of this wait is to increase the possibility that other PIUs will arrive in the interim and all of these can be transmitted on a single channel program. For the lowest latency, the DELAY value should be set to 0. Changing the value of the outbound coat-tailing delay value to 0 should have no noticeable effect on the host except for improved performance for outbound traffic. Some improvement in inbound traffic performance will also be realized.

Changing the DELAY on the NCP to 0 should be done with a little more care. The value can be set to 0 if the NCP is not overloaded and the inbound traffic does not consist of a significant percentage of small frames. Setting the values of DELAY to 0 may improve response time significantly, especially under light loads or test/benchmark environments.

```
VTAMB7   PCCU  CUADDR=CAF,
                                   AUTODMP=NO,
                                   AUTOIPL=NO,
                                   AUTOSYN=YES,
                                   BACKUP=YES,
                                   DELAY=0,
                                   VFYLM=YES,
                                   CHANCON=UNCOND,
                                   MAXDATA=32768,
                                   DUMPDS=NCPDUMP,
                                   OWNER=HOSTB7,
                                   SUBAREA=17
```

```
LNCTLS        GROUP   LNCTL=CA,CA=TYPE6,DELAY=0.0,TIMEOUT=500.0
CA0                 LINE         ADDRESS=00
PUCHAN0    PU            PUTYPE=5,TGN=1
CA1                 LINE         ADDRESS=01
PUCHAN1    PU            PUTYPE=5,TGN=1
```

DELAY considerations are documented in the *VTAM Network Implementation Guide*.

MAXBFRU

The MAXBFRU value should be set to a value two or three times larger than the largest PIU size.

DLC/LLC layer tuning

Ensure that the LLC2 window sizes (DLC send and receive window counts) between the NCP and the DB2 Connect Enterprise Edition gateway are the same. This has a significant effect specially when the server is DB2 Connect for AIX. It is recommended that the send window count be set higher than the receive window count.

In general, for any SNA connection across a Token-ring the LLC2 timers/windows should be optimized. In some cases, this change led to a six-fold improvement in throughput and response time.

LAN frame sizes

The token ring maximum frame size should be as large as possible.

Related concepts:

- "DB2 Connect performance considerations" on page 145
- "DB2 Connect performance troubleshooting" on page 168

Tuning DB2 for OS/390 and z/OS

OS/390® V1R3 is the minimum requirement for TCP/IP support. OS/390 V2R5 or later is highly recommended.

The Distributed Data Facility (DDF) is responsible for connecting distributed applications to DB2® for OS/390 and z/OS. The DDF should be set up as an application server. To do this, you can either insert the LU name of the remote system into the SYSIBM.LUNAMES table, or insert the LUNAME, SYSMODENAME, USERSECURITY, ENCRYPTPSWDS, MODESELECT, and USERNAMES values into the SYSIBM.SYSLUNAME table. Then perform a DDF update to the Boot Strap Data Set (BSDS). For example:

```
DDF LOCATION=LOC1,LUNAME=LU1,PORT=8000,RESPORT=8001
```

For best performance, you should use the recommended DDF address space prioritization (slightly lower or equal to DBM1 if you are in COMPAT mode). Use RACF® caching of authorizations in VLF, and use V5 package authorizations caching if you can. A value of CACHEPAC=32768 is sufficient for most operations.

Since DDF will try to connect to VTAM, VTAM® must be active when DDF starts. A sample VTAM APPL definition is included below:

```
SYD51TC* APPL  AUTH=(ACQ),                                       X
           PARSESS=YES,                                          X
           HAVAIL=YES,                                           X
           EAS=1600,                                             X
           APPC=YES,                                             X
           DSESLIM=1024,                                         X
           DMINWNL=512,                                          X
           DMINWNR=512,                                          X
           AUTOSES=1,                                            X
           SECACPT=ALREADYV,                                     X
           SRBEXIT=YES,                                          X
           SYNCLVL=SYNCPT,                                       X
           MODETAB=DB2MODET,                                     X
           VPACING=63                                            X
```

You can optimize inactive thread processing in OS/390 and z/OS. In V3, you are allowed up to 10,000 concurrently connected clients, and up to 25,000 in V4 and V5. 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.

Related concepts:

- "DB2 Connect performance considerations" on page 145

Related tasks:

- "Setting up DB2 as an application server (OS/390 and z/OS)" in the *Connectivity Supplement*
- "Setting up DB2 as an application requester (OS/390 and z/OS)" in the *Connectivity Supplement*

Additional SNA performance tuning

SNA performance tuning hints and tips

The performance characteristics of DB2 Connect are that it predominantly uses the processor and performs very little I/O. In general, the faster the processor speed, the faster DB2 Connect will run. DB2 Connect fully exploits SMP processor configurations.

A fast DB2 Connect Enterprise Edition server can handle an SQL request/reply pair in less than five milliseconds, not counting client time, network time, and processing time at the host or iSeries® server. A simple SQL statement or query with a few rows of data could be completed end-to-end in less than 0.1 seconds (from client to the host or iSeries server and back).

When there are more than four or five SQL statements in a query, then the use of stored procedures will help to ensure high OLTP performance and to avoid increases in lock contention due to network delays between SQL statements.

Performance problems are usually caused by to the type of host attachment in use, network routing and tuning characteristics, and application design.

Other DB2 Connect Performance Information Sources:

- Search the DB2® Technical Library web site at http://www.ibm.com/software/data/db2/library. Search the DB2 Universal Database™ library for ″Technotes″ with the keywords ″DB2CONNECT″ and ″Performance″ to find the latest information with a section on DB2 Connect considerations on the World Wide Web.

Related concepts:

- “DB2 Connect performance considerations” on page 145
- “DB2 Connect performance troubleshooting” on page 168

Selection and tuning of the network attachment

In order of likely best performance when using DB2 Connect, various types of network attachment include:

1. Channel attachment card
2. IBM® 3172 Model 3, or newer models, or equivalent
3. IBM 2216
4. Open System Adaptor Card (OSA-2, not OSA-1)
5. IBM 3745 with Network Control Program (NCP)
6. IBM 3174 Terminal Controllers, or equivalent (not recommended – see below)

The recommended best way to connect to the host is to use ESCON® channel attachment cards for AIX, Windows® NT or Windows 2000. The IBM 3172 Model 3 and 2216 also perform well, but they tend to deliver throughput inferior to ESCON.

When using AIX® with ESCON cards, please apply the PTFs related to MPC (Multi Path Channel). Without these PTFs the AIX SNA ESCON driver may deliver worse performance.

All the non-NCP specific recommendations are applicable to all types of DB2 Connect and client/server attachments.

The OSA-2 card on System/390® or zSeries might not deliver throughput as high as a 3272 Model 3 when there is a high volume of small transactions, owing to its lower frames-per-second capability.

3145 with NCP is usually tuned specifically for existing network traffic. Consequently it might not perform as well for database client/server applications. Most DB2 Connect performance problems are due to the time delay between the NCP and VTAM® and/or between NCPs.

In general, we recommend avoiding the use of 3174 Terminal Controllers because their packet size (RU size) of 256 bytes is too small. 3174 microcode level C is required in order to provide Independent LU support for APPC database connections. Some OEM 3174 equivalents may have similar dependencies.

Related concepts:

- “DB2 Connect performance considerations” on page 145

- “Network tuning considerations” on page 165
- “Network hardware” on page 187

OSA-2 enhancements

The following information is reproduced from the IBM® WSC Flash document number 9718.

```
TITLE: WSC  FLASH 9718:  OSA-2 ENHANCEMENTS AVAILABLE
DOCUMENT ID G023691 UNCLASSIFIED

 Open Systems Adapter 2 (OSA-2) Systems Network Architecture (SNA)
 enhancements are being made available earlier than previously announced.
 The enhancements are:

 o   SNA/APPN enhancements for OS/390, VM/ESA, and VSE/ESA™

     -   Enhanced availability: load balancing, redundancy, and overflow
     -   Enhanced connectivity: increased Physical Unit (PU) support
         (from 255 PUs per port to 2047 PUs per port).

 o   Support for ACF/VTAM® for VSE/ESA networks

 NOTE: These enhancements do not pertain to OSA-1.

 LOAD BALANCING, REDUNDANCY, AND OVERFLOW
 ________________________________________

 LOAD BALANCING: A single Medium Access Control (MAC) address can now be
 defined for attached OSA-2 SNA/APPN Physical Units (PUs), even though
 connections may be via multiple physical ports.  This support is offered for
 source-route bridged environments only (Token-Ring and FDDI). The number of
 sessions established through a port is monitored, and user session loads are
 evenly distributed across the equally configured ports.

 REDUNDANCY:  A secondary path between the LAN workstation and the host system
 can now be configured. If the primary path becomes unavailable, the secondary
 path will receive the LAN traffic. This increases system availability and
 simplifies network management.

 OVERFLOW:  User sessions flow through the primary OSA-2 port until the
 session capacity has been reached. Additional user sessions will
 automatically flow to the next OSA-2 port. Since all user workstations are
 identically configured, network administration is simplified and the network
 becomes more scalable. New users can be added non-disruptively.

 Load balancing, redundancy, and overflow support is provided by PTFs for
 OSA/SF as follows:

 o   OS/390®         - OW20205/UW34618        03/31/97
 o   VM/ESA®         - OW23952/UW37028        03/31/97
 o   VSE/ESA        - Provided with VSE/ESA V2.2.1      04/29/97
```

INCREASED PHYSICAL UNIT (PU) SUPPORT (VIA OSA/SF):

The architecture has been changed to allow up to a maximum of 2047 PUs per physical port to be defined for OSA-2 Ethernet, Token-Ring and FDDI features instead of the current 255 PUs per port. This enhancement is available for currently installed features, as well as new installations. Actual connectivity may vary based upon user workloads.

Increased Physical Unit (PU) Support is provided by PTFs for OSA/SF as follows:

o	OS/390	- OW23429/UW37210	03/31/97
o	VM/ESA	- OW24952/UW37028	03/31/97
o	VSE/ESA	- PQ03091/UQ04224	04/29/97

Increased Physical Unit (PU) Support is provided by PTFs for ACT/VTAM as follows:

o   ACF/VTAM for OS/390

- VTAM® 4.1 OW14043/UW24904
- VTAM 4.2 OW14043/UW24905
- VTAM 4.3 OW14043/UW24906

o   ACF/VTAM VM/ESA

- VM60877/UV59834

o   ACF/VTAM VSE/ESA

- DY44347/UD50254

VSE/ESA - SNA SUPPORT

OSA-2 and OSA/SF support is delivered via VSE/ESA Version 2 Release 2.1. This announcement of VSE/ESA support satisfies the Statement of General Direction contained in Hardware Announcement 196-194, and Hardware Announcement 196-193, dated September 10, 1996.

The OSA-2 feature provides ACF/VTAM for VSE/ESA host applications with direct access to Ethernet, Token-Ring, and FDDI LANs and Asynchronous Transfer Mode (ATM) Forum-compliant LAN emulation networks.

OSA/SF is available:

o   As a non-exclusive element of OS/390 Release 1 or above (5645-001)
o   As a separate program product, S/390® Open Systems Adapter Support Facility Version 1 Release 2 for MVS/ESA™ 4.3 or above (5655-104)
o   As a facility of VM/ESA Version 2 Release 2.0 (5654-030)
o   As a component of VSE Central Functions 6.1.1 in VSE/ESA Version 2 Release 2.1 (5690-VSE).

```
MORE™ INFORMATION
_______________

Announcements 297-043, 297-040
```

Related concepts:

- "DB2 Connect performance considerations" on page 145

Increasing DB2 Connect data transfer rates

In addition to blocking of rows for a query result set, DB2® for OS/390®and 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 OS/390 and 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 OS/390 and 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.

Related concepts:

- "DB2 Connect performance considerations" on page 145
- "Extra query block" on page 177
- "RFC-1323 Window scaling" on page 179

Extra query block

Extra query block support on servers with DB2® for z/OS™ Version 6.1 or above 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 the ’OPTIMIZE for N ROWS’ clause and/or the ’FETCH FIRST N ROWS ONLY’ clause on the select statement itself.
- With the ’OPTIMIZE for N ROWS’ clause, DB2 for OS/390 and 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 OS/390 and 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.
- DB2 Connect will tag on the ’OPTIMIZE for N ROWS’ clause for a DB2 for OS/390 and z/OS 6.x server. Even though the number of rows that could ultimately be returned for the query result set is not limited to N by DB2 for z/OS, CLI/ODBC would return SQL_NO_DATA_FOUND to the application if an attempt is made to fetch beyond N rows.
- The ’FETCH FIRST N ROWS ONLY’ clause is used instead for a DB2 for OS/390 and z/OS 7.1 or above server. Similar to the

embedded SQL case, the query result set is limited to N rows by DB2 for OS/390 and z/OS. Fetching beyond N rows would result in SQL_NO_DATA_FOUND.

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 OS/390 and 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 above server.

Related concepts:

- "DB2 Connect performance considerations" on page 145
- "Increasing DB2 Connect data transfer rates" on page 177
- "RFC-1323 Window scaling" on page 179

RFC-1323 Window scaling

Window scaling is supported as of DB2 Connect Version 7 FixPak 4 on all Windows® and UNIX® platforms that support the RFC-1323 extensions for TCP/IP. You can enable this feature on DB2® for Windows and 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 or UNIX, you can issue db2set DB2SORCVBUF =65537.

The maximum send and receive buffer sizes are dependent on the specific operating system. To ensure that buffer sizes configured have been accepted, the user can set the database manager configuration parameter DIAGLEVEL to 4 (informational) and check the db2diag.log file for messages.

For Window scaling to take effect it must be enabled on both ends of a connection; on both the workstation and the host, either directly through the operating system TCP/IP stack, or indirectly through the DB2 product. For instance, for DB2 for z/OS, Window Scaling can currently only be activated through the operating system by setting TCPRCVBUFRSIZE to any value above 64K. If you are using a remote DB2 client to access a host or iSeries 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 DB2 client and a workstation DB2 server when no host or iSeries 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 the white paper at http://www.networking.ibm.com/per/per10.html.

Related concepts:

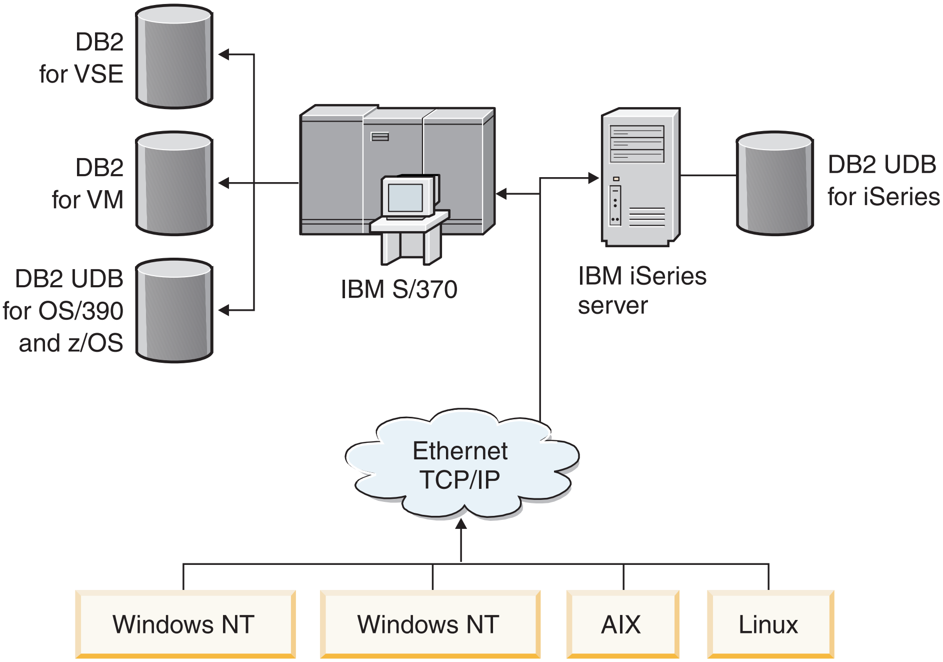
- "DB2 Connect performance considerations" on page 145
- "Increasing DB2 Connect data transfer rates" on page 177
- "Extra query block" on page 177

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, e-business, and spreadsheet applications require access to enterprise data. A reliable, fast and secure connection to host and iSeries™ databases must be established. This connection must be available 24/7 and be able to handle the high connection demands under critical load conditions. How can this connection be built?

High availability scenario:

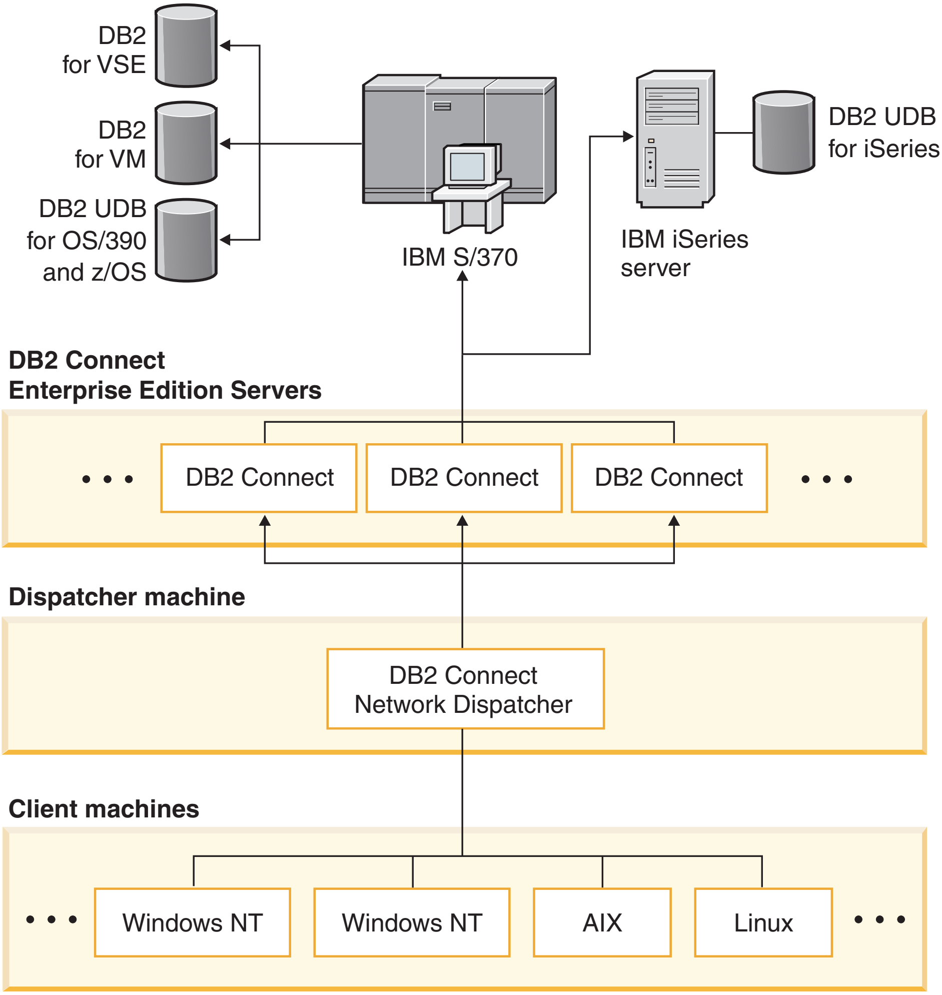
A company has several workstations and application servers running on Windows® and UNIX. These machines require access to data residing on several mainframe and iSeries databases. Applications running on these machines demand fast and reliable connections to the databases. The entire system is connected by an Ethernet network using TCP/IP.



For workstations and application servers to access host and iSeries databases, you need a connectivity component as an intermediary. This component must provide a highly available, robust, and fast connection to host and iSeries databases. It must also be scalable to anticipate for future growth in connection volume.

A solution using DB2 Connect EE, IBM Network Dispatcher, and DB2 Connect Custom Advisor:

A possible solution to this scenario can be built using IBM® DB2® Connect Enterprise Edition (EE), IBM Network Dispatcher and DB2 Connect™ Custom Advisor. All connection requests will be routed through the Network Dispatcher machine. This machine is setup with DB2 Connect EE, Network Dispatcher, and the DB2 Connect Custom Advisor. The Dispatcher machine efficiently distributes connection requests to the cluster of DB2 Connect EE servers. DB2 Connect EE provides a fast and secure connection to host and iSeries databases. The Network Dispatcher and the DB2 Connect EE servers all run on Windows NT® (and Windows 2000) platforms. The number of intermediate server machines depends on the volume of connections required by the clients.



The DB2 Connect Custom Advisor combines the power of DB2 Connect EE and Network Dispatcher to provide highly available connections from the clients to the host databases. DB2 Connect Custom Advisor is a light weight, Java-based extension to the SecureWay® Network Dispatcher. This advisor communicates with the DB2 Connect EE servers to retrieve information about the servers' health and connection workload.

Each DB2 System Monitor installed on the server provides this information. With the health and connection workload statistics of each DB2 Connect EE server, DB2 Connect Custom Advisor can calculate an accurate load for each server. The load information is passed to the Network Dispatcher for load balancing the cluster of DB2 Connect EE servers. Even under critical load conditions, the server workload will still be distributed accordingly.

IBM Network Dispatcher provides advanced IP level load balancing while remaining completely invisible to the clients. Through this intelligent load balancing mechanism, poor performance or even lost of connection due to unbalanced connection loads is virtually eliminated. If any one of the DB2 Connect EE servers goes down, the new connection requests will be made to the remaining healthy servers to ensure high availability.

How it works:

The Network Dispatcher load balances based on weights. Each DB2 Connect server in the cluster has an associated weight. The larger the weight, the more connections the server has to handle. The Dispatcher calculates the server weight using several parameters, one of them being the server load. This server load is determined by the DB2 Connect Custom Advisor.

During every scheduled interval, the DB2 Connect Custom Advisor attaches to one of the servers and takes a snapshot of its System Monitor status. From the snapshot, the Advisor can determine the number of connections the server has, the number of busy connections, the number of agents used, the number of communication errors and the number of idle DRDA® (Distributed Relational Database Architecture) agents. With these numbers, the Advisor can calculate an accurate load value that reflects closely to the actual workload of the server.

Once the load values from all servers are retrieved, the Dispatcher sets the weights and distributes the work accordingly. If the Advisor senses that a server is experiencing a critical workload, it temporarily marks the server as being down. New connections will not be routed to this server until the condition is resolved.

Advanced setup:

In addition to using the DB2 Connect Custom Advisor, it is also possible to simultaneously incorporate the Interactive Session Support (ISS) component of Network Dispatcher to aid in load balancing. ISS provides system related information such as CPU load and memory usage to the Dispatcher. The Dispatcher can then use both the DB2 Connect related information and the system related information to load balance the servers.

Extensibility:

As the number of connections grows, it may be necessary to add additional DB2 Connect servers in order to handle the additional traffic. The maximum number of servers allowable by the DB2 Connect Custom Advisor is only limited by the amount of memory available in the Dispatcher machine. The theoretical maximum number of servers allowable by IBM Network Dispatcher is a 32-bit number. In real world systems, neither limit will likely be reached.

Adding another DB2 Connect server requires no changes to the network architecture since all connection requests are routed to the Dispatcher machine as a single point of entry. Therefore, DB2 Connect along with Network Dispatcher and DB2 Connect Custom Advisor combine to provide a highly available, highly reliable and highly extensible solution for enterprise database connectivity to desktop clients.

Additional information:

IBM Network Dispatcher is packaged as a component of IBM WebSphere® Edge Server. For more information on Network Dispatcher visit the IBM Network Dispatcher web site at http://www.ibm.com/software/network/dispatcher/.

Related concepts:

- "DB2 Connect performance considerations" on page 145

Host data conversion

When data is transferred from one environment to another, it may need to be converted. This conversion can affect performance.

Consider the following platforms:

- Intel (Windows NT or Windows® 2000)
- IEEE (UNIX-based systems)
- System/370, System/390, zSeries™ (VM, VSE, OS/390, and z/OS)
- AS/400 and iSeries® (OS/400).

and the following types of numeric data:

- Packed decimal
- Zoned decimal
- Integer
- Floating point.

Table 15 shows when conversion takes place.

Table 15. Data Conversion

	Intel	**IEEE**	**S/370™ & S/390**	**iSeries**
Packed decimal data				
Intel	No	No	No	No
IEEE	No	No	No	No
S/370/390	No	No	No	No
OS/400	No	No	No	No
Zoned decimal data				
Intel	No	No	Yes	Yes
IEEE	No	No	Yes	Yes
S/370/390	Yes	Yes	No	No
OS/400	Yes	Yes	No	No
Integer data				
Intel	No	Yes	Yes	Yes
IEEE	Yes	No	No	No
S/370/390	Yes	No	No	No
OS/400	Yes	No	No	No
Floating point data				
Intel	No	Yes	Yes	Yes
IEEE	Yes	No	Yes	No
S/370/390	Yes	Yes	No	Yes
OS/400	Yes	No	Yes	No

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 may 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 host or iSeries™ database server.

Related concepts:

- "DB2 Connect performance considerations" on page 145
- "Data types for character data" on page 186

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.

Related concepts:

- "DB2 Connect performance considerations" on page 145
- "Host data conversion" on page 184

Multi Path Channel support for SNA over ESCON

Multi Path Channel (MPC) support for SNA over ESCON® allows a system running IBM® eNetwork Communications Server to use an ESCON adapter to create an MPC linkstation to the host. MPC is typically faster than CDLC because:

- MPC uses separate subchannels for read and write
- MPC is not limited by IOBUF size. Frames are 4K and may be blocked together.

Tests have shown as much as a threefold improvement on an MPC link compared to an ESCON Channel Data Link Control (CDLC) link with an IOBUF size less than 1K. AIX® SNA MPC requires ESCON and MVS™ VTAM® V4R4 or later and feature code 4024 of Communications Server for AIX (5765-652). Windows® NT systems must use IBM eNetwork Communications Server for Windows NT® Version 6.

The following are the Communications Server for AIX PTFs required for MPC:

APAR #	PTF #	LPP name
IX67032	U449693	sna.books.chdoc
IX67032	U449693	sna.books.escdoc
IX67032	U449300	sna.rte
IX67032	U450027	sna.msg.en_US.rte
IX65820	U447759	sna.dlcchannel
IX67618	U449691	mpc.rte
IX65813	U447758	devices.mca.8fc3.rte

Related concepts:

- "DB2 Connect performance considerations" on page 145

Network hardware

The following considerations relate to the hardware:

- Speed of the network or transmission media

  Performance improves with a faster transmission medium. For example, the following are 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 host or iSeries™ 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 SNA Network to another SNA 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 re-transmission.

Related concepts:

- “DB2 Connect performance considerations” on page 145

Chapter 14. Security

DB2 Connect security considerations for DB2 for OS/390 and z/OS

This topic describes DB2 Connect security considerations including authentication types and security settings. It also provides some additional hints and tips on security for DB2 for OS/390 and z/OS users.

Note: When using DB2 Connect with DCE security, DCE software is required on the DB2 client workstation and on the host or iSeries database server, but it is not necessary on the DB2 Connect server.

Related concepts:

- "DB2 Connect authentication considerations" on page 191
- "Security types supported with DB2 Connect" on page 192

Related reference:

- "Additional hints and tips about OS/390 and z/OS security" on page 189

Additional hints and tips about OS/390 and z/OS security

This topics provides some additional hints and tips about security for DB2 Connect connecting to a DB2 for OS/390 and z/OS database server.

Extended security field:

Ensure that the DB2 OS/390 and z/OS Extended Security Field is set to `YES`. This field appears in the DB2 for OS/390 and z/OS DSNTIPR panel.

Extended security codes:

Until DB2 Universal Database for OS/390 and z/OS Version 5.1, connect requests that provided user IDs or passwords could fail with `SQL30082` reason code 0, but no other indication as to what might be wrong.

DB2 Universal Database for OS/390 and z/OS Version 5.1 introduced an enhancement which provides support for extended security codes. Specifying extended security will provide additional diagnostics, such as `(PASSWORD EXPIRED)` in addition to the reason code.

To exploit this, the DB2 Universal Database for OS/390 and z/OS ZPARM installation parameter for extended security should be set to the value YES. Use the DB2 Universal Database for OS/390 and z/OS installation panel DSN6SYSP to set EXTSEC=YES. You can also use DDF panel 1 (DSNTIPR) to set this. The default value is EXTSEC=NO. In the case of an expired password, Windows, UNIX, and Web applications using DB2 Connect will receive an SQL30082 error message.

TCP/IP security already verified:

If you wish to provide support for the DB2 Universal Database security option AUTHENTICATION=CLIENT, then use DB2 Universal Database for OS/390 and z/OS installation panel DSNTIP4 (DDF panel 2) to set TCP/IP already verified security to YES.

Desktop ODBC and Java application security:

Workstation ODBC and Java applications use dynamic SQL. This may create security concerns in some installations. DB2 Universal Database for OS/390 and z/OS introduces a new bind option DYNAMICRULES(BIND) that allows execution of dynamic SQL under the authorization of either the owner or the binder.

DB2 Universal Database and DB2 Connect provide a new CLI/ODBC configuration parameter CURRENTPACKAGESET in the DB2CLI.INI configuration file. This should be set to a schema name that has the appropriate privileges. An SQL SET CURRENT PACKAGESET schema statement will automatically be issued after every connect for the application.

Use the ODBC Manager to update DB2CLI.INI.

Password change support:

If an SQL CONNECT statement returns a message indicating that the user ID's password has expired, with DB2 Connect it is possible to change the password without signing on to TSO. Through DRDA, DB2 Universal Database for OS/390 and z/OS can change the password for you.

The old password along with the new password and the verify password must be supplied by the user. A request to change the password is sent to the DB2 Universal Database for OS/390 and z/OS database server.

An additional benefit is that a separate LU definition is not required.

Related reference:

- "BIND" in the *Command Reference*

- “DB2 Connect security considerations for DB2 for OS/390 and z/OS” on page 189

DB2 Connect authentication considerations

As DB2 Connect administrator, in cooperation with your host or iSeries™ database administrator, you can determine where user names and passwords are validated:

- At the client
- At the host or iSeries server
- At a DCE security server
- Single sign-on and validation through a third-party system (Kerberos).

You determine where validation occurs by setting the authentication type parameter in the system database directory, and the security type parameter in the node directory for APPC or APPN® nodes.

Notes:

1. DB2 Connect itself performs no user validation. DB2® Connect passes all authentication information from the client to the server.

The following authentication types are allowed with DB2 Connect:

CLIENT
The user name and password are validated at the client.

SERVER
The user name and password are validated at the host or iSeries server database.

SERVER_ENCRYPT
As for SERVER authentication, the user name and password are validated at the host or iSeries database server, but the transferred passwords are encrypted at the client.

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.

DCE The user name and password are validated at the DCE security 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 his 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: If the remote client has not specified an authentication type, the client will default to SERVER_ENCRYPT. If this type is not accepted by the server, the client will attempt to retry 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.

Related concepts:

- "Security types supported with DB2 Connect" on page 192

Related reference:

- "Additional hints and tips about OS/390 and z/OS security" on page 189
- "DB2 Connect security considerations for DB2 for OS/390 and z/OS" on page 189

Security types supported with DB2 Connect

This topic lists the various combinations of authentication and security settings that are supported with DB2 Connect over both APPC and TCP/IP connections. The following applies to both types of connection.

Security types for APPC connections

The following security types are allowed for APPC connections to specify what security information will flow at the communications layer:

SAME Only the user name is passed to the host or iSeries™ database server.

PROGRAM

The user name and password are passed to the host or iSeries database server.

NONE

No security information flows.

Table 16 on page 193 shows the possible combinations of these values and the authentication type specified on the DB2 Connect server, and where validation is performed for each combination. Only the combinations shown in this table are supported by DB2 Connect over

APPC connections.

Table 16. Valid Security Scenarios for APPC connections

Scenario	Authentication setting in the database directory entry at the DB2® Connect server	Security	Validation
1	CLIENT	SAME	Client
2	SERVER	PROGRAM	Host or iSeries database server
3	SERVER_ENCRYPT	NONE	Host or iSeries database server
4	DCE	NONE	DCE security server
5	KERBEROS	NONE	Kerberos security

Notes:

1. For AIX® systems, all login users using APPC security type `SAME` must belong to the AIX system group.
2. For AIX systems with remote clients, the instance of DB2 Connect running on the DB2 Connect server must belong to the AIX system group.
3. Access to a host or iSeries database server is controlled by its own security mechanisms or subsystems. For example, the Virtual Telecommunications Access Method (VTAM) and Resource Access Control Facility (RACF). Access to protected database objects is controlled by the SQL GRANT and REVOKE statements.

Security types for TCP/IP connections

The TCP/IP communication protocol does not support security options at the network protocol layer. Thus only the authentication type controls where authentication takes place. Only the combinations shown in this table are supported by DB2 Connect over TCP/IP connections.

Table 17. Valid Security Scenarios for TCP/IP connections

Scenario	Authentication setting in the database directory entry at the DB2 Connect™ workstation	Validation
1	CLIENT	Client
2	SERVER	Host or iSeries server
3	SERVER_ENCRYPT	Host or iSeries server
4	DCE	DCE security server

Table 17. Valid Security Scenarios for TCP/IP connections (continued)

Scenario	Authentication setting in the database directory entry at the DB2 Connect™ workstation	Validation
5	KERBEROS	Kerberos security

Discussion of security types

The following discussion applies to both APPC and TCP/IP connections, as described above and listed in Table 16 on page 193 and Table 17 on page 193. 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 host or iSeries database server only. The userID and password is sent across the network from the remote client to the DB2 Connect server and from the DB2 Connect server to the host or iSeries database server.
- In scenario 3 is the same as scenario 2, except that the userID and password are encrypted.
- In scenario 4, a DCE encrypted ticket is obtained by the client from the DCE security server. The ticket is passed unaltered through DB2 Connect to the server, where it is validated by the server using DCE Security Services.
- In scenario 5, 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.

Related concepts:

- “DB2 Connect authentication considerations” on page 191

Related reference:

- “Additional hints and tips about OS/390 and z/OS security” on page 189
- “DB2 Connect security considerations for DB2 for OS/390 and z/OS” on page 189

Kerberos support

DB2® Universal Database currently supports the Kerberos security protocol as a means to authenticate users in a non-DRDA environment. Since DB2 for OS/390 and z/OS Version 7.1 supports Kerberos security, DB2 Connect™ has added DRDA® AR functionality to use Kerberos authentication to connect to DB2 for OS/390 and z/OS.

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 respectively. The Security Support Provider Interface (SSPI) provides a high level interface to the Kerberos SSP and other security protocols.

Communication protocol support:

For SNA connection, you must use SECURITY=NONE when cataloging the APPC node.

Typical setup:

To configure DB2 with Kerberos authentication, set up:

- An authorization policy for DB2 (as a service) in the Active Directory that is shared on a network, and
- Trust relationship between Kerberos Key Distribution Centers (KDCs)

In the simplest scenario, there is at least one KDC trust relationship to configure, that is, the one between the KDC controlling the client workstation, and the OS/390® or z/OS™ system. OS/390 or z/OS R10 provides Kerberos ticket processing through its RACF® facility which allows the host to act as an UNIX® KDC.

DB2 Connect provides as usual the router functionality in the 3-tier setting. It does not assume any role in authentication when Kerberos security is used. Instead, it merely passes the client's security token to DB2 for OS/390 and z/OS. Thus there is no need for the DB2 Connect gateway to be a member of the client or the host's Kerberos realm.

Downlevel compatibility:

DB2 requirements for Kerberos support:

DB2 UDB Client:
Version 7.1 (OS: Windows 2000)

DB2 Connect:
Version 7.1 + Fix Pack 1 (OS: Any)

DB2 UDB for OS/390 and z/OS:
Version 7.1

DB2 for OS/390 also have a requirement to be run on OS/390 Version 2 Release 10 or later. There are additional implied requirements on downlevel DB2 for OS/390 systems when connecting from DB2 Connect. Although these DB2 for OS/390 systems do not support Kerberos, they do not respond properly to unsupported DRDA SECMECs. To solve this problem, apply the proper PTF:

- UQ41941 (for DB2 for OS/390 Version 5.1)
- UQ41942 (for DB2 for OS/390 Version 6.1)

Related reference:

- "DB2 Connect security considerations for DB2 for OS/390 and z/OS" on page 189

Part 3. Appendixes

Appendix A. 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 from the host to the workstation, as well as the reverse (see Figure 13).

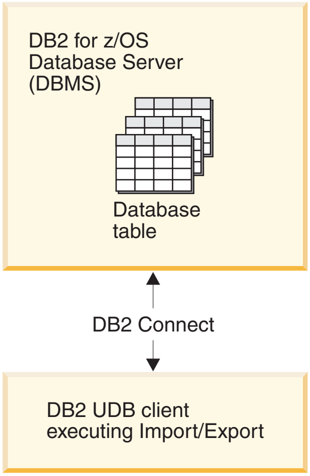


Figure 13. Import/Export through DB2 Connect

The DB2 export and import utilities allow you to move data from a host or AS/400 and iSeries 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 DB2 for MVS/ESA into a delimited ASCII file, and then import it into a DB2 for Windows database.

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

Notes:

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 SQL. Specify the `compound` file type modifier in the import utility to group a specified number of SQL statements into a block. This may reduce network overhead and improve response time.

Restrictions:

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.
- A commit count interval must not be specified for the import operation.

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:

- On systems that store data in EBCDIC (MVS, OS/390, OS/400, 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.

Moving Data from a workstation to a host server:

To move data to a host or AS/400 and iSeries 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 AS/400 and iSeries server database.

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

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

2. Issue the following command to establish a DRDA connection to the target DB2 UDB server:

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

3. If it doesn't already exit, create the target table on target DB2 UDB server_

   ```
   CREATE TABLE mydb.staff
   ```

4. To import the data issue the following command:

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

   One 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 RELAVIS.STAFF. Single rows will continue to be inserted until all of the data has been moved to the target table.

Related concepts:

- "Moving Data Across Platforms - File Format Considerations" in the *Data Movement Utilities Guide and Reference*

Related reference:

- "EXPORT" in the *Command Reference*
- "IMPORT" in the *Command Reference*

Appendix B. DB2 Universal Database technical information

Overview of DB2 Universal Database technical information

DB2 Universal Database technical information can be obtained in the following formats:

- Books (PDF and hard-copy formats)
- A topic tree (HTML format)
- Help for DB2 tools (HTML format)
- Sample programs (HTML format)
- Command line help
- Tutorials

This section is an overview of the technical information that is provided and how you can access it.

Categories of DB2 technical information

The DB2 technical information is categorized by the following headings:

- Core DB2 information
- Administration information
- Application development information
- Business intelligence information
- DB2 Connect information
- Getting started information
- Tutorial information
- Optional component information
- Release notes

The following tables describe, for each book in the DB2 library, the information needed to order the hard copy, print or view the PDF, or locate the HTML directory for that book. A full description of each of the books in the DB2 library is available from the IBM Publications Center at www.ibm.com/shop/publications/order

The installation directory for the HTML documentation CD differs for each category of information:

```
htmlcdpath/doc/htmlcd/%L/category
```

where:

- *htmlcdpath* is the directory where the HTML CD is installed.
- *%L* is the language identifier. For example, en_US.
- *category* is the category identifier. For example, core for the core DB2 information.

In the PDF file name column in the following tables, the character in the sixth position of the file name indicates the language version of a book. For example, the file name `db2d1e80` identifies the English version of the *Administration Guide: Planning* and the file name `db2d1g80` identifies the German version of the same book. The following letters are used in the sixth position of the file name to indicate the language version:

Language	**Identifier**
Arabic	w
Brazilian Portuguese	b
Bulgarian	u
Croatian	9
Czech	x
Danish	d
Dutch	q
English	e
Finnish	y
French	f
German	g
Greek	a
Hungarian	h
Italian	i
Japanese	j
Korean	k
Norwegian	n
Polish	p
Portuguese	v
Romanian	8
Russian	r
Simp. Chinese	c
Slovakian	7
Slovenian	l
Spanish	z
Swedish	s
Trad. Chinese	t
Turkish	m

No form number indicates that the book is only available online and does not have a printed version.

Core DB2 information

The information in this category cover DB2 topics that are fundamental to all DB2 users. You will find the information in this category useful whether you are a programmer, a database administrator, or you work with DB2 Connect, DB2 Warehouse Manager, or other DB2 products.

The installation directory for this category is doc/htmlcd/*%L*/core.

Table 18. Core DB2 information

Name	Form Number	PDF File Name
IBM DB2 Universal Database Command Reference	SC09-4828	db2n0x80
IBM DB2 Universal Database Glossary	No form number	db2t0x80
IBM DB2 Universal Database Master Index	SC09-4839	db2w0x80
IBM DB2 Universal Database Message Reference, Volume 1	GC09-4840	db2m1x80
IBM DB2 Universal Database Message Reference, Volume 2	GC09-4841	db2m2x80
IBM DB2 Universal Database What's New	SC09-4848	db2q0x80

Administration information

The information in this category covers those topics required to effectively design, implement, and maintain DB2 databases, data warehouses, and federated systems.

The installation directory for this category is doc/htmlcd/*%L*/admin.

Table 19. Administration information

Name	Form number	PDF file name
IBM DB2 Universal Database Administration Guide: Planning	SC09-4822	db2d1x80
IBM DB2 Universal Database Administration Guide: Implementation	SC09-4820	db2d2x80
IBM DB2 Universal Database Administration Guide: Performance	SC09-4821	db2d3x80
IBM DB2 Universal Database Administrative API Reference	SC09-4824	db2b0x80

Table 19. Administration information (continued)

Name	Form number	PDF file name
IBM DB2 Universal Database Data Movement Utilities Guide and Reference	SC09-4830	db2dmx80
IBM DB2 Universal Database Data Recovery and High Availability Guide and Reference	SC09-4831	db2hax80
IBM DB2 Universal Database Data Warehouse Center Administration Guide	SC27-1123	db2ddx80
IBM DB2 Universal Database Federated Systems Guide	GC27-1224	db2fpx80
IBM DB2 Universal Database Guide to GUI Tools for Administration and Development	SC09-4851	db2atx80
IBM DB2 Universal Database Replication Guide and Reference	SC27-1121	db2e0x80
IBM DB2 Installing and Administering a Satellite Environment	GC09-4823	db2dsx80
IBM DB2 Universal Database SQL Reference, Volume 1	SC09-4844	db2s1x80
IBM DB2 Universal Database SQL Reference, Volume 2	SC09-4845	db2s2x80
IBM DB2 Universal Database System Monitor Guide and Reference	SC09-4847	db2f0x80

Application development information

The information in this category is of special interest to application developers or programmers working with DB2. You will find information about supported languages and compilers, as well as the documentation required to access DB2 using the various supported programming interfaces, such as embedded SQL, ODBC, JDBC, SQLj, and CLI. If you view this information online in HTML you can also access a set of DB2 sample programs in HTML.

The installation directory for this category is doc/htmlcd/%*L*/ad.

Table 20. Application development information

Name	Form number	PDF file name
IBM DB2 Universal Database Application Development Guide: Building and Running Applications	SC09-4825	db2axx80
IBM DB2 Universal Database Application Development Guide: Programming Client Applications	SC09-4826	db2a1x80
IBM DB2 Universal Database Application Development Guide: Programming Server Applications	SC09-4827	db2a2x80
IBM DB2 Universal Database Call Level Interface Guide and Reference, Volume 1	SC09-4849	db2l1x80
IBM DB2 Universal Database Call Level Interface Guide and Reference, Volume 2	SC09-4850	db2l2x80
IBM DB2 Universal Database Data Warehouse Center Application Integration Guide	SC27-1124	db2adx80
IBM DB2 XML Extender Administration and Programming	SC27-1234	db2sxx80

Business intelligence information

The information in this category describes how to use components that enhance the data warehousing and analytical capabilities of DB2 Universal Database.

The installation directory for this category is doc/htmlcd/%*L*/wareh.

Table 21. Business intelligence information

Name	Form number	PDF file name
IBM DB2 Warehouse Manager Information Catalog Center Administration Guide	SC27-1125	db2dix80
IBM DB2 Warehouse Manager Installation Guide	GC27-1122	db2idx80

DB2 Connect information

The information in this category describes how to access host or iSeries data using DB2 Connect Enterprise Edition or DB2 Connect Personal Edition.

The installation directory for this category is doc/htmlcd/*%L*/conn.

Table 22. DB2 Connect information

Name	Form number	PDF file name
APPC, CPI-C, and SNA Sense Codes	No form number	db2apx80
IBM Connectivity Supplement	No form number	db2h1x80
IBM DB2 Connect Quick Beginnings for DB2 Connect Enterprise Edition	GC09-4833	db2c6x80
IBM DB2 Connect Quick Beginnings for DB2 Connect Personal Edition	GC09-4834	db2c1x80
IBM DB2 Connect User's Guide	SC09-4835	db2c0x80

Getting started information

The information in this category is useful when you are installing and configuring servers, clients, and other DB2 products.

The installation directory for this category is doc/htmlcd/*%L*/start.

Table 23. Getting started information

Name	Form number	PDF file name
IBM DB2 Universal Database Quick Beginnings for DB2 Clients	GC09-4832	db2itx80
IBM DB2 Universal Database Quick Beginnings for DB2 Servers	GC09-4836	db2isx80
IBM DB2 Universal Database Quick Beginnings for DB2 Personal Edition	GC09-4838	db2i1x80
IBM DB2 Universal Database Installation and Configuration Supplement	GC09-4837	db2iyx80
IBM DB2 Universal Database Quick Beginnings for DB2 Data Links Manager	GC09-4829	db2z6x80

Tutorial information

Tutorial information introduces DB2 features and teaches how to perform various tasks.

The installation directory for this category is doc/htmlcd/*%L*/tutr.

Table 24. Tutorial information

Name	Form number	PDF file name
Business Intelligence Tutorial: Introduction to the Data Warehouse	No form number	db2tux80
Business Intelligence Tutorial: Extended Lessons in Data Warehousing	No form number	db2tax80
Development Center Tutorial for Video Online using Microsoft Visual Basic	No form number	db2tdx80
Information Catalog Center Tutorial	No form number	db2aix80
Video Central for e-business Tutorial	No form number	db2twx80
Visual Explain Tutorial	No form number	db2tvx80

Optional component information

The information in this category describes how to work with optional DB2 components.

The installation directory for this category is doc/htmlcd/*%L*/opt.

Table 25. Optional component information

Name	Form number	PDF file name
IBM DB2 Life Sciences Data Connect Planning, Installation, and Configuration Guide	GC27-1235	db2lsx80
IBM DB2 Spatial Extender User's Guide and Reference	SC27-1226	db2sbx80
IBM DB2 Universal Database Data Links Manager Administration Guide and Reference	SC27-1221	db2z0x80

Table 25. Optional component information (continued)

Name	Form number	PDF file name
IBM DB2 Universal Database Net Search Extender Administration and Programming Guide **Note:** HTML for this document is not installed from the HTML documentation CD.	SH12-6740	N/A

Release notes

The release notes provide additional information specific to your product's release and FixPak level. They also provides summaries of the documentation updates incorporated in each release and FixPak.

Table 26. Release notes

Name	Form number	PDF file name	HTML directory
DB2 Release Notes	See note.	See note.	doc/prodcd/*%L*/db2ir where *%L* is the language identifier.
DB2 Connect Release Notes	See note.	See note.	doc/prodcd/*%L*/db2cr where *%L* is the language identifier.
DB2 Installation Notes	Available on product CD-ROM only.	Available on product CD-ROM only.	

Note: The HTML version of the release notes is available from the Information Center and on the product CD-ROMs. To view the ASCII file:

- On UNIX-based platforms, see the Release.Notes file. This file is located in the DB2DIR/Readme/*%L* directory, where `%L` represents the locale name and `DB2DIR` represents:
  - /usr/opt/db2_08_01 on AIX
  - /opt/IBM/db2/V8.1 on all other UNIX operating systems
- On other platforms, see the RELEASE.TXT file. This file is located in the directory where the product is installed.

Related tasks:

- "Printing DB2 books from PDF files" on page 211

- “Ordering printed DB2 books” on page 212
- “Accessing online help” on page 212
- “Finding product information by accessing the DB2 Information Center from the administration tools” on page 216
- “Viewing technical documentation online directly from the DB2 HTML Documentation CD” on page 217

Printing DB2 books from PDF files

You can print DB2 books from the PDF files on the *DB2 PDF Documentation* CD. Using Adobe Acrobat Reader, you can print either the entire book or a specific range of pages.

Prerequisites:

Ensure that you have Adobe Acrobat Reader. It is available from the Adobe Web site at www.adobe.com

Procedure:

To print a DB2 book from a PDF file:

1. Insert the *DB2 PDF Documentation* CD. On UNIX operating systems, mount the DB2 PDF Documentation CD. Refer to your *Quick Beginnings* book for details on how to mount a CD on UNIX operating systems.
2. Start Adobe Acrobat Reader.
3. Open the PDF file from one of the following locations:
   - On Windows operating systems:

     x:\doc\language directory, where *x* represents the CD-ROM drive letter and *language* represents the two-character territory code that represents your language (for example, EN for English).
   - On UNIX operating systems:

     /cdrom/doc/*%L* directory on the CD-ROM, where */cdrom* represents the mount point of the CD-ROM and *%L* represents the name of the desired locale.

Related tasks:

- “Ordering printed DB2 books” on page 212
- “Finding product information by accessing the DB2 Information Center from the administration tools” on page 216
- “Viewing technical documentation online directly from the DB2 HTML Documentation CD” on page 217

Related reference:

- “Overview of DB2 Universal Database technical information” on page 203

Ordering printed DB2 books

Procedure:

To order printed books:

- Contact your IBM authorized dealer or marketing representative. To find a local IBM representative, check the IBM Worldwide Directory of Contacts at www.ibm.com/shop/planetwide
- Phone `1-800-879-2755` in the United States or `1-800-IBM-4YOU` in Canada.
- Visit the IBM Publications Center at www.ibm.com/shop/publications/order

Related tasks:

- “Printing DB2 books from PDF files” on page 211
- “Finding topics by accessing the DB2 Information Center from a browser” on page 214
- “Viewing technical documentation online directly from the DB2 HTML Documentation CD” on page 217

Related reference:

- “Overview of DB2 Universal Database technical information” on page 203

Accessing online help

The online help that comes with all DB2 components is available in three types:

- Window and notebook help
- Command line help
- SQL statement help

Window and notebook help explain the tasks that you can perform in a window or notebook and describe the controls. This help has two types:

- Help accessible from the **Help** button
- Infopops

The **Help** button gives you access to overview and prerequisite information. The infopops describe the controls in the window or notebook. Window and notebook help are available from DB2 centers and components that have user interfaces.

Command line help includes Command help and Message help. Command help explains the syntax of commands in the command line processor. Message help describes the cause of an error message and describes any action you should take in response to the error.

SQL statement help includes SQL help and SQLSTATE help. DB2 returns an SQLSTATE value for conditions that could be the result of an SQL statement. SQLSTATE help explains the syntax of SQL statements (SQL states and class codes).

Note: SQL help is not available for UNIX operating systems.

Procedure:

To access online help:

- For window and notebook help, click **Help** or click that control, then click **F1**. If the **Automatically display infopops** check box on the **General** page of the **Tool Settings** notebook is selected, you can also see the infopop for a particular control by holding the mouse cursor over the control.
- For command line help, open the command line processor and enter:
  - For Command help:

    ```
    ? command
    ```

    where *command* represents a keyword or the entire command.

  For example, `? catalog` displays help for all the CATALOG commands, while `? catalog database` displays help for the CATALOG DATABASE command.
- For Message help:

  ```
  ? XXXnnnnn
  ```

  where *XXXnnnnn* represents a valid message identifier.

  For example, `? SQL30081` displays help about the SQL30081 message.
- For SQL statement help, open the command line processor and enter:
  - For SQL help:

    ```
    ? 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, while `? 08` displays help for the 08 class code.
  - For SQLSTATE help:

```
help statement
```

where *statement* represents an SQL statement.

For example, help SELECT displays help about the SELECT statement.

Related tasks:

- "Finding topics by accessing the DB2 Information Center from a browser" on page 214
- "Viewing technical documentation online directly from the DB2 HTML Documentation CD" on page 217

Finding topics by accessing the DB2 Information Center from a browser

The DB2 Information Center accessed from a browser enables you to access the information you need to take full advantage of DB2 Universal Database and DB2 Connect. The DB2 Information Center also documents major DB2 features and components including replication, data warehousing, metadata, Life Sciences Data Connect, and DB2 extenders.

The DB2 Information Center accessed from a browser is composed of the following major elements:

Navigation tree
The navigation tree is located in the left frame of the browser window. The tree expands and collapses to show and hide topics, the glossary, and the master index in the DB2 Information Center.

Navigation toolbar
The navigation toolbar is located in the top right frame of the browser window. The navigation toolbar contains buttons that enable you to search the DB2 Information Center, hide the navigation tree, and find the currently displayed topic in the navigation tree.

Content frame
The content frame is located in the bottom right frame of the browser window. The content frame displays topics from the DB2 Information Center when you click on a link in the navigation tree, click on a search result, or follow a link from another topic or from the master index.

Prerequisites:

To access the DB2 Information Center from a browser, you must use one of the following browsers:

- Microsoft Explorer, version 5 or later
- Netscape Navigator, version 6.1 or later

Restrictions:

The DB2 Information Center contains only those sets of topics that you chose to install from the *DB2 HTML Documentation CD*. If your Web browser returns a `File not found` error when you try to follow a link to a topic, you must install one or more additional sets of topics *DB2 HTML Documentation CD*.

Procedure:

To find a topic by searching with keywords:

1. In the navigation toolbar, click **Search**.
2. In the top text entry field of the Search window, enter two or more terms related to your area of interest and click **Search**. A list of topics ranked by accuracy displays in the **Results** field.

   Entering more terms increases the precision of your query while reducing the number of topics returned from your query.
3. In the **Results** field, click the title of the topic you want to read. The topic displays in the content frame.

To find a topic in the navigation tree:

1. In the navigation tree, click the book icon of the category of topics related to your area of interest. A list of subcategories displays underneath the icon.
2. Continue to click the book icons until you find the category containing the topics in which you are interested. Categories that link to topics display the category title as an underscored link when you move the cursor over the category title. The navigation tree identifies topics with a page icon.
3. Click the topic link. The topic displays in the content frame.

To find a topic or term in the master index:

1. In the navigation tree, click the “Index” category. The category expands to display a list of links arranged in alphabetical order in the navigation tree.
2. In the navigation tree, click the link corresponding to the first character of the term relating to the topic in which you are interested. A list of terms with that initial character displays in the content frame. Terms that have multiple index entries are identified by a book icon.
3. Click the book icon corresponding to the term in which you are interested. A list of subterms and topics displays below the term you clicked. Topics are identified by page icons with an underscored title.
4. Click on the title of the topic that meets your needs. The topic displays in the content frame.

Related concepts:

- "Accessibility" on page 223
- "DB2 Information Center for topics" on page 225

Related tasks:

- "Finding product information by accessing the DB2 Information Center from the administration tools" on page 216
- "Updating the HTML documentation installed on your machine" on page 218
- "Troubleshooting DB2 documentation search with Netscape 4.x" on page 220
- "Searching the DB2 documentation" on page 221

Related reference:

- "Overview of DB2 Universal Database technical information" on page 203

Finding product information by accessing the DB2 Information Center from the administration tools

The DB2 Information Center provides quick access to DB2 product information and is available on all operating systems for which the DB2 administration tools are available.

The DB2 Information Center accessed from the tools provides six types of information.

Tasks Key tasks you can perform using DB2.

Concepts
Key concepts for DB2.

Reference
DB2 reference information, such as keywords, commands, and APIs.

Troubleshooting
Error messages and information to help you with common DB2 problems.

Samples
Links to HTML listings of the sample programs provided with DB2.

Tutorials
Instructional aid designed to help you learn a DB2 feature.

Prerequisites:

Some links in the DB2 Information Center point to Web sites on the Internet. To display the content for these links, you will first have to connect to the Internet.

Procedure:

To find product information by accessing the DB2 Information Center from the tools:

1. Start the DB2 Information Center in one of the following ways:
   - From the graphical administration tools, click on the **Information Center** icon in the toolbar. You can also select it from the **Help** menu.
   - At the command line, enter **db2ic**.
2. Click the tab of the information type related to the information you are attempting to find.
3. Navigate through the tree and click on the topic in which you are interested. The Information Center will then launch a Web browser to display the information.
4. To find information without browsing the lists, click the **Search** icon to the right of the list.

   Once the Information Center has launched a browser to display the information, you can perform a full-text search by clicking the **Search** icon in the navigation toolbar.

Related concepts:

- "Accessibility" on page 223
- "DB2 Information Center for topics" on page 225

Related tasks:

- "Finding topics by accessing the DB2 Information Center from a browser" on page 214
- "Searching the DB2 documentation" on page 221

Viewing technical documentation online directly from the DB2 HTML Documentation CD

All of the HTML topics that you can install from the *DB2 HTML Documentation CD* can also be read directly from the CD. Therefore, you can view the documentation without having to install it.

Restrictions:

Because the following items are installed from the DB2 product CD and not the *DB2 HTML Documentation CD*, you must install the DB2 product to view these items:

- Tools help
- DB2 Quick Tour
- Release notes

Procedure:

1. Insert the *DB2 HTML Documentation* CD. On UNIX operating systems, mount the *DB2 HTML Documentation CD*. Refer to your *Quick Beginnings* book for details on how to mount a CD on UNIX operating systems.
2. Start your HTML browser and open the appropriate file:
   - For Windows operating systems:
     ```
     e:\Program Files\sqllib\doc\htmlcd\%L\index.htm
     ```

     where *e* represents the CD-ROM drive, and %L is the locale of the documentation that you wish to use, for example, **en_US** for English.
   - For UNIX operating systems:
     ```
     /cdrom/Program Files/sqllib/doc/htmlcd/%L/index.htm
     ```

     where */cdrom/* represents where the CD is mounted, and %L is the locale of the documentation that you wish to use, for example, **en_US** for English.

Related tasks:

- "Finding topics by accessing the DB2 Information Center from a browser" on page 214
- "Copying files from the DB2 HTML Documentation CD to a Web Server" on page 220

Related reference:

- "Overview of DB2 Universal Database technical information" on page 203

Updating the HTML documentation installed on your machine

It is now possible to update the HTML installed from the *DB2 HTML Documentation CD* when updates are made available from IBM. This can be done in one of two ways:

- Using the Information Center (if you have the DB2 administration GUI tools installed).
- By downloading and applying a DB2 HTML documentation FixPak .

Note: This will NOT update the DB2 code; it will only update the HTML documentation installed from the *DB2 HTML Documentation CD*.

Procedure:

To use the Information Center to update your local documentation:

1. Start the DB2 Information Center in one of the following ways:
   - From the graphical administration tools, click on the **Information Center** icon in the toolbar. You can also select it from the **Help** menu.
   - At the command line, enter **db2ic**.
2. Ensure your machine has access to the external Internet; the updater will download the latest documentation FixPak from the IBM server if required.
3. Select **Information Center —> Update Local Documentation** from the menu to start the update.
4. Supply your proxy information (if required) to connect to the external Internet.

The Information Center will download and apply the latest documentation FixPak, if one is available.

To manually download and apply the documentation FixPak :

1. Ensure your machine is connected to the Internet.
2. Open the DB2 support page in your Web browser at: www.ibm.com/software/data/db2/udb/winos2unix/support
3. Follow the link for version 8 and look for the ″Documentation FixPaks″ link.
4. Determine if the version of your local documentation is out of date by comparing the documentation FixPak level to the documentation level you have installed. This current documentation on your machine is at the following level: **DB2 v8.1 GA**.
5. If there is a more recent version of the documentation available then download the FixPak applicable to your operating system. There is one FixPak for all Windows platforms, and one FixPak for all UNIX platforms.
6. Apply the FixPak:
   - For Windows operating systems: The documentation FixPak is a self extracting zip file. Place the downloaded documentation FixPak in an empty directory, and run it. It will create a **setup** command which you can run to install the documentation FixPak.
   - For UNIX operating systems: The documentation FixPak is a compressed tar.Z file. Uncompress and untar the file. It will create a directory named `delta_install` with a script called **installdocfix**. Run this script to install the documentation FixPak.

Related tasks:

- “Copying files from the DB2 HTML Documentation CD to a Web Server” on page 220

Related reference:

- “Overview of DB2 Universal Database technical information” on page 203

Copying files from the DB2 HTML Documentation CD to a Web Server

The entire DB2 information library is delivered to you on the *DB2 HTML Documentation CD*, so you can install the library on a Web server for easier access. Simply copy to your Web server the documentation for the languages that you want.

Procedure:

To copy files from the *DB2 HTML Documentation CD* to a Web server, use the appropriate path:

- For Windows operating systems:

  ```
  E:\Program Files\sqllib\doc\htmlcd\%L\*.*
  ```

  where *E* represents the CD-ROM drive and *%L* represents the language identifier.
- For UNIX operating systems:

  ```
  /cdrom:Program Files/sqllib/doc/htmlcd/%L/*.*
  ```

  where *cdrom* represents the CD-ROM drive and *%L* represents the language identifier.

Related tasks:

- “Searching the DB2 documentation” on page 221

Related reference:

- “Supported DB2 interface languages, locales, and code pages” in the *Quick Beginnings for DB2 Servers*
- “Overview of DB2 Universal Database technical information” on page 203

Troubleshooting DB2 documentation search with Netscape 4.x

Most search problems are related to the Java support provided by web browsers. This task describes possible workarounds.

Procedure:

A common problem with Netscape 4.x involves a missing or misplaced security class. Try the following workaround, especially if you see the following line in the browser Java console:

```
Cannot find class  java/security/InvalidParameterException
```

- On Windows operating systems:

  From the *DB2 HTML Documentation CD*, copy the supplied *x*:Program Files\sqllib\doc\htmlcd\locale\InvalidParameterException.class file to the java\classes\java\security\ directory relative to your Netscape browser installation, where *x* represents the CD-ROM drive letter and *locale* represents the name of the desired locale.

  Note: You may have to create the java\security\ subdirectory structure.
- On UNIX operating systems:

  From the *DB2 HTML Documentation CD*, copy the supplied /*cdrom*/Program Files/sqllib/doc/htmlcd/*locale*/InvalidParameterException.class file to the java/classes/java/security/ directory relative to your Netscape browser installation, where *cdrom* represents the mount point of the CD-ROM and *locale* represents the name of the desired locale.

  Note: You may have to create the java/security/ subdirectory structure.

If your Netscape browser still fails to display the search input window, try the following:

- Stop all instances of Netscape browsers to ensure that there is no Netscape code running on the machine. Then open a new instance of the Netscape browser and try to start the search again.
- Purge the browser's cache.
- Try a different version of Netscape, or a different browser.

Related tasks:

- "Searching the DB2 documentation" on page 221

Searching the DB2 documentation

To search DB2's documentation, you need Netscape 6.1 or higher, or Microsoft's Internet Explorer 5 or higher. Ensure that your browser's Java support is enabled.

A pop-up search window opens when you click the search icon in the navigation toolbar of the Information Center accessed from a browser. If you are using the search for the first time it may take a minute or so to load into the search window.

Restrictions:

The following restrictions apply when you use the documentation search:

- Boolean searches are not supported. The boolean search qualifiers *and* and *or* will be ignored in a search. For example, the following searches would produce the same results:
  - servlets *and* beans
  - servlets *or* beans
- Wildcard searches are not supported. A search on *java** will only look for the literal string *java** and would not, for example, find *javadoc*.

In general, you will get better search results if you search for phrases instead of single words.

Procedure:

To search the DB2 documentation:

1. In the navigation toolbar, click **Search**.
2. In the top text entry field of the Search window, enter two or more terms related to your area of interest and click **Search**. A list of topics ranked by accuracy displays in the **Results** field.

   Entering more terms increases the precision of your query while reducing the number of topics returned from your query.
3. In the **Results** field, click the title of the topic you want to read. The topic displays in the content frame.

Note: When you perform a search, the first result is automatically loaded into your browser frame. To view the contents of other search results, click on the result in results lists.

Related tasks:

- "Troubleshooting DB2 documentation search with Netscape 4.x" on page 220

Online DB2 troubleshooting information

With the release of DB2® UDB Version 8, there will no longer be a *Troubleshooting Guide*. The troubleshooting information once contained in this guide has been integrated into the DB2 publications. By doing this, we are able to deliver the most up-to-date information possible. To find information on the troubleshooting utilities and functions of DB2, access the DB2 Information Center from any of the tools.

Refer to the DB2 Online Support site if you are experiencing problems and want help finding possible causes and solutions. The support site contains a large, constantly updated database of DB2 publications, TechNotes, APAR (product problem) records, FixPaks, and other resources. You can use the support site to search through this knowledge base and find possible solutions to your problems.

Access the Online Support site at www.ibm.com/software/data/db2/udb/winos2unix/support, or by clicking the **Online Support** button in the DB2 Information Center. Frequently changing information, such as the listing of internal DB2 error codes, is now also available from this site.

Related concepts:

- "DB2 Information Center for topics" on page 225

Related tasks:

- "Finding product information by accessing the DB2 Information Center from the administration tools" on page 216

Accessibility

Accessibility features help users with physical disabilities, such as restricted mobility or limited vision, to use software products successfully. These are the major accessibility features in DB2® Universal Database Version 8:

- DB2 allows you to operate all features using the keyboard instead of the mouse. See "Keyboard Input and Navigation".
- DB2 enables you customize the size and color of your fonts. See "Accessible Display" on page 224.
- DB2 allows you to receive either visual or audio alert cues. See "Alternative Alert Cues" on page 224.
- DB2 supports accessibility applications that use the Java™ Accessibility API. See "Compatibility with Assistive Technologies" on page 224.
- DB2 comes with documentation that is provided in an accessible format. See "Accessible Documentation" on page 224.

Keyboard Input and Navigation

Keyboard Input

You can operate the DB2 Tools using only the keyboard. You can use keys or key combinations to perform most operations that can also be done using a mouse.

Keyboard Focus

In UNIX-based systems, the position of the keyboard focus is highlighted, indicating which area of the window is active and where your keystrokes will have an effect.

Accessible Display

The DB2 Tools have features that enhance the user interface and improve accessibility for users with low vision. These accessibility enhancements include support for customizable font properties.

Font Settings

The DB2 Tools allow you to select the color, size, and font for the text in menus and dialog windows, using the Tools Settings notebook.

Non-dependence on Color

You do not need to distinguish between colors in order to use any of the functions in this product.

Alternative Alert Cues

You can specify whether you want to receive alerts through audio or visual cues, using the Tools Settings notebook.

Compatibility with Assistive Technologies

The DB2 Tools interface supports the Java Accessibility API enabling use by screen readers and other assistive technologies used by people with disabilities.

Accessible Documentation

Documentation for the DB2 family of products is available in HTML format. This allows you to view documentation according to the display preferences set in your browser. It also allows you to use screen readers and other assistive technologies.

DB2 tutorials

The DB2® tutorials help you learn about various aspects of DB2 Universal Database. The tutorials provide lessons with step-by-step instructions in the areas of developing applications, tuning SQL query performance, working with data warehouses, managing metadata, and developing Web services using DB2.

Before you begin:

Before you can access these tutorials using the links below, you must install the tutorials from the *DB2 HTML Documentation* CD-ROM.

If you do not want to install the tutorials, you can view the HTML versions of the tutorials directly from the *DB2 HTML Documentation CD*. PDF versions of these tutorials are also available on the *DB2 PDF Documentation CD*.

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

DB2 Universal Database tutorials:

If you installed the tutorials from the *DB2 HTML Documentation* CD-ROM, you can click on a tutorial title in the following list to view that tutorial.

Business Intelligence Tutorial: Introduction to the Data Warehouse Center
Perform introductory data warehousing tasks using the Data Warehouse Center.

Business Intelligence Tutorial: Extended Lessons in Data Warehousing
Perform advanced data warehousing tasks using the Data Warehouse Center.

Development Center Tutorial for Video Online using Microsoft® Visual Basic
Build various components of an application using the Development Center Add-in for Microsoft Visual Basic.

Information Catalog Center Tutorial
Create and manage an information catalog to locate and use metadata using the Information Catalog Center.

Video Central for e-business Tutorial
Develop and deploy an advanced DB2 Web Services application using WebSphere® products.

Visual Explain Tutorial
Analyze, optimize, and tune SQL statements for better performance using Visual Explain.

DB2 Information Center for topics

The DB2® Information Center gives you access to all of the information you need to take full advantage of DB2 Universal Database™ and DB2 Connect™ in your business. The DB2 Information Center also documents major DB2 features and components including replication, data warehousing, the Information Catalog Center, Life Sciences Data Connect, and DB2 extenders.

The DB2 Information Center accessed from a browser has the following features:

Regularly updated documentation
Keep your topics up-to-date by downloading updated HTML.

Search

Search all of the topics installed on your workstation by clicking **Search** in the navigation toolbar.

Integrated navigation tree

Locate any topic in the DB2 library from a single navigation tree. The navigation tree is organized by information type as follows:

- Tasks provide step-by-step instructions on how to complete a goal.
- Concepts provide an overview of a subject.
- Reference topics provide detailed information about a subject, including statement and command syntax, message help, requirements.

Master index

Access the information in topics and tools help from one master index. The index is organized in alphabetical order by index term.

Master glossary

The master glossary defines terms used in the DB2 Information Center. The glossary is organized in alphabetical order by glossary term.

Related tasks:

- "Finding topics by accessing the DB2 Information Center from a browser" on page 214
- "Finding product information by accessing the DB2 Information Center from the administration tools" on page 216
- "Updating the HTML documentation installed on your machine" on page 218

Appendix C. Notices

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

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

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

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

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

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

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

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product, and use of those Web sites is at your own risk.

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

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

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

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

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

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

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

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

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

COPYRIGHT LICENSE:

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

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

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

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both, and have been used in at least one of the documents in the DB2 UDB documentation library.

ACF/VTAM
AISPO
AIX
AIXwindows
AnyNet
APPN
AS/400
BookManager
C Set++
C/370
CICS
Database 2
DataHub
DataJoiner
DataPropagator
DataRefresher
DB2
DB2 Connect
DB2 Extenders
DB2 OLAP Server
DB2 Universal Database
Distributed Relational Database Architecture
DRDA
eServer
Extended Services
FFST
First Failure Support Technology
IBM
IMS
IMS/ESA
iSeries
LAN Distance
MVS
MVS/ESA
MVS/XA
Net.Data
NetView
OS/390
OS/400
PowerPC
pSeries
QBIC
QMF
RACF
RISC System/6000
RS/6000
S/370
SP
SQL/400
SQL/DS
System/370
System/390
SystemView
Tivoli
VisualAge
VM/ESA
VSE/ESA
VTAM
WebExplorer
WebSphere
WIN-OS/2
z/OS
zSeries

The following terms are trademarks or registered trademarks of other companies and have been used in at least one of the documents in the DB2 UDB documentation library:

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

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

Index

Contacting IBM

In the United States, call one of the following numbers to contact IBM:

- 1-800-237-5511 for customer service
- 1-888-426-4343 to learn about available service options
- 1-800-IBM-4YOU (426-4968) for DB2 marketing and sales

In Canada, call one of the following numbers to contact IBM:

- 1-800-IBM-SERV (1-800-426-7378) for customer service
- 1-800-465-9600 to learn about available service options
- 1-800-IBM-4YOU (1-800-426-4968) for DB2 marketing and sales

To locate an IBM office in your country or region, check IBM's Directory of Worldwide Contacts on the web at www.ibm.com/planetwide

Product information

Information regarding DB2 Universal Database products is available by telephone or by the World Wide Web at www.ibm.com/software/data/db2/udb

This site contains the latest information on the technical library, ordering books, client downloads, newsgroups, FixPaks, news, and links to web resources.

If you live in the U.S.A., then you can call one of the following numbers:

- 1-800-IBM-CALL (1-800-426-2255) to order products or to obtain general information.
- 1-800-879-2755 to order publications.

For information on how to contact IBM outside of the United States, go to the IBM Worldwide page at www.ibm.com/planetwide

Part Number: CT17XNA

Printed in the U.S.A.

SC09-4835-00

(1P) P/N: CT17XNA

Spine information:

IBM® DB2® Connect User's Guide Version 8