

IBM DB2 10.1
for Linux, UNIX, and Windows

*Developing ADO.NET and OLE DB
Applications*
Updated January, 2013



IBM DB2 10.1
for Linux, UNIX, and Windows

*Developing ADO.NET and OLE DB
Applications*
Updated January, 2013



Note

Before using this information and the product it supports, read the general information under Appendix B, "Notices," on page 87.

Edition Notice

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

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

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

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

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

© Copyright IBM Corporation 2006, 2013.

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

Contents

Chapter 1. ADO.NET application development. 1

Deploying .NET applications (Windows)	2
Supported .NET development software	2
DB2 integration in Visual Studio.	4
IBM Data Server Provider for .NET.	4
IBM Data Server Provider for .NET database system requirements.	5
Programming applications to use the IBM Data Server Provider for .NET	6
Building .NET Applications	22
OLE DB .NET Data Provider	26
OLE DB .NET Data Provider restrictions.	27
Hints and tips	31
ODBC .NET Data Provider	32
ODBC .NET Data Provider restrictions	33

Chapter 2. IBM OLE DB Provider for DB2 41

Application Types Supported by the IBM OLE DB Provider for DB2	41
OLE DB services.	42
Thread model supported by the IBM OLE DB Provider	42
Large object manipulation with the IBM OLE DB Provider	42
Schema rowsets supported by the IBM OLE DB Provider	42
OLE DB services automatically enabled by the IBM OLE DB Provider.	45
Data services	45
Supported cursor modes for the IBM OLE DB Provider	45
Data type mappings between DB2 and OLE DB	45
Data conversion for setting data from OLE DB Types to DB2 Types	46
Data conversion for setting data from DB2 types to OLE DB types	49
IBM OLE DB Provider restrictions.	52
IBM OLE DB Provider support for OLE DB components and interfaces	53
IBM OLE DB Provider support for OLE DB properties	55

Connections to data sources using the IBM OLE DB Provider	60
ADO applications	60
ADO connection string keywords	60
Connections to data sources with Visual Basic ADO applications	61
Updatable scrollable cursors in ADO applications	61
Limitations for ADO applications	61
IBM OLE DB Provider support for ADO methods and properties	62
Compilation and linking of C/C++ applications and the IBM OLE DB Provider	68
Connections to data sources in C/C++ applications using the IBM OLE DB Provider	68
COM+ distributed transaction support and the IBM OLE DB Provider	68
Enablement of COM+ support in C/C++ database applications	68

Chapter 3. IBM Data Server Provider for .NET. 71

IBM.Data.DB2 Namespace	72
IBM.Data.DB2Types Namespace	75

Appendix A. Overview of the DB2 technical information 77

DB2 technical library in hardcopy or PDF format.	77
Displaying SQL state help from the command line processor	80
Accessing different versions of the DB2 Information Center	80
Updating the DB2 Information Center installed on your computer or intranet server	80
Manually updating the DB2 Information Center installed on your computer or intranet server	82
DB2 tutorials	84
DB2 troubleshooting information	84
Terms and conditions	84

Appendix B. Notices 87

Index 91

Chapter 1. ADO.NET application development

The following highlight the key .NET Framework features and how .NET Framework replaced COM technology.

In recent years, Microsoft has been promoting a new software development platform for Windows, known as the .NET Framework. The .NET Framework is Microsoft's replacement for Component Object Model (COM) technology. The following points highlight the key .NET Framework features:

- You can code .NET applications in over forty different programming languages. The most popular languages for .NET development are C# and Visual Basic .NET.
- The .NET Framework class library provides the building blocks with which you build .NET applications. This class library is language agnostic and provides interfaces to operating system and application services.
- Your .NET application (regardless of language) compiles into Intermediate Language (IL), a type of bytecode.
- The Common Language Runtime (CLR) is the heart of the .NET Framework, compiling the IL code on the fly, and then running it. In running the compiled IL code, the CLR activates objects, verifies their security clearance, allocates their memory, executes them, and cleans up their memory once execution is finished.

Through these features, the .NET Framework facilitates a wide variety of application implementations (for example, Windows forms, web forms, and web services), rapid application development, and secure application deployment. COM and COM+ proved to be inadequate or cumbersome for all the aforementioned features.

The .NET Framework provides extensive data access support through ADO.NET. ADO.NET supports both connected and disconnected access. The key component of disconnected data access in ADO.NET is the DataSet class, instances of which act as a database cache that resides in your application's memory.

For both connected and disconnected access, your applications use databases through what's known as a data provider. Various database products include their own .NET data providers for, including DB2[®] for Windows.

A .NET data provider features implementations of the following basic classes:

- Connection: Establishes and manages a database connection.
- Command: Executes an SQL statement against a database.
- DataReader: Reads and returns result set data from a database.
- DataAdapter: Links a DataSet instance to a database. Through a DataAdapter instance, the DataSet can read and write database table data.

Microsoft provides two data providers, the OLE DB .NET Data Provider and ODBC .NET Data Provider. The OLE DB .NET Data Provider is a bridge provider that feeds ADO.NET requests to the IBM[®] OLE DB Provider (by way of the COM interop module). ODBC .NET Data Provider is a bridge provider that feeds ADO.NET requests to the IBM ODBC Driver. These .NET data provider are not recommended for access to DB2 family databases. The IBM Data Server Provider for .NET is a high performance, managed ADO.NET data provider. This is the recommended .NET data provider for use with DB2 family databases. ADO.NET

database access using the IBM Data Server Provider for .NET has fewer restrictions, and provides significantly better performance than the OLE DB and ODBC .NET bridge providers.

Deploying .NET applications (Windows)

To simplify .NET application deployment, IBM provides the IBM Data Server Driver Package, a small-footprint client that is ideal for use in mass deployment scenarios.

You can use the IBM Data Server Runtime Client instead, if the additional features of that client are required over the IBM Data Server Driver Package.

Before you begin

- Before deployment, you must build your .NET application, which you can do with either Visual Studio or the command line. For more information about building .NET applications, see the related tasks.
- Computers that you use to build .NET applications and computers where you will deploy .NET applications must have a supported version of the Windows operating system, in addition to other software, as described in “Supported .NET development software”:
 - Build systems
 - Windows operating system
 - Visual Studio
 - .NET Framework Redistributable Package
 - .NET Framework Software Development Kit
 - Deployment systems
 - Windows operating system
 - .NET Framework Redistributable Package

Procedure

To deploy a .NET application:

1. Install the IBM Data Server Driver Package onto the computers where you will deploy your application. During the installation, set the IBM Data Server Driver Package installation to be the default database client interface copy.

Note: Any existing database applications that run against an IBM data server will use this new installation of the IBM Data Server Driver Package. Test those applications against the new driver before rolling out your deployed .NET application.

2. Install your built application onto the computers where your application will run.

Supported .NET development software

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

Supported operating systems for developing and deploying .NET Framework 2.0, 3.0, 3.5, and 4.0 applications

The following operating systems are supported for developing and deploying .NET Framework 2.0, 3.0, 3.5, and 4.0 applications:

- Windows XP, Service Pack 2 (32-bit and 64-bit editions)
- Windows Server 2003 (32-bit and 64-bit editions)
- Windows Vista (32-bit and 64-bit editions)
- Windows Server 2008 (32-bit and 64-bit editions)
- Windows Server 2008 R2 (64-bit edition)
- Windows 7 (32-bit and 64-bit editions)

Supported operating systems for developing and deploying .NET Framework 4.5 applications

The following operating systems are supported for developing and deploying .NET Framework 4.5 applications:

- Windows Vista Service Pack 2 (32-bit and 64-bit editions)
- Windows Server 2008 Service Pack 2 (32-bit and 64-bit editions)
- Windows Server 2008 R2 (64-bit edition)
- Windows 7 (32-bit and 64-bit editions)

Microsoft .NET Framework 4.5 is supported with Version 10.1 Fix Pack 2 and later fix packs.

Supported development software for .NET Framework applications

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

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

Microsoft Visual Studio 2012 is supported with Version 10.1 Fix Pack 2 and later fix packs.

Supported deployment software for .NET Framework applications

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

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

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

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

DB2 integration in Visual Studio

The IBM Database Add-Ins for Visual Studio are a collection of features that integrate seamlessly into your Visual Studio development environment so that you can work with DB2 servers and develop DB2 procedures, functions, and objects.

IBM Database Add-Ins for Visual Studio present a simple interface to DB2 databases. For example, instead of using SQL, the creation of database objects can be done using designers and wizards. And for situations where you do need to write SQL code, the integrated DB2 SQL editor has the following features:

- Colored SQL text for increased readability
- Integration with the Microsoft Visual Studio IntelliSense feature, which provides for intelligent auto-completion while you are typing DB2 scripts

With IBM Database Add-Ins for Visual Studio, you can:

- Open various DB2 development and administration tools.
- Create and manage DB2 projects in the Solution Explorer.
- Access and manage DB2 data connections from the Server Explorer.
- Create and modify DB2 scripts, including scripts to create stored procedures, functions, tables, views, indexes, and triggers.

Visual Studio 2008 and 2010

The IBM Database Add-Ins for Visual Studio are included as a separately installable component with the DB2 Client and the DB2 servers. Once you are finished installing your DB2 product, you will be presented with an option to install the IBM Database Add-Ins for Visual Studio. If you do not have Visual Studio installed on your computer, the add-ins will not install. Once you install Visual Studio, you can then install the add-ins at any time from the DB2 product's setup menu.

For more details about using the IBM Database Add-Ins and the Data Server Provider for .NET for rapid application development, visit the IBM Information Management and Visual Studio .NET zone at <http://www.ibm.com/developerworks/data/zones/vstudio/index.html>.

IBM Data Server Provider for .NET

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

Two providers are included in the IBM Data Server Provider for .NET client package. These providers are sometimes referred to as the Common .NET Providers.

The DB2 .NET Provider (IBM.Data.DB2.dll)

With the DB2 .NET Provider your .NET applications can access the following database management systems:

- DB2 for Linux, UNIX, and Windows, Version 9.1, Version 9.5, Version 9.7, Version 9.8, and Version 10.1
- DB2 Universal Database™ Version 8 for Windows, UNIX, and Linux

- DB2 for z/OS® Version 8, Version 9, and Version 10 , through DB2 Connect™
- IBM DB2 for IBM i Version 5 Release 4, Version 6 Release 1 and Version 7 Release 1, through DB2 Connect (for IBM DB2 Version 9.7 Fix Pack 4 and higher versions)
- IBM DB2 for IBM i Version 5 Release 4 and Version 6 Release 1, through DB2 Connect (for IBM DB2 Version 9.7 Fix Pack 3 and earlier versions)
- IBM Informix® Version 11.10, and later

The remainder of these topics discuss the Common DB2 .NET Provider.

The Informix database server .NET Provider (IBM.Data.Informix.dll)

With the Informix database server .NET Provider your .NET applications can access the following database management systems:

- IBM Informix, Version 11.10, and later

For more information about this provider, see the: IBM Informix Dynamic Server Information Center.

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

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

IBM Data Server Provider for .NET database system requirements

An application can use IBM Data Server Provider for .NET to access IBM database systems.

The DB2 .NET Provider (IBM.Data.DB2.dll)

With the DB2 .NET Provider your .NET applications can access the following database management systems:

- DB2 for Linux, UNIX, and Windows, Version 9.1, Version 9.5, Version 9.7, Version 9.8, and Version 10.1
- DB2 Universal Database Version 8 for Windows, UNIX, and Linux
- DB2 for z/OS Version 8, Version 9, and Version 10 , through DB2 Connect
- IBM DB2 for IBM i Version 5 Release 4, Version 6 Release 1 and Version 7 Release 1, through DB2 Connect (for IBM DB2 Version 9.7 Fix Pack 4 and higher versions)
- IBM DB2 for IBM i Version 5 Release 4 and Version 6 Release 1, through DB2 Connect (for IBM DB2 Version 9.7 Fix Pack 3 and earlier versions)
- IBM Informix Version 11.10, and later

The remainder of these topics discuss the Common DB2 .NET Provider.

The Informix database server .NET Provider (IBM.Data.Informix.dll)

With the Informix database server .NET Provider your .NET applications can access the following database management systems:

- IBM Informix, Version 11.10, and later

For more information about this provider, see the: IBM Informix Dynamic Server Information Center.

For DB2 for i, you require the following fix on the server: APAR II13348.

Programming applications to use the IBM Data Server Provider for .NET

Programming applications to use the IBM Data Server Provider for .NET requires understanding of available features in the IBM Data Provider for .NET. After you determine the requirement of your application, you can leverage available features in the IBM Data Provider for .NET.

Generic coding with the ADO.NET common base classes

The .NET Framework, versions 2.0, 3.0, and 3.5, features a namespace called `System.Data.Common`, which features a set of base classes that can be shared by any .NET data provider. This facilitates a generic ADO.NET database application development approach, featuring a constant programming interface across different databases.

The following C# demonstrates a generic approach to establishing a database connection.

```
DbProviderFactory factory = DbProviderFactories.GetFactory("IBM.Data.DB2");
DbConnection conn = factory.CreateConnection();
DbConnectionStringBuilder sb = factory.CreateConnectionStringBuilder();

if( sb.ContainsKey( "Database" ) )
{
    sb.Remove( "database" );
    sb.Add( "database", "SAMPLE" );
}

conn.ConnectionString = sb.ConnectionString;

conn.Open();
```

The `DbProviderFactory` object is the point where any generic ADO.NET application begins. This object creates generic instances of .NET data provider objects, such as connections, data adapters, commands, and data readers, which work with a specific database product. In the previous example, the "IBM.Data.DB2" string passed into the `GetFactory` method uniquely identifies the IBM Data Server Provider for .NET, and results in the initialization of a `DbProviderFactory` instance that creates database provider object instances specific to the IBM Data Server Provider for .NET.

The `DbConnection` object can connect to DB2 family Informix databases, just as a `DB2Connection` object, which is actually inherited from `DbConnection`. Using the `DbConnectionStringBuilder` class, you can determine the connection string keywords for a data provider, and generate a custom connection string. The code in the previous example checks if a keyword named "database" exists in the IBM Data Server Provider for .NET, and if so, generates a connection string to connect to the SAMPLE database.

Connecting to a database from an application using the IBM Data Server Provider for .NET

When using the IBM Data Server Provider for .NET, a database connection is established through the `DB2Connection` class.

Procedure

To connect to a database:

1. Create a string that stores the connection parameters. The format for a typical connection string format is:

```
Server=<ip address/localhost>:<port number>;  
Database=<db name>;  
UID=<userID>;  
PWD=<password>;  
Connect Timeout=<Timeout value>
```

Examples of possible connection strings are:

Example 1:

```
String connectionString = "Database=SAMPLE";  
// When used, attempts to connect to the SAMPLE database.
```

Note: If you specify only the database name in the connection string, the other information such as the server, userid, and password, must be included in the `db2dsdriver.cfg` file.

Example 2:

```
String cs = "Server=srv:50000;  
Database=SAMPLE;  
UID=db2adm;  
PWD=ab1d;Connect Timeout=30";  
// When used, attempts to connect to the SAMPLE database on the server  
// 'srv' through port 50000 using 'db2adm' and 'ab1d' as the user id and  
// password respectively. If the connection attempt takes  
// more than thirty seconds, the attempt will be terminated and an error  
// will be generated.
```

2. Pass the `connectString` to the `DB2Connection` constructor.

- Connecting to a database in C#:

```
String connectionString = "Database=SAMPLE";  
DB2Connection conn = new DB2Connection(connectionString);  
conn.Open();  
return conn;
```

- Connecting to a database in Visual Basic .NET:

```
Dim connectionString As String = "Database=SAMPLE"  
Dim conn As DB2Connection = new DB2Connection(connectionString)  
conn.Open()  
Return conn
```

3. Use the `DB2Connection` object's `Open` method to formally connect to the database identified in `connectString`.

Connection pooling with the IBM Data Server Provider for .NET

When a connection is first opened against a DB2 database, a connection pool is created. As connections are closed, they enter the pool, ready to be reused within the same process by other applications that need connections.

The IBM Data Server Provider for .NET uses a normalized set of connection string attributes for determining the connection pool. By using normalized attributes, the chances of an application reusing connections is increased.

The IBM Data Server Provider for .NET enables connection pooling by default.

Note: You can turn connection pooling off using the `Pooling=false` connection string keyword/value pair. However, if you turn off connection pooling COM+ applications will not work.

You can control the behavior of the connection pool by setting following connection string keywords:

- The minimum and maximum pool size (**MinPoolSize** and **MaxPoolSize**)

- The length of time a connection can be idle before it is returned to the pool (**ConnectionLifetimeInPool**)

Creating a trusted connection through IBM Data Server Provider for .NET

.NET applications support trusted context using connection string keywords.

The following keywords are available in the connection string:

- **TrustedContextSystemUserID**, or **tcsuid**, which specifies the trusted context SYSTEM AUTHID to be used with the connection.
- **TrustedContextSystemPassword**, or **tccspwd**, which specifies the password corresponding to the trusted context SYSTEM AUTHID to be used with the connection.

If the **TrustedContextSystemPassword** keyword is specified without a **TrustedContextSystemUserID** keyword value, an **InvalidArgument** exception is thrown. The **UserID** keyword is also required in a trusted context scenario.

The trusted context through IBM Data Server Provider for .NET is currently supported by:

- DB2 Database for Linux, UNIX, and Windows, Version 9.5, Version 9.7, Version 9.8, and Version 10.1
- DB2 Universal Database Version 9 and Version 10 for z/OS

Example

Suppose a trusted context has been established on the server with the following information:

```
CREATE TRUSTED CONTEXT ctxName1
BASED UPON CONNECTION USING SYSTEM AUTHID masteruser
ATTRIBUTES ( PROTOCOL 'TCPIP',
              ADDRESS '9.26.146.201',
              ENCRYPTION 'NONE' )
ENABLE
WITH USE FOR userapp1 WITH AUTHENTICATION, userapp2 WITH AUTHENTICATION;
```

The SYSTEM AUTHID, **masteruser**, has a corresponding password, **masterpassword**. Each specific user/application, **userapp1**, and **userapp2**, has a corresponding password, **passapp1** and **passapp2**.

In order to use this trusted context, applications would issue connection strings as follows:

- Application 1


```
database=db;server=server1:446;
UserID=userapp1;Password=passapp1;
TrustedContextSystemUserID=masteruser;TrustedContextSystemPassword=masterpassword
```
- Application 2


```
database=db;server=server1:446;
UserID=userapp2;Password=passapp2;
TrustedContextSystemUserID=masteruser;TrustedContextSystemPassword=masterpassword
```

Note: The **UserID** keyword corresponds to the end user of the connection in a trusted context situation, just as in standard applications.

Following .NET program open and close a connection:

```

[C#]
DB2Connection conn = new DB2Connection();

conn.ConnectionString = "database=db;server=server1:446;"
    + "UserID=userapp1;Password=passapp1;"
    + "TrustedContextSystemUserID=masteruser;"
    + "TrustedContextSystemPassword=masterpassword;"

conn.Open();

// Do processing as userapp1, such as querying tables

conn.Close();

conn.ConnectionString = "database=db;server=server1:446;UserID=userapp2;"
    + "Password=passapp2;TrustedContextSystemUserID=masteruser;"
    + "TrustedContextSystemPassword=masterpassword;"

conn.Open();

// Do processing as userapp2

conn.Close();

```

If the trusted context processing fails because no trusted context was set up on the server, or the server does not support trusted contexts, an error with SQLCODE CLI0197E will be thrown. If the TrustedContextSystemUserID keyword value is invalid (too long, for example), an error with SQLCODE CLI0124E will be thrown. The server might report an error with SQLCODE SQL1046N, SQL30082N, or SQL0969N with a native error code of -20361. Any of these errors will cause Open() to fail.

Note: The trusted context processing happens on the next communication with the server.

SQL data type representation in ADO.NET database applications

ADO.NET database applications can reference DB2 SQL data type values as parameter values to be used as part of SQL statement execution and as variables, however the appropriate IBM Data Server Provider for .NET data type values and .NET Framework data type values must be used to ensure that there is no truncation or loss of data when accessing or retrieving the values.

For specifying parameter values to be used as part of a SQL statement to be executed, IBM Data Server Provider for .NET objects must be used. The DB2Parameter object is used to represent a parameter to be added to a DB2Command object which represents a SQL statement. When specifying the data type value for the parameter, the IBM Data Server Provider for .NET data type values available in the IBM.Data.DB2Types namespace must be used. The IBM.Data.DB2Types namespace provides classes and structures to represent each of the supported DB2 SQL data types.

For local variables that might temporarily hold SQL data type values, appropriate IBM Data Server Provider for .NET data types, as defined in the IBM.Data.DB2Types Namespace, must be used.

The following table shows mappings between DB2Type data types, DB2 data types, Informix data types, Microsoft .NET Framework types, and DB2Types classes and structures.

Category	DB2Types Classes and Structures	DB2Type Data Type	DB2 Data Type	Informix Data Type	.NET Data Type
Binary data	DB2Binary	Binary	CHAR FOR BIT DATA		Byte[]
	DB2Binary	Binary ³	BINARY		Byte[]
	DB2Binary	VarBinary ³	VARBINARY		Byte[]
	DB2Binary	LongVarBinary ¹	LONG VARCHAR FOR BIT DATA		Byte[]
Character data	DB2String	Char	CHAR	CHAR	String
	DB2String	VarChar	VARCHAR	VARCHAR	String
	DB2String	LongVarChar ¹	LONG VARCHAR	LVARCHAR	String
Graphic data	DB2String	Graphic	GRAPHIC		String
	DB2String	VarGraphic	VARGRAPHIC		String
	DB2String	LongVarGraphic ¹	LONG VARGRAPHIC		String
LOB data	DB2Clob	Clob	CLOB	CLOB, TEXT	String
	DB2Blob	Blob	BLOB	BLOB, BYTE	Byte[]
	DB2Clob	DbClob	DBCLOB		String
Numeric data	DB2Int16	SmallInt	SMALLINT	BOOLEAN, SMALLINT	Int16
	DB2Int32	Integer	INT	INTEGER, INT, SERIAL	Int32
	DB2Int64	BigInt, BigSerial	BIGINT	BIGINT, BIGSERIAL, INT8, SERIAL8	Int64
	DB2Real, DB2Real370	Real	REAL	REAL, SMALLFLOAT	Single
	DB2Double	Double	DOUBLE PRECISION	DECIMAL (≤ 29), DOUBLE PRECISION	Double
	DB2Double	Float	FLOAT	DECIMAL (32), FLOAT	Double
	DB2Decimal	Decimal	DECIMAL	MONEY	Decimal
	DB2DecimalFloat	DecimalFloat	DECFLOAT(16 34) ¹⁴		Decimal
	DB2Decimal	Numeric	DECIMAL	DECIMAL (≤ 29), NUMERIC	Decimal
Date/Time data	DB2Date	Date	DATE	DATETIME (date precision)	DateTime String ⁵
	DB2Time	Time	TIME	DATETIME (time precision)	TimeSpan String ⁵
	DB2TimeStamp	Timestamp	TIMESTAMP	DATETIME (time and date precision)	DateTime String ⁵
	DB2TimeStamp Offset	TimestampWith TimeZone	TIMESTAMP WITH TIME ZONE	N/A	DateTimeOffset String ⁵
Row ID data	DB2RowId	RowId	ROWID		Byte[]
XML data	DB2Xml	Xml ²	XML		Byte[]

1. These data types are not supported as parameters in DB2 .NET common language runtime routines.
2. A DB2ParameterClass.ParameterName property of the type DB2Type.Xml can accept variables of the following types: String, byte[], DB2Xml, and XmlReader.
3. These data types are applicable only to DB2 for z/OS and DB2 for i V6R1 and later.
4. This data type is only supported for DB2 for z/OS Version 9 and later releases and for DB2 for Linux, UNIX, and Windows Version 9.5 and later releases.
5. Date and Time objects can be timestamp string literals. Timestamp objects can be date string literals

Executing SQL statements from an application using the IBM Data Server Provider for .NET

When using the IBM Data Server Provider for .NET, the execution of SQL statements is done through a `DB2Command` class using its methods `ExecuteReader()` and `ExecuteNonQuery()`, and its properties `CommandText`, `CommandType` and `Transaction`.

About this task

For SQL statements that produce output, you can use the `ExecuteReader()` method, and retrieve the results from a `DB2DataReader` object. For all other SQL statements, you can use the `ExecuteNonQuery()` method. You should initialize the `Transaction` property of the `DB2Command` object to a `DB2Transaction` object. A `DB2Transaction` object is responsible for rolling back and committing database transactions.

Executing an UPDATE statement in C#:

```
// assume a DB2Connection conn
DB2Command cmd = conn.CreateCommand();
DB2Transaction trans = conn.BeginTransaction();
cmd.Transaction = trans;
cmd.CommandText = "UPDATE staff " +
    " SET salary = (SELECT MIN(salary) " +
    " FROM staff " +
    " WHERE id >= 310) " +
    " WHERE id = 310";

cmd.ExecuteNonQuery();
```

Executing an UPDATE statement in Visual Basic .NET:

```
' assume a DB2Connection conn
DB2Command cmd = conn.CreateCommand();
DB2Transaction trans = conn.BeginTransaction();
cmd.Transaction = trans;
cmd.CommandText = "UPDATE staff " +
    " SET salary = (SELECT MIN(salary) " +
    " FROM staff " +
    " WHERE id >= 310) " +
    " WHERE id = 310";
cmd.ExecuteNonQuery();
```

Executing a SELECT statement in C#:

```
// assume a DB2Connection conn
DB2Command cmd = conn.CreateCommand();
DB2Transaction trans = conn.BeginTransaction();
cmd.Transaction = trans;
cmd.CommandText = "SELECT deptnumb, location " +
    " FROM org " +
    " WHERE deptnumb < 25";

DB2DataReader reader = cmd.ExecuteReader();
```

Executing a SELECT statement in Visual Basic .NET:

```
' assume a DB2Connection conn
Dim cmd As DB2Command = conn.CreateCommand()
Dim trans As DB2Transaction = conn.BeginTransaction()
cmd.Transaction = trans
cmd.CommandText = "UPDATE staff " +
    " SET salary = (SELECT MIN(salary) " +
    " FROM staff " +
```

```

        "           WHERE id >= 310) " +
        " WHERE id = 310";

cmd.ExecuteNonQuery()

```

After your application performs a database transaction, you must either roll it back or commit it. This is done through the `Commit()` and `Rollback()` methods of a `DB2Transaction` object.

Rolling back or committing a transaction in C#:

```

// assume a DB2Transaction object conn
trans.Rollback();
...
trans.Commit();

```

Rolling back or committing a transaction in Visual Basic.NET:

```

' assume a DB2Transaction object conn
trans.Rollback()
...
trans.Commit()

```

In Version 9.7 Fix Pack 2 and later fix packs, DB2 for Linux, UNIX, and Windows servers support returning result sets from the execution of anonymous blocks. In Version 9.7 Fix Pack 6 and later fix packs, the .NET data provider supports an application to retrieve the result sets from execution of anonymous blocks by using `DB2DataReader` or `DB2ResultSet` classes. For the .NET data provider to retrieve the result sets from anonymous block execution, the database server must support PL/SQL statements and the database must be enabled to process PL/SQL statements. The .NET data provider must declare cursors for the results sets that are returned from anonymous block execution by using the `BEGIN` statement, and not the `BEGIN COMPOUND` statement.

Retrieving a single result set from execution of anonymous block by using the `DB2DataReader` class in C#:

```

...
cmd.CommandText = "begin " +
                  "declare cursor1 cursor with return to client with hold for select c1 from t1; " +
                  "open cursor1; " +
                  "end;";
//Returns a result set by opened cursor cursor1
DB2DataReader dr = cmd.ExecuteReader();
while (dr.Read())
{
    //Process read data if required
}
dr.Close();
...

```

Retrieving a single result set from execution of anonymous block by using the `DB2ResultSet` class in C#:

```

...
cmd.CommandText = "begin " +
                  "declare cursor1 cursor with return to client with hold for select c1 from t1; " +
                  "open cursor1; " +
                  "end;";
//Returns a result set by opened cursor cursor1
DB2ResultSet ds = cmd.ExecuteResultSet(DB2CursorType.ForwardOnly);
while (ds.Read())
{
    //Process read data if required
}
ds.Close();
...

```

Retrieving multiple result sets from execution of anonymous block by using the DB2DataReader class in C#:

```
...
cmd.CommandText = " begin " +
                  "declare cursor1 cursor with return to client with hold for select c1 from t1; " +
                  "declare cursor2 cursor with return to client for select c2 from t2; " +
                  "open cursor1; " +
                  "open cursor2; " +
                  "end;";
//Returns multiple result sets by opened cursors
DB2DataReader dr = cmd.ExecuteReader();
while (dr.Read())
{
    //Process read data if required from cursor1
}
dr.NextResult(); //Get next result set
while (dr.Read())
{
    //Process read data if required from cursor2
}
dr.Close();
...
```

Retrieving multiple result sets from execution of anonymous block by using the DB2ResultSet class in C#:

```
...
cmd.CommandText = " begin " +
                  "declare cursor1 cursor with return to client with hold for select c1 from t1; " +
                  "declare cursor2 cursor with return to client for select c2 from t2; " +
                  "open cursor1; " +
                  "open cursor2; " +
                  "end;";
//Returns multiple result sets by opened cursors
DB2ResultSet ds = cmd.ExecuteResultSet(DB2CursorType.ForwardOnly);
while (ds.Read())
{
    //Process read data if required from cursor1
}
ds.NextResult(); //Get next result set
while (ds.Read())
{
    //Process read data if required from cursor2
}
ds.Close();
...
```

Reading result sets from an application using the IBM Data Server Provider for .NET

When using the IBM Data Server Provider for .NET, the reading the result sets is done through a DB2DataReader object. The DB2DataReader method, Read() is used to advance to the next row in the result set.

About this task

The methods GetString(), GetInt32(), GetDecimal(), and other methods for all of the available data types are used to extract data from the individual columns of output. The DB2DataReader's Close() method is used to close the DB2DataReader object, which should always be done when reading the output is finished.

Reading a result set in C#:

```
// assume a DB2DataReader reader
Int16 deptnum = 0;
String location="";

// Output the results of the query
while(reader.Read())
{
    deptnum = reader.GetInt16(0);
```

```

        location = reader.GetString(1);
        Console.WriteLine("      " + deptnum + " " + location);
    }
    reader.Close();

```

Reading a result set in Visual Basic .NET:

```

' assume a DB2DataReader reader
Dim deptnum As Int16 = 0
Dim location As String ""

' Output the results of the query
Do While (reader.Read())
    deptnum = reader.GetInt16(0)
    location = reader.GetString(1)
    Console.WriteLine("      " & deptnum & " " & location)
Loop
reader.Close();

```

Calling stored procedures from an application using the IBM Data Server Provider for .NET

When using the IBM Data Server Provider for .NET, you can call stored procedures by using a DB2Command object.

About this task

The default value of the CommandType property is CommandType.Text. This is the appropriate value for SQL statements and can also be used to call stored procedures. However, calling stored procedures is easier when you set CommandType to CommandType.StoredProcedure. In this case, you only need to specify the stored procedure name and any parameters.

When working with stored procedures you can pass in the parameters using host variables, named parameters, or positioned parameters. However, you cannot use them in combination within the same SQL statement.

The following C# and Visual Basic examples demonstrate how to invoke a stored procedure called INOUT_PARAM, with the CommandType property set to either CommandType.StoredProcedure or CommandType.Text.

Procedure

- In C#, call a stored procedure by setting the CommandType property of the DB2Command to CommandType.Text:

```

// assume a DB2Connection conn
DB2Transaction trans = conn.BeginTransaction();
DB2Command cmd = conn.CreateCommand();
String procName = "INOUT_PARAM";
String procCall = "CALL INOUT_PARAM (@param1, @param2, @param3)";
cmd.Transaction = trans;
cmd.CommandType = CommandType.Text;
cmd.CommandText = procCall;

// Register input-output and output parameters for the DB2Command
cmd.Parameters.Add( new DB2Parameter("@param1", "Value1");
cmd.Parameters.Add( new DB2Parameter("@param2", "Value2");
DB2Parameter param3 = new DB2Parameter("@param3", IfxType.Integer);
param3.Direction = ParameterDirection.Output;
cmd.Parameters.Add( param3 );

// Call the stored procedure
Console.WriteLine(" Call stored procedure named " + procName);
cmd.ExecuteNonQuery();

```

- In Visual Basic, call a stored procedure by setting the CommandType property of the DB2Command to CommandType.Text:

```
' assume a DB2Connection conn
Dim trans As DB2Transaction = conn.BeginTransaction()
Dim cmd As DB2Command = conn.CreateCommand()
Dim procName As String = "INOUT_PARAM"
Dim procCall As String = "CALL INOUT_PARAM (?, ?, ?)"
cmd.Transaction = trans
cmd.CommandType = CommandType.Text
cmd.CommandText = procCall

' Register input-output and output parameters for the DB2Command
...

' Call the stored procedure
Console.WriteLine(" Call stored procedure named " & procName)
cmd.ExecuteNonQuery()
```

Note: Both CALL and EXECUTE PROCEDURE are supported.

- In C#, call a stored procedure by setting the CommandType property of the DB2Command to CommandType.StoredProcedure. Named parameters are not supported when using this method:

```
// assume a DB2Connection conn
DB2Transaction trans = conn.BeginTransaction();
DB2Command cmd = conn.CreateCommand();
String procName = "INOUT_PARAM";
cmd.Transaction = trans;
cmd.CommandType = CommandType.StoredProcedure;
cmd.CommandText = procName;

// Register input-output and output parameters for the DB2Command
...

// Call the stored procedure
Console.WriteLine(" Call stored procedure named " + procName);
cmd.ExecuteNonQuery();
```

- In Visual Basic, call a stored procedure by setting the CommandType property of the DB2Command to CommandType.StoredProcedure:

```
' assume a DB2Connection conn
Dim trans As DB2Transaction = conn.BeginTransaction()
Dim cmd As DB2Command = conn.CreateCommand()
Dim procName As String = "INOUT_PARAM"
cmd.Transaction = trans
cmd.CommandType = CommandType.StoredProcedure
cmd.CommandText = procName

' Register input-output and output parameters for the DB2Command
...

' Call the stored procedure
Console.WriteLine(" Call stored procedure named " & procName)
cmd.ExecuteNonQuery()
```

- Parameters can be passed to the stored procedure in any order, when qualified by the parameter name. This Named Argument feature is supported only by DB2 for Linux, UNIX, and Windows data servers. For example, in the following SQL statement a stored procedure is defined and then called with the parameters in a different order:

```
CREATE PROCEDURE schema.my_proc ( IN var1 int, INOUT var2 int )
LANGUAGE SQL
BEGIN
-- procedure code here
```

```
END
```

```
CALL my_proc (var2=>@param2, var1=>@param1)
```

Simultaneously accessing the result sets returned by CURSOR type output parameters

When using the IBM Data Server Provider for .NET, the DB2Type.Cursor is specified to simultaneously access all the cursors in output parameters.

About this task

For Stored procedure that has multiple CURSOR type output parameters, binding DB2TYPE.Cursor to the output parameter object allows simultaneous access to all the cursors in output parameters.

For example, OrderDetails stored procedure declares three cursors, each giving relevant information about the product and its sales.

```
CREATE OR REPLACE TYPE cur AS CURSOR
CREATE PROCEDURE OrderDetails (p_startDate TIMESTAMP, p_endDate TIMESTAMP,
    OUT prodDetails cur, OUT prodOrderDetails cur, OUT custOrderDetails cur)
LANGUAGE SQL
BEGIN
    SET prodDetails = CURSOR WITH HOLD FOR
        SELECT p.pid, price, quantity FROM products p, inventory i
        WHERE p.pid = i.pid AND p.pid IN (SELECT DISTINCT pid FROM orders) ORDER BY pid;
    SET prodOrderDetails = CURSOR WITH HOLD FOR
        SELECT pid, COUNT(*), SUM(quantity) FROM orders
        WHERE date >= p_startDate AND date <= p_endDate GROUP BY pid ORDER BY pid;
    SET custOrderDetails = CURSOR WITH HOLD FOR
        SELECT pid, custID, COUNT(*), SUM(quantity) FROM orders
        WHERE date >= p_startDate AND date <= p_endDate
        GROUP BY pid, custID ORDER BY pid, custID;
    OPEN prodDetails;
    OPEN prodOrderDetails;
    OPEN custOrderDetails;
END;
```

The caller needs to access the cursors simultaneously so that it can gather the relevant information for a particular product from each of the cursors and calculate the discount. To provide simultaneous access to the cursors, the stored procedure returns the cursors as output parameters. The application must set the DB2Type to DB2Type.Cursor when binding the CURSOR type output parameters for simultaneous access to occur.

```
//C# Code sample
cmd.CommandText = "CALL OrderDetails(
    @p_startDate, @p_endDate, @prodDetails, @prodOrderDetails, @custOrderDetails)";
cmd.Parameters.Add("@p_startDate", DateTime.Parse("1/1/2010"));
cmd.Parameters.Add("@p_endDate", DateTime.Parse("12/31/2010"));
cmd.Parameters.Add("@prodDetails", DB2Type.Cursor);
cmd.Parameters["@prodDetails"].Direction = ParameterDirection.Output;
cmd.Parameters.Add("@prodOrderDetails", DB2Type.Cursor);
cmd.Parameters["@prodOrderDetails"].Direction = ParameterDirection.Output;
cmd.Parameters.Add("@custOrderDetails", DB2Type.Cursor);
cmd.Parameters["@custOrderDetails"].Direction = ParameterDirection.Output;
cmd.ExecuteNonQuery();
DB2DataReader prodDetailsDR =
    (DB2DataReader)cmd.Parameters["@prodDetails"].Value;
DB2DataReader prodOrderDetailsDR =
    (DB2DataReader)cmd.Parameters["@prodOrderDetails"].Value;
DB2DataReader custOrderDetailsDR =
    (DB2DataReader)cmd.Parameters["@custOrderDetails"].Value;

while (prodOrderDetailsDR.Read())
{
    pid = prodOrderDetailsDR.GetInt32(0);
    numOrders = prodOrderDetailsDR.GetInt32(1);
    totalOrderQuantity = prodOrderDetailsDR.GetInt32(2);
    prodDetailsDR.Read();
    price = prodDetailsDR.GetDecimal(1);
    currentInventory = prodDetailsDR.GetInt32(2);
    int totalCustOrders = 0;
    while (custOrderDetailsDR.Read())
    {
        custID = custOrderDetailsDR.GetInt32(1);
        numOrdersByCust = custOrderDetailsDR.GetInt32(2);
        totalCustOrders += numOrdersByCust;
        totalOrderQuantityByCust = custOrderDetailsDR.GetInt32(3);
        //Calculate discount based on numOrders, numOrdersByCust,
        // totalOrderQuantity, totalOrderQuantityByCust, price and currentInventory
        if (totalCustOrders == numOrders) //done with this pid
            break;
    }
}
```

```

}
prodDetailsDR.Close();
prodOrderDetailsDR .Close();
custOrderDetailsDR .Close();

```

The data reader from a cursor type output parameter can be accessed from the Value property only after invoking the ExecuteNonQuery method.

If the command is executed using either the ExecuteReader or ExecuteResultSet methods, the result sets are returned in the DB2DataReader or DB2ResultSet object. The subsequent result sets must be accessed sequentially by calling the NextResult method. Although the output parameters have been bound, accessing the output parameter Value property will result in an InvalidOperationException exception because the query was not executed with the ExecuteNonQuery method.

When working with cursors simultaneously, the application might want to commit the work done before continuing with reading the cursor. For application to issue commit without destroying the open cursor, the cursor must be declared as holdable within the stored procedure.

Tracing IBM Data Server Provider for .NET with System.Diagnostics.Trace

This topic talks about tracing the IBM Data Server Provider for .NET with System.Diagnostics.Trace.

About this task

Function entry and exit trace points in the .NET public API will be traced using System.Diagnostics.Trace. Only first level calls will be traced, for example, only the provider calls explicitly made by the customer's application will be traced. Any calls to the public API made internally by the provider itself will not be traced.

For the IBM.Data.DB2 namespace, the TraceSwitch will be named DB2NMPTRACE and for the IBM.Data.Informix namespace, the TraceSwitch will be named IFXNMPTRACE.

```

<configuration>
  <system.diagnostics>
    <switches>
      <add name="DB2NMPTRACE" value="1" />
    </switches>
  </system.diagnostics>
</configuration>

```

The TraceSwitch class uses the System.Diagnostics.TraceLevel enumeration to control the level of information that is traced - 0 through 4 representing Off, Error, Warning, Info, and Verbose. For the .NET provider, 0 will represent Off and 1 through 4 will represent On.

Example

The following output is an example of what the trace output for a simple program might look like:

Note: Trace points dumped by application itself are prefixed with *.

```

* * Started tracing program
* Creating connection
DB2Connection.DB2Connection1 api entry - database=nmpfvту;
DB2Connection.DB2Connection1 api exit, rc = 0
* Opening connection

```

```
DB2Connection.Open api entry
DB2Connection.Open api exit, rc = 0
* Closing connection
DB2Connection.Close api entry
DB2Connection.Close api exit, rc = 0
* Ending program
DB2Connection.~DB2Connection api entry
DB2Connection.~DB2Connection api exit, rc = 0
DB2Connection.Dispose api entry
DB2Connection.Dispose api exit, rc = 0
```

Optimizing queries in .NET applications using pureQuery

The .NET client drivers can leverage features found in pureQuery technology. These features enables existing .NET application queries to execute as static SQL. Static queries avoid the need to prepare certain statements at runtime. This can lead to improved security and performance in your applications.

For more information, see <http://www.ibm.com/support/docview.wss?uid=swg27023946>

IBM Data Server Provider for .NET support for Microsoft Entity Framework

Take advantage of the Microsoft ADO.NET Entity Framework with IBM data servers by using the IBM Data Server Provider for .NET. You can generate EDM schemas, as well as write and execute Entity SQL and LINQ statements to Entities applications, with the supported server versions.

System requirements

IBM Data Server Provider for .NET works with the following IBM data servers:

- All supported levels of IBM DB2 for Linux, UNIX, and Windows
- DB2 for i Version 5 Release 4, Version 6 Release 1, and Version 7 Release 1, through DB2 Connect (for IBM DB2 Version 9.7 Fix Pack 4 and later versions)
- DB2 for z/OS, Version 8 or later
- IBM Informix, Version 11.170 or later

You must have Microsoft .NET Framework 3.5 SP1 or later with the Microsoft ADO.NET Entity Framework. To manipulate entity data models by using the Microsoft Entity Data Model Wizard or ADO.NET Entity Designer, you also require Microsoft Visual Studio 2008 or later.

The following table lists the canonical functions that the IBM entity provider supports. Canonical functions are translated to the corresponding data source functions by the data provider.

Table 1. IBM entity provider support for canonical functions

Canonical function type	LINQ function	DB2 for Linux, UNIX, and Windows	DB2 for z/OS	DB2 for i	Informix
Aggregate	Average	Y	Y	Y	Y
	BigCount	Y	Y	Y	Y
	Count	Y	Y	Y	Y
	Maximum	Y	Y	Y	Y
	Minimum	Y	Y	Y	Y
	NewGuid	Y*	Y*	Y*	Y*
	StDev	Y	Y	Y	Y
	StDevP	Y	Y	Y	Y
	Sum	Y	Y	Y	Y
	Var	Y	Y	Y	Y
	VarP	Y	Y	Y	Y
Bitwise	BitWiseAnd	Y	Y*	Y*	Y
	BitWiseNot	Y	Y*	Y*	Y
	BitWiseOr	Y	Y*	Y*	Y
	BitWiseXor	Y	Y*	Y*	Y
Math	Abs	Y	Y	Y	Y
	Ceiling	Y	Y	Y	Y
	Floor	Y	Y	Y	Y
	Power	Y	Y	Y	Y
	Round (value,digits)	Y	Y	Y	Y
	Truncate (value,digits)	Y	Y	Y	Y
String	Concat	Y	Y	Y	Y
	Contains	Y	Y	Y	Y*
	EndsWith	Y	Y	Y	Y
	IndexOf	Y	Y	Y	Y*
	Left	Y	Y	Y	Y
	Length	Y	Y	Y	Y
	LTrim	Y	Y	Y	Y
	Replace	Y	Y	Y	Y
	Right	Y	Y	Y	Y
	RTrim	Y	Y	Y	Y
	StartsWith	Y	Y	Y	Y
	Substring	Y	Y	Y	Y
	ToLower	Y	Y	Y	Y
	ToUpper	Y	Y	Y	Y
Trim	Y	Y	Y	Y	

Table 1. IBM entity provider support for canonical functions (continued)

Canonical function type	LINQ function	DB2 for Linux, UNIX, and Windows	DB2 for z/OS	DB2 for i	Informix
Datetime	AddNanoseconds	Y	Y	Y	Y
	AddMicroseconds	Y	Y	Y	Y
	AddMilliseconds	Y	Y	Y	Y
	AddSeconds	Y	Y	Y	Y
	AddMinutes	Y	Y	Y	Y
	AddHours	Y	Y	Y	Y
	AddDays	Y	Y	Y	Y
	AddMonths	Y	Y	Y	Y
	AddYears	Y	Y	Y	Y
	CreateDateTime	Y	Y	Y	Y
	CreateDateTimeOffset			Y	
	CurrentDateTimeOffset			Y	
	CreateTime	Y	Y	Y	Y
	CurrentDateTime	Y	Y	Y	Y
	CurrentUtcDateTime	Y	Y	Y	
	Day	Y	Y	Y	Y
	DayOfYear	Y	Y	Y	Y
	DiffNanoseconds	Y	Y	Y	Y*
	DiffMicroseconds	Y	Y	Y	Y*
	DiffMilliseconds	Y	Y	Y	Y*
	DiffSeconds	Y	Y	Y	Y*
	DiffMinutes	Y	Y	Y	Y*
	DiffHours	Y	Y	Y	Y*
	DiffDays	Y	Y	Y	Y*
	DiffMonths	Y	Y	Y	Y*
	DiffYears	Y	Y	Y	Y*
	GetTotalOffsetMinutes			Y	
	Hour	Y	Y	Y	Y
	Millisecond	Y	Y	Y	Y
	Minute	Y	Y	Y	Y
	Month	Y	Y	Y	Y
	Second	Y	Y	Y	Y
	Truncate (datetime exp)	Y	Y	Y	Y
	Year	Y	Y	Y	Y

Known limitations

Important: Some of the canonical functions depend on the server. Error SQL0440N* indicates that your server does not support certain functionality. To get more information about the supported functionality, contact IBM technical support for your server.

The following known limitations apply to IBM Data Server Provider for .NET support for the Microsoft ADO.NET Entity Framework:

Generic:

- Only database first scenarios are supported; all database objects must exist before you use the Entity Framework.
- Invocation of store-specific functions is not supported.

- Trusted context connection properties that you set in the Server Explorer Add Connection dialog are not passed to the Entity Framework connections.

DB2 for z/OS:

- Data type REAL is not supported. Applications must either use the FLOAT data type in the schema of the table or specify the type as FLOAT in the client schema (EDM) even if the actual type on the server is REAL.
- Version 8 and Version 7 specific: An exception indicating some kind of syntax error might be generated for queries that include Take, Top, First, Intersect, or Except expressions. The result of the queries that include those expressions is undefined.

Examples:

```

1) var query = from o in context.Orders
where o.ShipCity == "Seattle"
select o;
var result = query.First();

2) var mexico =
context.OrderDetails.Where(od => od.Order.ShipCountry
== "Mexico").Select(od => od.Product);
var canada =
context.OrderDetails.Where(od => od.Order.ShipCountry
== "Canada").Select(od => od.Product);
var query = mexico.Intersect(canada);

3) var query =
context.Customers.Select(e => e).Except(context.Orders.Where
(o => o.ShipCountry == "Mexico").Select(o => o.Customer));

4) var query = context.Orders.Include("OrderDetails").Top("1");

5) var query = context.Orders.Include("OrderDetails").
Include("OrderDetails.Product").Take(3).Select(o => o);

```

Using the Enterprise Library data access module

The Enterprise Library is a collection of application blocks designed to assist developers with common development challenges. Application blocks are provided as source code that can be used as is or modified for development projects.

The Enterprise Library data access module for IBM data servers can be obtained along with other modules at <http://codeplex.com/entlibcontrib/SourceControl/PatchList.aspx>.

For information about how to install and use the Enterprise Library data access module with IBM data servers (DB2, Informix database server, and U2), see the readme file found in the download package.

Resources

Below are several online resources that describe how to use the data access modules:

- EntLib Contrib Project Homepage: <http://www.codeplex.com/entlibcontrib>
- patterns & practices for Enterprise Library: <http://www.codeplex.com/entlib>
- Microsoft Enterprise Library Homepage: <http://msdn.microsoft.com/en-us/library/cc467894.aspx>

- IBM DB2 for .NET: <http://www.ibm.com/software/data/db2/windows/dotnet.html>

Building .NET Applications

Resources for building .NET applications.

Building Visual Basic .NET applications

DB2 products provide a bldapp.bat batch file for compiling and linking DB2 Visual Basic .NET applications.

This file is located in the sqllib\samples\.NET\vb directory along with sample programs that can be built with this file. The batch file takes one parameter, %1, for the name of the source file to be compiled (without the .vb extension).

About this task

This task will take you through the basic steps of building a Visual Basic .NET application using bldapp.bat with the DbAuth sample file.

Procedure

To build the program, DbAuth, from the source file, DbAuth.vb, enter:

```
bldapp DbAuth
```

To ensure you have the parameters you need when you run the executable, you can specify different combinations of parameters depending on the number entered:

1. No parameters. Enter just the program name:
DbAuth
2. One parameter. Enter the program name plus the database alias:
DbAuth <db_alias>
3. Two parameters. Enter the program name plus user ID and password:
DbAuth <userid> <passwd>
4. Three parameters. Enter the program name plus the database alias, user ID, and password:
DbAuth <db_alias> <userid> <passwd>
5. Four parameters. Enter the program name plus server name, port number, user ID, and password:
DbAuth <server> <portnum> <userid> <passwd>
6. Five parameters. Enter the program name plus database alias, server name, port number, user ID, and password:
DbAuth <db_alias> <server> <portnum> <userid> <passwd>

What to do next

To build and run the LCTrans sample program, you need to follow more detailed instructions given in the source file, LCTrans.vb.

Building C# .NET applications

DB2 products provide a bldapp.bat batch file for compiling and linking DB2 C# .NET applications. This batch file is located in the sqllib\samples\.NET\cs directory along with sample programs that can be built with this file.

The batch file takes one parameter, %1, for the name of the source file to be compiled (without the .cs extension).

About this task

This task will take you through the basic steps of building a C# .NET application using bldapp.bat with the DbAuth sample file.

Procedure

To build the program, DbAuth, from the source file, DbAuth.cs, enter:

```
bldapp DbAuth
```

To ensure you have the parameters you need when you run the executable, you can specify different combinations of parameters depending on the number entered:

1. No parameters. Enter just the program name:
DbAuth
2. One parameter. Enter the program name plus the database alias:
DbAuth <db_alias>
3. Two parameters. Enter the program name plus user ID and password:
DbAuth <userid> <passwd>
4. Three parameters. Enter the program name plus the database alias, user ID, and password:
DbAuth <db_alias> <userid> <passwd>
5. Four parameters. Enter the program name plus server name, port number, user ID, and password:
DbAuth <server> <portnum> <userid> <passwd>
6. Five parameters. Enter the program name plus database alias, server name, port number, user ID, and password:
DbAuth <db_alias> <server> <portnum> <userid> <passwd>

What to do next

To build and run the LCTrans sample program, you need to follow more detailed instructions given in the source file, LCTrans.cs.

Visual Basic .NET application compile and link options

This topic describes the various options available when compiling and linking Visual Basic .NET applications.

The following compile and link options are available for building Visual Basic .NET applications on Windows with the Microsoft Visual Basic .NET compiler, as demonstrated in the bldapp.bat batch file.

Note: The .NET Framework Version 1.1 is supported only with the .NET Provider Version 9.5 and earlier.

Compile and link options for stand-alone VB .NET applications using bldapp

Compile and link options for stand-alone VB .NET applications:

%BLDCOMP%

Variable for the compiler. The default is vbc, the Microsoft Visual Basic .NET compiler.

/r:"%DB2PATH%\bin\%VERSION%\IBM.Data.DB2.d11

Reference the DB2 dynamic link library for the .NET framework version that you are using.

%DB2PATH%

The %DB2PATH% variable represents root path of the DB2 product installation. The %DB2PATH% variable is not present on IBM Data Server Driver for ODBC and CLI or Data Server Driver Package installation. When using IBM IBM Data Server Driver for ODBC and CLI or Data Server Driver Package replace %DB2PATH% with a path where driver product is installed.

%VERSION%

There are several supported versions of the .NET framework for applications. DB2 has a dynamic link library for each. For .NET Framework Version 2.0, 3.0, and 3.5, %VERSION% points to the netf20\ sub-directory.

Compile and link options for the loosely-coupled sample program, LCTrans using bldapp:**%BLDCOMP%**

Variable for the compiler. The default is vbc, the Microsoft Visual Basic .NET compiler.

/out:RootCOM.d11

Output the RootCOM dynamic link library, used by the LCTrans application, from the RootCOM.vb source file,

/out:SubCOM.d11

Output the SubCOM dynamic link library, used by the LCTrans application, from the SubCOM.vb source file,

/target:library %1.cs

Create the dynamic link library from the input source file (RootCOM.vb or SubCOM.vb).

/r:System.EnterpriseServices.d11

Reference the Microsoft Windows System EnterpriseServices data link library.

/r:"%DB2PATH%\bin\%VERSION%\IBM.Data.DB2.d11

Reference the DB2 dynamic link library for the .NET framework version you are using.

%DB2PATH%

The %DB2PATH% variable represents root path of the DB2 product installation. The %DB2PATH% variable is not present on IBM Data Server Driver for ODBC and CLI or Data Server Driver Package installation. When using IBM IBM Data Server Driver for ODBC and CLI or Data Server Driver Package replace %DB2PATH% with a path where driver product is installed.

%VERSION%

There are several supported versions of the .NET framework for

applications. DB2 has a dynamic link library for each. For .NET Framework Version 2.0 and 3.0, %VERSION% points to the netf20\ sub-directory.

/r:System.Data.d11

Reference the Microsoft Windows System Data dynamic link library.

/r:System.d11

Reference the Microsoft Windows System dynamic link library.

/r:System.Xml.d11

Reference the Microsoft Windows System XML dynamic link library (for SubCOM.vb).

/r:SubCOM.d11

Reference the SubCOM dynamic link library (for RootCOM.vb and LCTrans.vb).

/r:RootCOM.d11

Reference the RootCOM dynamic link library (for LCTrans.vb).

Refer to your compiler documentation for additional compiler options.

C# .NET application compile and link options

This topic describes the various options available when compiling and linking C# .NET applications.

The compile and link options available to DB2 for building C# applications on Windows with the Microsoft C# compiler, as demonstrated in the bldapp.bat batch file.

Note: The .NET Framework Version 1.1 is supported only with the .NET Provider Version 9.5 and earlier.

Compile and link options for stand-alone C# applications using bldapp:

Compile and link options for stand-alone C# applications:

%BLDCOMP%

Variable for the compiler. The default is csc, the Microsoft C# compiler.

/r:"%DB2PATH%\bin\%VERSION%IBM.Data.DB2.d11

Reference the DB2 dynamic link library for the .NET framework version you are using.

%VERSION%

There are several supported versions of the .NET framework for applications. DB2 has a dynamic link library for each version. For .NET Framework Version 2.0, 3.0, and 3.5, %VERSION% points to the netf20\ sub-directory.

Compile and link options for the loosely-coupled sample program, LCTrans using bldapp:

%BLDCOMP%

Variable for the compiler. The default is csc, the Microsoft C# compiler.

/out:RootCOM.d11

Output the RootCOM dynamic link library, used by the LCTrans application, from the RootCOM.cs source file,

/out:SubCOM.dll

Output the SubCOM dynamic link library, used by the LCTrans application, from the SubCOM.cs source file,

/target:library %1.cs

Create the dynamic link library from the input source file (RootCOM.cs or SubCOM.cs).

/r:System.EnterpriseServices.dll

Reference the Microsoft Windows System EnterpriseServices data link library.

/r:"%DB2PATH%\bin\%VERSION%IBM.Data.DB2.dll

Reference the DB2 dynamic link library for the .NET framework version you are using.

%VERSION%

There are several supported versions of the .NET framework for applications. DB2 has a dynamic link library for each. For .NET Framework Version 2.0, 3.0, and 3.5, %VERSION% points to the netf20\ sub-directory.

/r:System.Data.dll

Reference the Microsoft Windows System Data dynamic link library.

/r:System.dll

Reference the Microsoft Windows System dynamic link library.

/r:System.Xml.dll

Reference the Microsoft Windows System XML dynamic link library (for SubCOM.cs).

/r:SubCOM.dll

Reference the SubCOM dynamic link library (for RootCOM.cs and LCTrans.cs).

/r:RootCOM.dll

Reference the RootCOM dynamic link library (for LCTrans.cs).

Refer to your compiler documentation for additional compiler options.

OLE DB .NET Data Provider

The OLE DB .NET Data Provider uses the IBM DB2 OLE DB Driver, which is referred to in a `ConnectionString` object as `IBMDADB2`.

The connection string keywords supported by the OLE DB .NET Data Provider are the same as those supported by the IBM OLE DB Provider for DB2. This provider is no longer tested. Users are recommended to use the IBM Data Server Provider for .NET.

Also, the OLE DB .NET Data Provider has the same restrictions as the IBM DB2 OLE DB Provider. There are additional restrictions for the OLE DB .NET Data Provider, which are identified in the topic: "OLE DB .NET Data Provider restrictions" in *Developing ADO.NET and OLE DB Applications*.

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

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

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

*Table 2. Useful **ConnectionString** keywords for the OLE DB .NET Data Provider*

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

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

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

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

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

OLE DB .NET Data Provider restrictions

The OLE DB .NET Data Provider is no longer tested. Users are recommended to use the IBM Data Server Provider for .NET.

The following table identifies usage restrictions for the OLE DB .NET Data Provider:

Table 3. OLE DB .NET Data Provider restrictions

Class or feature	Restriction description	DB2 servers affected
ASCII character streams	You cannot use ASCII character streams with <code>OleDbParameters</code> when using <code>DbType.AnsiString</code> or <code>DbType.AnsiStringFixedLength</code> . The OLE DB .NET Data Provider will throw the following exception: "Specified cast is not valid" Workaround: Use <code>DbType.Binary</code> instead of using <code>DbType.AnsiString</code> or <code>DbType.AnsiStringFixedLength</code> .	All
ADORecord	ADORecord is not supported.	All

Table 3. OLE DB .NET Data Provider restrictions (continued)

Class or feature	Restriction description	DB2 servers affected
ADORecordSet and Timestamp	<p>As documented in MSDN, the ADORecordSet variant time resolves to one second. Consequently, all fractional seconds are lost when a DB2 Timestamp column is stored into a ADORecordSet. Similarly, after filling a DataSet from a ADORecordSet, the Timestamp columns in the DataSet will not have any fractional seconds.</p> <p>Workaround: This workaround only works for DB2 Universal Database for Linux, UNIX, and Windows, Version 8.1, FixPak 4 or later. In order to avoid the loss of fraction of seconds, you can set the following CLI keyword: MAPTIMESTAMPDESCRIBE = 2</p> <p>This keyword will describe the Timestamp as a WCHAR(26). To set the keyword, execute the following command from a DB2 Command Window: db2 update cli cfg for section common using MAPTIMESTAMPDESCRIBE 2</p>	All
Chapters	Chapters are not supported.	All
Key information	The OLE DB .NET Data Provider cannot retrieve key information when opening an IDataReader at the same time.	DB2 for VM/VSE
Key information from stored procedures	<p>The OLE DB .NET Data Provider can retrieve key information about a result set returned by a stored procedure only from DB2 for Linux, UNIX, and Windows. This is because the DB2 servers for platforms other than Linux, UNIX, and Windows do not return extended describe information for the result sets opened in the stored procedure.</p> <p>In order to retrieve key information of a result set returned by a stored procedure on DB2 for Linux, UNIX, and Windows, you need to set the following registry variable on the DB2 server: db2set DB2_APM_PERFORMANCE=8</p> <p>Setting this server-side DB2 registry variable will keep the result set meta-data available on the server for a longer period of time, thus allowing OLE DB to successfully retrieve the key information. However, depending on the server workload, the meta-data might not be available long enough before the OLE DB Provider queries for the information. As such, there is no guarantee that the key information will always be available for result sets returned from a store procedure.</p> <p>In order to retrieve any key information about a CALL statement, the application must execute the CALL statement. Calling OleDbDataAdapter.FillSchema() or OleDbCommand.ExecuteReader(CommandBehavior.SchemaOnly CommandBehavior.KeyInfo), will not actually execute the stored procedure call. Therefore, you will not retrieve any key information for the result set that is to be returned by the stored procedure.</p>	All
Key information from batched SQL statements	<p>When using batched SQL statements that return multiple results, the FillSchema() method attempts to retrieve schema information only for the first SQL statement in the batched SQL statement list. If this statement does not return a result set then no table is created. For example:</p> <pre>[C#] cmd.CommandText = "INSERT INTO ORG(C1) VALUES(1000); SELECT C1 FROM ORG;"; da = new OleDbDataAdapter(cmd); da.FillSchema(ds, SchemaType.Source);</pre> <p>No table will be created in the data set because the first statement in the batch SQL statement is an "INSERT" statement, which does not return a result set.</p>	All

Table 3. OLE DB .NET Data Provider restrictions (continued)

Class or feature	Restriction description	DB2 servers affected
OleDbCommandBuilder	<p>The UPDATE, DELETE and INSERT statements automatically generated by the OleDbCommandBuilder are incorrect if the SELECT statement contains any columns of the following data types:</p> <ul style="list-style-type: none"> • CLOB • BLOB • DBCLOB • LONG VARCHAR • LONG VARCHAR FOR BIT DATA • LONG VARGRAPHIC <p>If you are connecting to a DB2 server other than DB2 for Linux, UNIX, and Windows, then columns of the following data types also cause this problem:</p> <ul style="list-style-type: none"> • VARCHAR¹ • VARCHAR FOR BIT DATA¹ • VARGRAPHIC¹ • REAL • FLOAT or DOUBLE • TIMESTAMP <p>Note:</p> <p>1. Columns of these data types are applicable if they are defined to be VARCHAR values greater than 254 bytes, VARCHAR values FOR BIT DATA greater than 254 bytes, or VARGRAPHICs greater than 127 bytes. This condition is only valid if you are connecting to a DB2 server other than DB2 for Linux, UNIX, and Windows.</p> <p>The OleDbCommandBuilder generates SQL statements that use all of the selected columns in an equality comparison in the WHERE clause, but the data types listed previously cannot be used in an equality comparison.</p> <p>Note: Note that this restriction will affect the IDbDataAdapter.Update() method that relies on the OleDbCommandBuilder to automatically generate the UPDATE, DELETE, and INSERT statements. The UPDATE operation will fail if the generated statement contains any one of the data types listed previously.</p> <p>Workaround: You will need to explicitly remove all columns that are of the data types listed previously from the WHERE clause of the generated SQL statement. It is recommended that you code your own UPDATE, DELETE and INSERT statements.</p>	All
OleDbCommandBuilder. DeriveParameters	<p>Case-sensitivity is important when using DeriveParameters(). The stored procedure name specified in the OleDbCommand.CommandText needs to be in the same case as how it is stored in the DB2 system catalog tables. To see how stored procedure names are stored, call OpenSchema(OleDbSchemaGuid.Procedures) without supplying the procedure name restriction. This will return all the stored procedure names. By default, DB2 stores stored procedure names in uppercase, so most often, you need to specify the stored procedure name in uppercase.</p>	All
OleDbCommandBuilder. DeriveParameters	<p>The OleDbCommandBuilder.DeriveParameters() method does not include the ReturnValue parameter in the generated OleDbParameterCollection. SqlClient and the IBM Data Server Provider for .NET by default adds the parameter with ParameterDirection.ReturnValue to the generated ParameterCollection.</p>	All
OleDbCommandBuilder. DeriveParameters	<p>The OleDbCommandBuilder.DeriveParameters() method will fail for overloaded stored procedures. If you have multiple stored procedures of the name "MYPROC" with each of them taking a different number of parameters or different type of parameter, the OleDbCommandBuilder.DeriveParameters() will retrieve all the parameters for all the overloaded stored procedures.</p>	All
OleDbCommandBuilder. DeriveParameters	<p>If the application does not qualify a stored procedure with a schema, DeriveParameters() will return all the parameters for that procedure name. Therefore, if multiple schemas exist for the same procedure name, DeriveParameters() will return all the parameters for all the procedures with the same name.</p>	All

Table 3. OLE DB .NET Data Provider restrictions (continued)

Class or feature	Restriction description	DB2 servers affected
OLEDBConnection. ChangeDatabase	The OLEDBConnection.ChangeDatabase() method is not supported.	All
OLEDBConnection. ConnectionString	<p>Use of nonprintable characters such as '\b', '\a' or '\O' in the connection string will cause an exception to be thrown.</p> <p>The following keywords have restrictions:</p> <p>Data Source The data source is the name of the database, not the server. You can specify the SERVER keyword, but it is ignored by the IBMDADB2 provider.</p> <p>Initial Catalog and Connect Timeout These keywords are not supported. In general, the OLE DB .NET Data Provider will ignore all unrecognized and unsupported keywords. However, specifying these keywords will cause the following exception: Multiple-step OLE DB operation generated errors. Check each OLE DB status value, if available. No work was done.</p> <p>ConnectionTimeout ConnectionTimeout is read only.</p>	All
OLEDBConnection. GetOLEDBSchemaTable	<p>Restriction values are case-sensitive, and need to match the case of the database objects stored in the system catalog tables, which defaults to uppercase.</p> <p>For instance, if you have created a table in the following manner: CREATE TABLE abc(c1 SMALLINT)</p> <p>DB2 will store the table name in uppercase ("ABC") in the system catalog. Therefore, you will need to use "ABC" as the restriction value. For instance: <pre>schemaTable = con.GetOLEDBSchemaTable(OLEDBSchemaGuid.Tables, new object[] { null, null, "ABC", "TABLE" });</pre> </p> <p>Workaround: If you need case-sensitivity or spaces in your data definitions, you must put quotation marks around them. For example: <pre>cmd.CommandText = "create table \"Case Sensitive\"(c1 int)"; cmd.ExecuteNonQuery(); tablename = "\"Case Sensitive\""; schemaTable = con.GetOLEDBSchemaTable(OLEDBSchemaGuid.Tables, new object[] { null, null, tablename, "TABLE" });</pre> </p>	All
OLEDBDataAdapter and DataColumnMapping	<p>The source column name is case-sensitive. It needs to match the case as stored in the DB2 catalogs, which by default is uppercase.</p> <p>For example: <pre>colMap = new DataColumnMapping("EMPNO", "Employee ID");</pre> </p>	All
OLEDBDataReader. GetSchemaTable	<p>The OLE DB .NET Data Provider is not able to retrieve extended describe information from servers that do not return extended describe information. If you are connecting to a server that does not support extended describe (the affected servers), the following columns in the metadata table returned from IDataReader.GetSchemaTable() are invalid:</p> <ul style="list-style-type: none"> • IsReadOnly • IsUnique • IsAutoIncrement • BaseSchemaName • BaseCatalogName 	DB2 for OS/390®, V7 or earlier DB2 for OS/400 DB2 for VM/VSE
Stored procedures: no column names for result sets	The DB2 for OS/390 version 6.1 server does not return column names for result sets returned from a stored procedure. The OLE DB .NET Data Provider maps these unnamed columns to their ordinal position (for example, "1", "2" "3"). This is contrary to the mapping documented in MSDN: "Column1", "Column2", "Column3".	DB2 for OS/390 version 6.1

Hints and tips

Connection pooling in OLE DB .NET Data Provider applications

The OLE DB .NET Data Provider automatically pools connections using OLE DB session pooling.

The OLE DB .NET Data Provider automatically pools connections using OLE DB session pooling. Connection string arguments can be used to enable or disable OLE DB services including pooling. For example, the following connection string will disable OLE DB session pooling and automatic transaction enlistment.

```
Provider=IBMDADB2;OLE DB Services=-4;Data Source=SAMPLE;
```

The following table describes the ADO connection string attributes you can use to set the OLE DB services:

Table 4. Setting OLE DB services by using ADO connection string attributes

Services enabled	Value in connection string
All services (the default)	"OLE DB Services = -1;"
All services except pooling	"OLE DB Services = -2;"
All services except pooling and auto-enlistment	"OLE DB Services = -4;"
All services except client cursor	"OLE DB Services = -5;"
All services except client cursor and pooling	"OLE DB Services = -6;"
No services	"OLE DB Services = 0;"

For more information about OLE DB session pooling or resource pooling, as well as how to disable pooling by overriding OLE DB provider service defaults, see the OLE DB Programmer's Reference in the MSDN library located at:

<http://msdn.microsoft.com/library>

Time columns in OLE DB .NET Data Provider applications

The following sections describe how to implement time columns in OLE DB .NET Data Provider applications.

The following sections describe how to implement time columns in OLE DB .NET Data Provider applications.

Inserting using parameter markers

You want to insert a time value into a Time column:

```
command.CommandText = "insert into mytable(c1) values( ? )";
```

where column c1 is a Time column. Here are two methods to bind a time value to the parameter marker:

```
Using OleDbParameter.OleDbType = OleDbType.DBTime
```

Because `OleDbType.DBTime` maps to a `TimeSpan` object, you must supply a `TimeSpan` object as the parameter value. The parameter value cannot be a `String` or a `DateTime` object, it must be a `TimeSpan` object. For example:

```
p1.OleDbType = OleDbType.DBTime;  
p1.Value = TimeSpan.Parse("0.11:20:30");  
rowsAffected = cmd.ExecuteNonQuery();
```

The format of the TimeSpan is represented as a string in the format "[-]d.hh:mm:ss.ff" as documented in the MSDN documentation.

```
Using OleDbParameter.OleDbType = OleDbType.DateTime
```

This will force the OLE DB .NET Data Provider to convert the parameter value to a DateTime object, instead of a TimeSpan object, consequently the parameter value can be any valid string/object that can be converted into a DateTime object. This means values such as "11:20:30" will work. The value can also be a DateTime object. The value cannot be a TimeSpan object since a TimeSpan object cannot be converted to a DateTime object -- TimeSpan doesn't implement IConvertible.

For example:

```
p1.OleDbType = OleDbType.DBTimeStamp;  
p1.Value = "11:20:30";  
rowsAffected = cmd.ExecuteNonQuery();
```

Retrieval

To retrieve a time column you need to use the IDataRecord.GetValue() method or the OleDbDataReader.GetTimeSpan() method.

For example:

```
TimeSpan ts1 = ((OleDbDataReader)reader).GetTimeSpan( 0 );  
TimeSpan ts2 = (TimeSpan) reader.GetValue( 0 );
```

ADORecordset objects in OLE DB .NET Data Provider applications

When using ADORecordset objects we must consider the ADO type addBTime class.

Considerations regarding the use of ADORecordset objects.

- The ADO type addBTime class is mapped to the .NET Framework DateTime class. OleDbType.DBTime corresponds to a TimeSpan object.
- You cannot assign a TimeSpan object to an ADORecordset object's Time field. This is because the ADORecordset object's Time field expects a DateTime object. When you assign a TimeSpan object to an ADORecordset object, you will get the following message:
Method's type signature is not Interop compatible.

You can only populate the Time field with a DateTime object, or a String that can be parsed into a DateTime object.

- When you fill a DataSet with a ADORecordset using the OleDbDataAdapter, the Time field in the ADORecordset is converted to a TimeSpan column in the DataSet.
- Recordsets do not store primary keys or constraints. Therefore, no key information is added when filling out a DataSet from a Recordset using the MissingSchemaAction.AddWithKey.

ODBC .NET Data Provider

The ODBC .NET Data Provider makes ODBC calls to a DB2 data source using the CLI Driver. Therefore, the connection string keywords supported by the ODBC .NET Data Provider are the same as those supported by the CLI driver. This provider is no longer tested. Users are recommended to use the IBM Data Server Provider for .NET.

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

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

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

Table 5. Useful **ConnectionString** keywords for the ODBC .NET Data Provider

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

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

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

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

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

ODBC .NET Data Provider restrictions

The ODBC .NET Data Provider is no longer tested. Users are recommended to use the IBM Data Server Provider for .NET.

The following table identifies usage restrictions for the ODBC .NET Data Provider:

Table 6. ODBC .NET Data Provider restrictions

Class or feature	Restriction description	DB2 servers affected
ASCII character streams	<p>You cannot use ASCII character streams with <code>OdbcParameters</code> when using <code>DbType.AnsiString</code> or <code>DbType.AnsiStringFixedLength</code>.</p> <p>The ODBC .NET Data Provider will throw the following exception: "Specified cast is not valid"</p> <p>Workaround: Use <code>DbType.Binary</code> instead of using <code>DbType.AnsiString</code> or <code>DbType.AnsiStringFixedLength</code>.</p>	All

Table 6. ODBC .NET Data Provider restrictions (continued)

Class or feature	Restriction description	DB2 servers affected
Command.Prepare	<p>Before executing a command (Command.ExecuteNonQuery or Command.ExecuteReader), you must explicitly run OdbcCommand.Prepare() if the CommandText has changed since the last prepare. If you do not call OdbcCommand.Prepare() again, the ODBC .NET Data Provider will execute the previously prepared CommandText.</p> <p>For Example:</p> <pre>[C#] command.CommandText="select CLOB('ABC') from table1"; command.Prepare(); command.ExecuteReader(); command.CommandText="select CLOB('XYZ') from table2"; // This ends up re-executing the first statement command.ExecuteReader();</pre>	All
CommandBehavior.SequentialAccess	<p>When using IDataReader.GetChars() to read from a reader created with CommandBehavior.SequentialAccess, you must allocate a buffer that is large enough to hold the entire column. Otherwise, you will hit the following exception:</p> <pre>Requested range extends past the end of the array. at System.Runtime.InteropServices.Marshal.Copy(Int32 source, Char[] destination, Int32 startIndex, Int32 length) at System.Data.Odbc.OdbcDataReader.GetChars(Int32 i, Int64 dataIndex, Char[] buffer, Int32 bufferSize, Int32 length) at OLEDBTest.TestGetCharsAndBufferSize(IDbConnection con)</pre> <p>The following example demonstrates how to allocate an adequate buffer:</p> <pre>CREATE TABLE myTable(c0 int, c1 CLOB(10K)) SELECT c1 FROM myTable;</pre> <pre>[C#] cmd.CommandText = "SELECT c1 from myTable"; IDataReader reader = cmd.ExecuteReader(CommandBehavior.SequentialAccess); Int32 iChunkSize = 10; Int32 iBufferSize = 10; Int32 iFieldOffset = 0; Char[] buffer = new Char[iBufferSize]; reader.Read(); reader.GetChars(0, iFieldOffset, buffer, 0, iChunkSize);</pre> <p>The call to GetChars() will throw the following exception:</p> <pre>"Requested range extends past the end of the array"</pre> <p>To ensure that GetChars() does not throw the exception mentioned previously, you must set the BufferSize to the size of the column, as follows:</p> <pre>Int32 iBufferSize = 10000;</pre> <p>Note that the value of 10,000 for iBufferSize corresponds to the value of 10K allocated to the CLOB column c1.</p>	All
CommandBehavior.SequentialAccess	<p>The ODBC .NET Data Provider throws the following exception when there is no more data to read when using OdbcDataReader.GetChars():</p> <pre>NO_DATA - no error information available at System.Data.Odbc.OdbcConnection.HandleError(HandleRef hrHandle, SQL_HANDLE hType, RETCODE retcode) at System.Data.Odbc.OdbcDataReader.GetData(Int32 i, SQL_C sqlctype, Int32 cb) at System.Data.Odbc.OdbcDataReader.GetChars(Int32 i, Int64 dataIndex, Char[] buffer, Int32 bufferSize, Int32 length)</pre>	All

Table 6. ODBC .NET Data Provider restrictions (continued)

Class or feature	Restriction description	DB2 servers affected
CommandBehavior. SequentialAccess	<p>You may not be able to use large chunk sizes, such as a value of 5000, when using <code>OdbcDataReader.GetChars()</code>. When you attempt to use a large chunk size, the ODBC .NET Data Provider will throw the following exception:</p> <p>Object reference not set to an instance of an object. at System.Runtime.InteropServices.Marshal.Copy(Int32 source, Char[] destination, Int32 startIndex, Int32 length) at System.Data.Odbc.OdbcDataReader.GetChars(Int32 i, Int64 dataIndex, Char[] buffer, Int32 bufferSize, Int32 length) at OdbcRestrict.TestGetCharsAndBufferSize(IDbConnection con)</p>	All
Connection pooling	<p>The ODBC .NET Data Provider does not control connection pooling. Connection pooling is handled by the ODBC Driver Manager. For more information about connection pooling, see the ODBC Programmer's Reference in the MSDN library located at http://msdn.microsoft.com/library</p>	All
DataColumnMapping	<p>The case of the source column name needs to match the case used in the system catalog tables, which is upper-case by default.</p>	All
Decimal columns	<p>Parameter markers are not supported for Decimal columns.</p> <p>You generally use <code>OdbcType.Decimal</code> for an <code>OdbcParameter</code> if the target <code>SQLType</code> is a Decimal column; however, when the ODBC .NET Data Provider sees the <code>OdbcType.Decimal</code>, it binds the parameter using C-type of <code>SQL_C_WCHAR</code> and <code>SQLType</code> of <code>SQL_VARCHAR</code>, which is invalid.</p> <p>For example:</p> <pre>[C#] cmd.CommandText = "SELECT dec_col FROM MYTABLE WHERE dec_col > ? "; OdbcParameter p1 = cmd.CreateParameter(); p1.DbType = DbType.Decimal; p1.Value = 10.0; cmd.Parameters.Add(p1); IDataReader rdr = cmd.ExecuteReader();</pre> <p>You will get an exception:</p> <pre>ERROR [07006] [IBM][CLI Driver][SQLDS/VM] SQL0301N The value of input host variable or parameter number "" cannot be used because of its data type. SQLSTATE=07006</pre> <p>Workaround: Instead of using <code>OdbcParameter</code> values, use literals exclusively.</p>	DB2 for VM/VSE
Key information	<p>The schema name used to qualify the table name (for example, <code>MYSHEMA.MYTABLE</code>) must match the connection user ID. The ODBC .NET Data Provider is unable to retrieve any key information in which the specified schema is different from the connection user id.</p> <p>For example:</p> <pre>CREATE TABLE USERID2.TABLE1(c1 INT NOT NULL PRIMARY KEY);</pre> <pre>[C#] // Connect as user bob odbcCon = new OdbcConnection("DSN=sample;UID=bob;PWD=mypassword"); OdbcCommand cmd = odbcCon.CreateCommand(); // Select from table with schema USERID2 cmd.CommandText="SELECT * FROM USERID2.TABLE1"; // Fails - No key info retrieved da.FillSchema(ds, SchemaType.Source); // Fails - SchemaTable has no primary key cmd.ExecuteReader(CommandBehavior.KeyInfo) // Throws exception because no primary key cbuilder.GetUpdateCommand();</pre>	All

Table 6. ODBC .NET Data Provider restrictions (continued)

Class or feature	Restriction description	DB2 servers affected
Key information	<p>The ODBC .NET Data Provider cannot retrieve key information when opening a <code>IDataReader</code> at the same time. When the ODBC .NET Data Provider opens a <code>IDataReader</code>, a cursor on the server is opened. If key information is requested, it will then call <code>SQLPrimaryKeys()</code> or <code>SQLStatistic()</code> to get the key information, but these schema functions will open another cursor. Since DB2 for VM/VSE does not support cursor withhold, the first cursor is then closed. Consequently, <code>IDataReader.Read()</code> calls to the <code>IDataReader</code> will result in the following exception:</p> <pre>System.Data.Odbc.OdbcException: ERROR [HY010] [IBM] [CLI Driver] CLI0125E Function sequence error. SQLSTATE=HY010</pre> <p>Workaround: You will need to retrieve key information first then retrieve the data. For example:</p> <pre>[C#] OdbcCommand cmd = odbcCon.CreateCommand(); OdbcDataAdapter da = new OdbcDataAdapter(cmd); cmd.CommandText = "SELECT * FROM MYTABLE"; // Use FillSchema to retrieve just the schema information da.FillSchema(ds, SchemaType.Source); // Use FillSchema to retrieve just the schema information da.Fill(ds);</pre>	DB2 for VM/VSE
Key information	<p>You must refer to database objects in your SQL statements using the same case that the database objects are stored in the system catalog tables. By default database objects are stored in uppercase in the system catalog tables, so most often, you need to use uppercase.</p> <p>The ODBC .NET Data Provider scans SQL statements to retrieve database object names and passes them to schema functions such as <code>SQLPrimaryKeys</code> and <code>SQLStatistics</code>, which issue queries for these objects in the system catalog tables. The database object references must match exactly how they are stored in the system catalog tables, otherwise, an empty result set is returned.</p>	DB2 for OS/390 DB2 for OS/400 DB2 for VM/VSE
Key information for batched non-select SQL statements	<p>The ODBC .NET Data Provider is unable to retrieve any key information for a batch statement that does not start with "SELECT".</p>	DB2 for OS/390 DB2 for OS/400 DB2 for VM/VSE

Table 6. ODBC .NET Data Provider restrictions (continued)

Class or feature	Restriction description	DB2 servers affected
LOB columns	<p>The ODBC .NET Data Provider does not support LOB datatypes. Consequently, whenever the DB2 server returns a SQL_CLOB (-99), SQL_BLOB (-98) or SQL_DBCLOB (-350) the ODBC .NET Data Provider will throw the following exception:</p> <pre>"Unknown SQL type - -98" (for Blob column) "Unknown SQL type - -99" (for Clob column) "Unknown SQL type - -350" (for DbClob column)</pre> <p>Any methods that directly or indirectly access LOB columns will fail.</p> <p>Workaround: Set the CLI/ODBC LongDataCompat keyword to 1. Doing so will force the CLI driver to make the following data type mappings to data types the ODBC .NET Data Provider will understand:</p> <ul style="list-style-type: none"> • SQL_CLOB to SQL_LONGVARCHAR • SQL_BLOB to SQL_LONGVARBINARY • SQL_DBCLOB to SQL_WLONGVARCHAR <p>To set the LongDataCompat keyword, run the following DB2 command from a DB2 command window on the client machine:</p> <pre>db2 update cli cfg for section common using longdatacompat 1</pre> <p>You can also set this keyword in your application, using the connection string as follows:</p> <pre>[C#] OdbcConnection con = new OdbcConnection("DSN=SAMPLE;UID=uid;PWD=mypwd;LONGDATACOMPAT=1;");</pre> <p>For a list of all the CLI/ODBC keywords, refer to the "UID CLI/ODBC configuration keyword" in the DB2 CLI Guide and Reference.</p>	All
OdbcCommand.Cancel	<p>Executing statements after running OdbcCommand.Cancel can lead to the following exception:</p> <pre>"ERROR [24000] [Microsoft][ODBC Driver Manager] Invalid cursor state"</pre>	All
OdbcCommandBuilder	<p>The OdbcCommandBuilder fails to generate commands against servers that do not support escape characters. When the OdbcCommandBuilder generates commands, it first makes a call to SQLGetInfo, requesting the SQL_SEARCH_PATTERN_ESCAPE attribute. If the server does not support escape characters an empty string is returned, which causes the ODBC .NET Data Provider to throw the following exception:</p> <pre>Index was outside the bounds of the array. at System.Data.Odbc.OdbcConnection.get_EscapeChar() at System.Data.Odbc.OdbcDataReader.GetTableNameFromCommandText() at System.Data.Odbc.OdbcDataReader.BuildMetaDataInfo() at System.Data.Odbc.OdbcDataReader.GetSchemaTable() at System.Data.Common.CommandBuilder.BuildCache(Boolean closeConnection) at System.Data.Odbc.OdbcCommandBuilder.GetUpdateCommand()</pre>	DB2 for OS/390, DBCS servers only; DB2 for VM/VSE, DBCS servers only

Table 6. ODBC .NET Data Provider restrictions (continued)

Class or feature	Restriction description	DB2 servers affected
OdbcCommandBuilder	<p>Case-sensitivity is important when using the OdbcCommandBuilder to automatically generate UPDATE, DELETE, and INSERT statements. By default, DB2 stores schema information (such as table names, and column names) in the system catalog tables in upper case, unless they have been explicitly created with case-sensitivity (by adding quotation marks around database objects during create-time). As such, your SQL statements must match the case that is stored in the catalogs (which by default is uppercase).</p> <p>For example, if you created a table using the following statement: "db2 create table mytable (c1 int) "</p> <p>then DB2 will store the table name "mytable" in the system catalog tables as "MYTABLE".</p> <p>The following code example demonstrates proper use the OdbcCommandBuilder class:</p> <pre>[C#] OdbcCommand cmd = odbcCon.CreateCommand(); cmd.CommandText = "SELECT * FROM MYTABLE"; OdbcDataAdapter da = new OdbcDataAdapter(cmd); OdbcCommandBuilder cb = new OdbcCommandBuilder(da); OdbcCommand updateCmd = cb.GetUpdateCommand();</pre> <p>In this example, if you do not refer to the table name in upper-case characters, then you will get the following exception: "Dynamic SQL generation for the UpdateCommand is not supported against a SelectCommand that does not return any key column information."</p>	All
OdbcCommandBuilder	<p>The commands generated by the OdbcCommandBuilder are incorrect when the SELECT statement contains the following column data types: REAL FLOAT or DOUBLE TIMESTAMP</p> <p>These data types cannot be used in the WHERE clause for SELECT statements.</p>	DB2 for OS/390 DB2 for OS/400 DB2 for VM/VSE
OdbcCommandBuilder. DeriveParameters	<p>The DeriveParameters() method is mapped to SQLProcedureColumns and it uses the CommandText property for the name of the stored procedure. Since CommandText does not contain the name of the stored procedure (using full ODBC call syntax), SQLProcedureColumns is called with the procedure name identified according to the ODBC call syntax. For example: "{ CALL myProc(?) }"</p> <p>This which will result in an empty result set, where no columns are found for the procedure).</p>	All
OdbcCommandBuilder. DeriveParameters	<p>To use DeriveParameters(), specify the stored procedure name in the CommandText (for example, cmd.CommandText = "MYPROC"). The procedure name must match the case stored in the system catalog tables. DeriveParameters() will return all the parameters for that procedure name it finds in the system catalog tables. Remember to change the CommandText back to the full ODBC call syntax before executing the statement.</p>	All
OdbcCommandBuilder. DeriveParameters	<p>The ReturnValue parameter is not returned for the ODBC .NET Data Provider.</p>	All
OdbcCommandBuilder. DeriveParameters	<p>DeriveParameters() does not support fully qualified stored procedure names. For example, calling DeriveParameters() for CommandText = "MYSCHEMA.MYPROC" will fail. Here, no parameters are returned.</p>	All
OdbcCommandBuilder. DeriveParameters	<p>DeriveParameters() will not work for overloaded stored procedures. The SQLProcedureColumns will return all the parameters for all versions of the stored procedure.</p>	All
OdbcConnection. ChangeDatabase	<p>The OdbcConnection.ChangeDatabase() method is not supported.</p>	All

Table 6. ODBC .NET Data Provider restrictions (continued)

Class or feature	Restriction description	DB2 servers affected
OdbcConnection. ConnectionString	<ul style="list-style-type: none"> The Server keyword is ignored. The Connect Timeout keyword is ignored. CLI does not support connection timeouts, so setting this property will not affect the driver. Connection pooling keywords are ignored. Specifically, this affects the following keywords: Pooling, Min Pool Size, Max Pool Size, Connection Lifetime and Connection Reset. 	All
OdbcDataReader. GetSchemaTable	<p>The ODBC .NET Data Provider is not able to retrieve extended describe information from servers that do not return extended describe information. Therefore, if you are connecting to a server that does not support extended describe (the affected servers), the following columns in the metadata table returned from IDataReader.GetSchemaTable() are invalid:</p> <ul style="list-style-type: none"> IsReadOnly IsUnique IsAutoIncrement BaseSchemaName BaseCatalogName 	DB2 for OS/390, version 7 or lower DB2 for OS/400 DB2 for VM/VSE
Stored procedures	<p>To call a stored procedure, you need to specify the full ODBC call syntax.</p> <p>For example, to call the stored procedure, MYPROC, that takes a VARCHAR(10) as a parameter:</p> <pre>[C#] OdbcCommand cmd = odbcCon.CreateCommand(); cmd.CommandType = CommandType.Text; cmd.CommandText = "{ CALL MYPROC(?) }" OdbcParameter p1 = cmd.CreateParameter(); p1.Value = "Joe"; p1.OdbcType = OdbcType.NVarChar; cmd.Parameters.Add(p1); cmd.ExecuteNonQuery();</pre> <p>Note: Note that you must use the full ODBC call syntax even if you are using CommandType.StoredProcedure. This is documented in MSDN, under the OdbcCommand.CommandText Property.</p>	All
Stored procedures: no column names for result sets	<p>The DB2 for OS/390 version 6.1 server does not return column names for result sets returned from a stored procedure. The ODBC .NET Data Provider maps these unnamed columns to their ordinal position (for example, "1", "2" "3"). This is contrary to the mapping documented in MSDN: "Column1", "Column2", "Column3".</p>	DB2 for OS/390 version 6.1
Unique index promotion to primary key	<p>The ODBC .NET Data Provider promotes nullable unique indexes to primary keys. This is contrary to the MSDN documentation, which states that nullable unique indexes should not be promoted to primary keys.</p>	All

Chapter 2. IBM OLE DB Provider for DB2

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

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

The IBM OLE DB Provider for DB2, whose provider name is IBMDADB2, enables OLE DB consumers to access data on a DB2 database server. If DB2 Connect is installed, these OLE DB consumers can also access data on a host DBMS such as DB2 for z/OS, DB2 Server for VM and VSE, or DB2 Universal Database for AS/400.

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

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

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

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

Version Compliance

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

System Requirements

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

To install the IBM OLE DB Provider for DB2, you must first be running on one of the supported operating systems listed previously. You also need to install a full DB2 product, IBM Data Server Driver for ODBC and CLI, or IBM Data Server Driver Package.

Application Types Supported by the IBM OLE DB Provider for DB2

With the IBM OLE DB Provider for DB2 you can create either Microsoft Visual Studio C++ or Visual Basics applications.

With the IBM OLE DB Provider for DB2, you can create the following types of applications:

- ADO applications, including:
 - Microsoft Visual Studio C++ applications
 - Microsoft Visual Basic applications
- ADO.NET applications using the OLE DB .NET Data Provider
- C/C++ applications which access IBMDADB2 directly using the OLE DB interfaces, including ATL applications whose Data Access Consumer Objects were generated by the ATL COM AppWizard.

OLE DB services

Thread model supported by the IBM OLE DB Provider

The IBM OLE DB Provider for DB2 supports the Free Threaded model. This allows applications to create components in one thread and use these components in any other thread.

The IBM OLE DB Provider for DB2 supports the Free Threaded model. This allows applications to create components in one thread and use these components in any other thread.

Large object manipulation with the IBM OLE DB Provider

To get and set data as storage objects (DBTYPE_IUNKNOWN) with the IBMDADB2 provider, use the ISequentialStream interface.

To get and set data as storage objects (DBTYPE_IUNKNOWN) with the IBMDADB2 provider, use the ISequentialStream interface as follows:

- To bind a storage object to a parameter, the DBOBJECT in the DBBINDING structure can only contain the value STGM_READ for the dwFlag field. IBMDADB2 will execute the Read method of the ISequentialStream interface of the bound object.
- To get data from a storage object, your application must run the Read method on the ISequentialStream interface of the storage object.
- When getting data, the value of the length part is the length of the real data, not the length of the IUnknown pointer.

Schema rowsets supported by the IBM OLE DB Provider

The following table shows the schema rowsets that are supported by IDBSchemaRowset.

The following table shows the schema rowsets that are supported by IDBSchemaRowset. Unsupported columns will be set to null in the rowsets.

Table 7. Schema Rowsets Supported by the IBM OLE DB Provider for DB2

Supported GUIDs	Supported Restrictions	Supported Columns	Notes
DBSCHEMA _COLUMN_PRIVILEGES	COLUMN_NAME TABLE_NAME TABLE_SCHEMA	COLUMN_NAME GRANTEE GRANTOR IS_GRANTABLE PRIVILEGE_TYPE TABLE_NAME TABLE_SCHEMA	
DBSCHEMA_COLUMNS	COLUMN_NAME TABLE_NAME TABLE_SCHEMA	CHARACTER_MAXIMUM_LENGTH CHARACTER_OCTET_LENGTH COLUMN_DEFAULT COLUMN_FLAGS COLUMN_HASDEFAULT COLUMN_NAME DATA_TYPE DESCRIPTION IS_NULLABLE NUMERIC_PRECISION NUMERIC_SCALE ORDINAL_POSITION TABLE_NAME TABLE_SCHEMA	
DBSCHEMA_FOREIGN_KEYS	FK_TABLE_NAME FK_TABLE_SCHEMA PK_TABLE_NAME PK_TABLE_SCHEMA	DEFERRABILITY DELETE_RULE FK_COLUMN_NAME FK_NAME FK_TABLE_NAME FK_TABLE_SCHEMA ORDINAL PK_COLUMN_NAME PK_NAME PK_TABLE_NAME PK_TABLE_SCHEMA UPDATE_RULE	Must specify at least one of the following restrictions: PK_TABLE_NAME or FK_TABLE_NAME No "%" wildcard allowed.
DBSCHEMA_INDEXES	TABLE_NAME TABLE_SCHEMA	CARDINALITY CLUSTERED COLLATION COLUMN_NAME INDEX_NAME INDEX_SCHEMA ORDINAL_POSITION PAGES TABLE_NAME TABLE_SCHEMA TYPE UNIQUE	No sort order supported. Sort order, if specified, will be ignored.
DBSCHEMA_PRIMARY_KEYS	TABLE_NAME TABLE_SCHEMA	COLUMN_NAME ORDINAL PK_NAME TABLE_NAME TABLE_SCHEMA	Must specify at least the following restrictions: TABLE_NAME No "%" wildcard allowed.

Table 7. Schema Rowsets Supported by the IBM OLE DB Provider for DB2 (continued)

Supported GUIDs	Supported Restrictions	Supported Columns	Notes
DBSCHEMA _PROCEDURE_PARAMETERS	PARAMETER_NAME PROCEDURE_NAME PROCEDURE_SCHEMA	CHARACTER_MAXIMUM_LENGTH CHARACTER_OCTET_LENGTH DATA_TYPE DESCRIPTION IS_NULLABLE NUMERIC_PRECISION NUMERIC_SCALE ORDINAL_POSITION PARAMETER_DEFAULT PARAMETER_HASDEFAULT PARAMETER_NAME PARAMETER_TYPE PROCEDURE_NAME PROCEDURE_SCHEMA TYPE_NAME	
DBSCHEMA_PROCEDURES	PROCEDURE_NAME PROCEDURE_SCHEMA	DESCRIPTION PROCEDURE_NAME PROCEDURE_SCHEMA PROCEDURE_TYPE	
DBSCHEMA_PROVIDER_TYPES	DATA_TYPE BEST_MATCH	AUTO_UNIQUE_VALUE BEST_MATCH CASE_SENSITIVE CREATE_PARAMS COLUMN_SIZE DATA_TYPE FIXED_PREC_SCALE IS_FIXEDLENGTH IS_LONG IS_NULLABLE LITERAL_PREFIX LITERAL_SUFFIX LOCAL_TYPE_NAME MINIMUM_SCALE MAXIMUM_SCALE SEARCHABLE TYPE_NAME UNSIGNED_ATTRIBUTE	
DBSCHEMA_STATISTICS	TABLE_NAME TABLE_SCHEMA	CARDINALITY TABLE_NAME TABLE_SCHEMA	No sort order supported. Sort order, if specified, will be ignored.
DBSCHEMA _TABLE_PRIVILEGES	TABLE_NAME TABLE_SCHEMA	GRANTEE GRANTOR IS_GRANTABLE PRIVILEGE_TYPE TABLE_NAME TABLE_SCHEMA	
DBSCHEMA_TABLES	TABLE_NAME TABLE_SCHEMA TABLE_TYPE	DESCRIPTION TABLE_NAME TABLE_SCHEMA TABLE_TYPE	

OLE DB services automatically enabled by the IBM OLE DB Provider

By default, the IBM OLE DB Provider for DB2 automatically enables all the OLE DB services by adding a registry entry `OLEDB_SERVICES` under the class ID (CLSID) of the provider with the DWORD value of `0xFFFFFFFF`.

By default, the IBM OLE DB Provider for DB2 automatically enables all the OLE DB services by adding a registry entry `OLEDB_SERVICES` under the class ID (CLSID) of the provider with the DWORD value of `0xFFFFFFFF`. The meaning of this value is as follows:

Table 8. OLE DB Services

Enabled Services	DWORD Value
All services (default)	0xFFFFFFFF
All except pooling and AutoEnlistment	0xFFFFFFFFC
All except client cursor	0xFFFFFFFFB
All except pooling, enlistment and cursor	0xFFFFFFFF8
No services	0x00000000

Data services

Supported cursor modes for the IBM OLE DB Provider

The IBM OLE DB Provider for DB2 natively supports read-only, forward-only, updatable scrollable, and updatable scrollable cursors.

The IBM OLE DB Provider for DB2 natively supports read-only, forward-only, updatable scrollable, and updatable scrollable cursors.

Data type mappings between DB2 and OLE DB

The IBM OLE DB Provider for DB2 supports data type mappings between DB2 data types and OLE DB data types.

The following table provides a complete list of supported mappings and available names for indicating the data types of columns and parameters.

Table 9. Data type mappings between DB2 data types and OLE DB data types

DB2 Data Types	OLE DB Data Types Indicators	OLE DB Standard Type Names	DB2 Specific Names
SMALLINT	DBTYPE_I2	"DBTYPE_I2"	"SMALLINT"
INTEGER	DBTYPE_I4	"DBTYPE_I4"	"INTEGER" or "INT"
BIGINT	DBTYPE_I8	"DBTYPE_I8"	"BIGINT"
REAL	DBTYPE_R4	"DBTYPE_R4"	"REAL"
FLOAT	DBTYPE_R8	"DBTYPE_R8"	"FLOAT"
DOUBLE	DBTYPE_R8	"DBTYPE_R8"	"DOUBLE" or "DOUBLE PRECISION"
DECIMAL	DBTYPE_NUMERIC	"DBTYPE_NUMERIC"	"DEC" or "DECIMAL"

Table 9. Data type mappings between DB2 data types and OLE DB data types (continued)

DB2 Data Types	OLE DB Data Types Indicators	OLE DB Standard Type Names	DB2 Specific Names
NUMERIC	DBTYPE_NUMERIC	"DBTYPE_NUMERIC"	"NUM" or "NUMERIC"
DATE	DBTYPE_DBDATE	"DBTYPE_DBDATE"	"DATE"
TIME	DBTYPE_DBTIME	"DBTYPE_DBTIME"	"TIME"
TIMESTAMP	DBTYPE_DBTIMESTAMP	"DBTYPE_DBTIMESTAMP"	"TIMESTAMP"
CHAR	DBTYPE_STR	"DBTYPE_CHAR"	"CHAR" or "CHARACTER"
VARCHAR	DBTYPE_STR	"DBTYPE_VARCHAR"	"VARCHAR"
LONG VARCHAR	DBTYPE_STR	"DBTYPE_LONGVARCHAR"	"LONG VARCHAR"
CLOB	DBTYPE_STR and DBCOLUMNFLAGS_ISLONG or DBPARAMFLAGS_ISLONG	"DBTYPE_CHAR" "DBTYPE_VARCHAR" "DBTYPE_LONGVARCHAR" and DBCOLUMNFLAGS_ISLONG or DBPARAMFLAGS_ISLONG	"CLOB"
GRAPHIC	DBTYPE_WSTR	"DBTYPE_WCHAR"	"GRAPHIC"
VARGRAPHIC	DBTYPE_WSTR	"DBTYPE_WVARCHAR"	"VARGRAPHIC"
LONG VARGRAPHIC	DBTYPE_WSTR	"DBTYPE_WLONGVARCHAR"	"LONG VARGRAPHIC"
DBCLOB	DBTYPE_WSTR and DBCOLUMNFLAGS_ISLONG or DBPARAMFLAGS_ISLONG	"DBTYPE_WCHAR" "DBTYPE_WVARCHAR" "DBTYPE_WLONGVARCHAR" and DBCOLUMNFLAGS_ISLONG or DBPARAMFLAGS_ISLONG	"DBCLOB"
CHAR(n) FOR BIT DATA	DBTYPE_BYTES	"DBTYPE_BINARY"	
VARCHAR(n) FOR BIT DATA	DBTYPE_BYTES	"DBTYPE_VARBINARY"	
LONG VARCHAR FOR BIT DATA	DBTYPE_BYTES	"DBTYPE_LONGVARBINARY"	
BLOB	DBTYPE_BYTES and DBCOLUMNFLAGS_ISLONG or DBPARAMFLAGS_ISLONG	"DBTYPE_BINARY" "DBTYPE_VARBINARY" "DBTYPE_LONGVARBINARY" and DBCOLUMNFLAGS_ISLONG or DBPARAMFLAGS_ISLONG	"BLOB"

Data conversion for setting data from OLE DB Types to DB2 Types

The IBM OLE DB Provider for DB2 supports data conversions for setting data from OLE DB types to DB2 types.

Supported data conversion from OLE DB Types to DB2 Types

The following table shows data conversions from OLE DB types to DB2 types. Note that truncation of the data may occur in some cases, depending on the types and the value of the data.

Table 10. Data conversions from OLE DB types to DB2 types

OLE DB Type Indicator	DB2 Data Types																						
	S M A L L I N T	I N T E G E R	B I G I N T	R E A L	F L O A T	D E C I M A L N U M B E R	D A T E	T I M E	T I M E S T A M P	C H A R	V A R C H A R	L O N G V A R C H A R	C L O B	G R A P H I C	V A R G R A P H I C	L O N G V A R G R A P H I C	D B C L O B	For Bit Data		B L O B	D A T A L I N K		
																		C H A R	V A R C H A R				
DBTYPE_EMPTY																							
DBTYPE_NULL																							
DBTYPE_RESERVED																							
DBTYPE_I1	X	X	X	X	X	X				X	X												
DBTYPE_I2	X	X	X	X	X	X				X	X												
DBTYPE_I4	X	X	X	X	X	X				X	X												
DBTYPE_I8	X	X	X	X	X	X				X	X												
DBTYPE_UI1	X	X	X	X	X	X				X	X												
DBTYPE_UI2	X	X	X	X	X	X				X	X												
DBTYPE_UI4	X	X	X	X	X	X				X	X												
DBTYPE_UI8	X	X	X	X	X	X				X	X												
DBTYPE_R4	X	X	X	X	X	X				X	X												
DBTYPE_R8	X	X	X	X	X	X				X	X												
DBTYPE_CY																							

Table 10. Data conversions from OLE DB types to DB2 types (continued)

OLE DB Type Indicator	DB2 Data Types																				
	S M A L L I N T	I N T E G E R	B I T	R E A L	F L O A T I N G	D E C I M A L N U M E R I C	D A T E	T I M E	T I M E S T A M P	C H A R	V A R C H A R	L O N G V A R C H A R	C L O B	G R A P H I C	V A R G R A P H I C	L O N G V A R G R A P H I C	For Bit Data			D A T A L I N K	
																	D B C L O B	C H A R	V A R C H A R		
DBTYPE_DECIMAL	X	X	X	X	X	X				X	X										
DBTYPE_NUMERIC	X	X	X	X	X	X				X	X										
DBTYPE_DATE																					
DBTYPE_BOOL	X	X	X	X	X	X				X	X										
DBTYPE_BYTES			X			X				X	X	X			X		X	X	X		
DBTYPE_BSTR - to be determined																					
DBTYPE_STR	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		X	X	X		X
DBTYPE_WSTR													X	X	X						
DBTYPE_VARIANT - to be determined																					
DBTYPE_IDISPATCH																					
DBTYPE_IUNKNOWN										X	X	X	X	X	X	X	X	X	X	X	
DBTYPE_GUID																					
DBTYPE_ERROR																					
DBTYPE_BYREF																					
DBTYPE_ARRAY																					

Table 10. Data conversions from OLE DB types to DB2 types (continued)

OLE DB Type Indicator	DB2 Data Types																				
	S M A L L I N T	I N T E G E R	B I T	R E A L	F L O A T I N G	D E C I M A L N U M E R I C	D A T E	T I M E	T I M E S T A M P	C H A P T E R	V A R C H A R	V A R C H A R C H A R	C L O B	G R A P H I C	V A R G R A P H I C	L O N G V A R R A Y	For Bit Data			D A T A L I N K	
																	C H A R	V A R C H A R	L O N G V A R C H A R		
DBTYPE_VECTOR																					
DBTYPE_UDT																					
DBTYPE_DBDATE						X		X	X	X											
DBTYPE_DBTIME							X	X	X	X											
DBTYPE_DBTIMESTAMP						X	X	X	X	X											
DBTYPE_FILETIME																					
DBTYPE_PROP_VARIANT																					
DBTYPE_HCHAPTER																					
DBTYPE_VARNUMERIC																					

Data conversion for setting data from DB2 types to OLE DB types

For getting data, the IBM OLE DB Provider allows data conversions from DB2 types to OLE DB types.

Supported data conversions from DB2 types to OLE DB types

The following table shows supported data conversions from DB2 types to OLE DB types. Note that truncation of the data may occur in some cases, depending on the types and the value of the data.

Table 11. Data conversions from DB2 types to OLE DB types

OLE DB Type Indicator	DB2 Data Types																					
	S M A L L I N T	I N T E G E R	B I G I N T	R E A L	F L O A T	D E C I M A L	D A T E	T I M E	T I M E S T A M P	C H A R	V A R C H A R	L O N G V A R C H A R	C L O B	G R A P H I C	V A R G R A P H I C	L O N G V A R G R A P H I C	For Bit Data			D A T A L I N K		
																	D B C L O B	C H A R	V A R C H A R		L O N G V A R C H A R	
DBTYPE_EMPTY																						
DBTYPE_NULL																						
DBTYPE_RESERVED																						
DBTYPE_I1	X	X		X	X	X				X	X	X		X	X	X		X	X	X		X
DBTYPE_I2	X	X		X	X	X				X	X	X		X	X	X		X	X	X		X
DBTYPE_I4	X	X		X	X	X				X	X	X		X	X	X		X	X	X		X
DBTYPE_I8	X	X	X	X	X	X				X	X	X		X	X	X		X	X	X		X
DBTYPE_UI1	X	X		X	X	X				X	X	X		X	X	X		X	X	X		X
DBTYPE_UI2	X	X		X	X	X				X	X	X		X	X	X		X	X	X		X
DBTYPE_UI4	X	X		X	X	X				X	X	X		X	X	X		X	X	X		X
DBTYPE_UI8	X	X	X	X	X	X				X	X	X		X	X	X		X	X	X		X
DBTYPE_R4	X	X		X	X	X				X	X	X		X	X	X		X	X	X		X
DBTYPE_R8	X	X		X	X	X				X	X	X		X	X	X		X	X	X		X
DBTYPE_CY	X	X		X	X	X				X	X	X		X	X	X		X	X	X		X
DBTYPE_DECIMAL	X	X		X	X	X				X	X	X		X	X	X		X	X	X		X
DBTYPE_NUMERIC	X	X		X	X	X				X	X	X		X	X	X		X	X	X		X

Table 11. Data conversions from DB2 types to OLE DB types (continued)

OLE DB Type Indicator	DB2 Data Types																					
	S M A L L I N T	I N T E G E R	B I G I N T	R E A L	F L O A T I N G	D E C I M A L N U M E R I C	D A T E	T I M E	T I M E S T A M P	C H A R	V A R C H A R	L O N G V A R C H A R	C L O B	G R A P H I C	V A R G R A P H I C	L O N G V A R G R A P H I C	For Bit Data			D A T A L I N K		
																	C H A R	V A R C H A R	L O N G V A R C H A R			
DBTYPE_DATE	X	X		X	X		X	X	X	X	X	X		X	X	X						X
DBTYPE_BOOL	X	X		X	X	X				X	X	X		X	X	X		X	X	X		X
DBTYPE_BYTES	X	X		X	X	X	X	X	X	X	X	X		X	X	X		X	X	X		X
DBTYPE_BSTR	X	X	X	X	X	X	X	X	X	X	X	X		X	X	X		X	X	X		X
DBTYPE_STR	X	X	X	X	X	X	X	X	X	X	X	X		X	X	X		X	X	X		X
DBTYPE_WSTR	X	X	X	X	X	X	X	X	X	X	X	X		X	X	X		X	X	X		X
DBTYPE_VARIANT	X	X	X	X	X	X	X	X	X	X	X	X		X	X	X		X	X	X		X
DBTYPE_IDISPATCH																						
DBTYPE_IUNKNOWN	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
DBTYPE_GUID										X	X	X		X	X	X		X	X	X		X
DBTYPE_ERROR																						
DBTYPE_BYREF																						
DBTYPE_ARRAY																						
DBTYPE_VECTOR																						
DBTYPE_UDT																						
DBTYPE_DBDATE							X	X	X	X	X	X		X	X	X		X	X	X		X

Table 11. Data conversions from DB2 types to OLE DB types (continued)

OLE DB Type Indicator	DB2 Data Types																For Bit Data			DATA LINK		
	SMALLINT	INTEGER	BIGINT	REAL	FLOAT	DECIMAL	DATE	TIME	TIMESTAMP	CHAR	VARCHAR	LONG VARCHAR	CLOB	GRAPHIC	VARGRAPHIC	DBCLOB	CHAR	VARCHAR	LONG VARCHAR			
																					CHAR	VARCHAR
DBTYPE_DBTIME							X	X	X	X	X			X	X	X					X	
DBTYPE_DBTIMESTAMP							X	X	X	X	X			X	X	X		X	X	X		X
DBTYPE_FILETIME			X				X	X	X	X	X			X	X	X		X	X	X		X
DBTYPE_PROP_VARIANT	X	X	X	X	X					X	X	X		X	X	X		X	X	X		X
DBTYPE_HCHAPTER																						
DBTYPE_VARNUMERIC																						

Note: When the application performs the ISequentialStream::Read to get the data from the storage object, the format of the data returned depends on the column data type:

- For non character and binary data types, the data of the column is exposed as a sequence of bytes which represent those values in the operating system.
- For character data type, the data is first converted to DBTYPE_STR.
- For DBCLOB, the data is first converted to DBTYPE_WCHAR.

IBM OLE DB Provider restrictions

The restrictions for the IBM OLE DB Provider are IBMDADB2 supports auto commit and user-controlled transaction scope, RestartPosition is not supported and IBMDADB2 does not quote table names passed through the DBID parameters.

The restrictions for the IBM OLE DB Provider are:

- IBMDADB2 supports auto commit and user-controlled transaction scope with the ITransactionLocal interface. Auto commit transaction scope is the default scope. Nested transactions are not supported.
- RestartPosition is not supported when the command text contains parameters.
- IBMDADB2 does not quote table names passed through the DBID parameters, which are parameters used by the IOpenRowset interface. Instead, the OLE DB consumer must add quotation marks to the table names when quotes are required.

IBM OLE DB Provider support for OLE DB components and interfaces

The following tables list the OLE DB components and interfaces that are supported by the IBM OLE DB Provider for DB2 and the Microsoft OLE DB Provider for ODBC.

The following tables list the OLE DB components and interfaces that are supported by the IBM OLE DB Provider for DB2 and the Microsoft OLE DB Provider for ODBC.

Table 12. Blobs

Interface	DB2	ODBC Provider
ISequentialStream	Yes	Yes

Table 13. Commands

Interface	DB2	ODBC Provider
IAccessor	Yes	Yes
ICommand	Yes	Yes
ICommandPersist	No	No
ICommandPrepare	Yes	Yes
ICommandProperties	Yes	Yes
ICommandText	Yes	Yes
ICommandWithParameters	Yes	Yes
IColumnsInfo	Yes	Yes
IColumnsRowset	Yes	Yes
IConvertType	Yes	Yes
ISupportErrorInfo	Yes	Yes

Table 14. DataSources

Interface	DB2	ODBC Provider
IConnectionPoint	No	Yes
IDBAsynchNotify (consumer)	No	No
IDBAsynchStatus	No	No
IDBConnectionPointContainer	No	Yes
IDBCreateSession	Yes	Yes
IDBDataSourceAdmin	No	No
IDBInfo	Yes	Yes
IDBInitialize	Yes	Yes
IDBProperties	Yes	Yes
IPersist	Yes	No
IPersistFile	Yes	Yes
ISupportErrorInfo	Yes	Yes

Table 15. Enumerators

Interface	DB2	ODBC Provider
IDBInitialize	Yes	Yes
IDBProperties	Yes	Yes
IParseDisplayName	Yes	No
ISourcesRowset	Yes	Yes
ISupportErrorInfo	Yes	Yes

Table 16. Error Lookup Service

Interface	DB2	ODBC Provider
IErrorLookup	Yes	Yes

Table 17. Error Objects

Interface	DB2	ODBC Provider
IErrorInfo	Yes	No
ISQLErrorInfo (custom)	Yes	No

Table 18. Multiple Results

Interface	DB2	ODBC Provider
IMultipleResults	Yes	Yes
ISupportErrorInfo	Yes	Yes

Table 19. Rowsets

Interface	DB2	ODBC Provider
IAccessor	Yes	Yes
IColumnsRowset	Yes	Yes
IColumnsInfo	Yes	Yes
IConvertType	Yes	Yes
IChapteredRowset	No	No
IConnectionPointContainer	Yes	Yes
IDBAsynchStatus	No	No
IParentRowset	No	No
IRowset	Yes	Yes
IRowsetChange	Yes	Yes
IRowsetChapterMember	No	No
IRowsetFind	No	No
IRowsetIdentity	Yes	Yes
IRowsetIndex	No	No
IRowsetInfo	Yes	Yes
IRowsetLocate	Yes	Yes
IRowsetNotify (consumer)	Yes	No
IRowsetRefresh	Cursor Service Component	Yes

Table 19. Rowsets (continued)

Interface	DB2	ODBC Provider
IRowsetResynch	Cursor Service Component	Yes
IRowsetScroll	Yes ¹	Yes
IRowsetUpdate	Cursor Service Component	Yes
IRowsetView	No	No
ISupportErrorInfo	Yes	Yes
Note:		
1. The values to be returned are approximations. Deleted rows will not be skipped.		

Table 20. Sessions

Interface	DB2	ODBC Provider
IAlterIndex	No	No
IAlterTable	No	No
IDBCreateCommand	Yes	Yes
IDBSchemaRowset	Yes	Yes
IGetDataSource	Yes	Yes
IIndexDefinition	No	No
IOpenRowset	Yes	Yes
ISessionProperties	Yes	Yes
ISupportErrorInfo	Yes	Yes
ITableDefinition	No	No
ITableDefinitionWithConstraints	No	No
ITransaction	Yes	Yes
ITransactionJoin	Yes	Yes
ITransactionLocal	Yes	Yes
ITransactionObject	No	No
ITransactionOptions	No	Yes

Table 21. View Objects

Interface	DB2	ODBC Provider
IViewChapter	No	No
IViewFilter	No	No
IViewRowset	No	No
IViewSort	No	No

IBM OLE DB Provider support for OLE DB properties

The following table shows the OLE DB properties that are supported by the IBM OLE DB Provider for DB2 and default value.

The following table shows the OLE DB properties that are supported by the IBM OLE DB Provider for DB2:

Table 22. Properties Supported by the IBM OLE DB Provider for DB2: Data Source (DBPROPSET_DATASOURCE)

Properties	Default Value	R/W
DBPROP_MULTIPLECONNECTIONS	VARIANT_FALSE	R
DBPROP_RESETDATASOURCE	DBPROPVAL_RD_RESETALL	R/W

Table 23. Properties Supported by the IBM OLE DB Provider for DB2: DB2 Data Source (DBPROPSET_DB2DATASOURCE)

Properties	Default Value	R/W
DB2PROP_REPORTISLONGFORLONGTYPES	VARIANT_FALSE	R/W
DB2PROP_RETURNCHARASWCHAR	VARIANT_TRUE	R/W
DB2PROP_SORTBYORDINAL	VARIANT_FALSE	R/W

Table 24. Properties Supported by the IBM OLE DB Provider for DB2: Data Source Information (DBPROPSET_DATASOURCEINFO)

Properties	Default Value	R/W
DBPROP_ACTIVESESSIONS	0	R
DBPROP_ASYNCCTXNABORT	VARIANT_FALSE	R
DBPROP_ASYNCCTXNCOMMIT	VARIANT_FALSE	R
DBPROP_BYREFACCESSORS	VARIANT_FALSE	R
DBPROP_COLUMNDEFINITION	DBPROPVAL_CD_NOTNULL	R
DBPROP_CONCATNULLBEHAVIOR	DBPROPVAL_CB_NULL	R
DBPROP_CONNECTIONSTATUS	DBPROPVAL_CS_INITIALIZED	R
DBPROP_DATASOURCENAME	N/A	R
DBPROP_DATASOURCEREADONLY	VARIANT_FALSE	R
DBPROP_DBMSNAME	N/A	R
DBPROP_DBMSVER	N/A	R
DBPROP_DSOTHREADMODEL	DBPROPVAL_RT_FREETHREAD	R
DBPROP_GROUPBY	DBPROPVAL_GB_CONTAINS_SELECT	R
DBPROP_IDENTIFIERCASE	DBPROPVAL_IC_UPPER	R
DBPROP_MAXINDEXSIZE	0	R
DBPROP_MAXROWSIZE	0	R
DBPROP_MAXROWSIZEINCLUDESBLOB	VARIANT_TRUE	R
DBPROP_MAXTABLEINSELECT	0	R
DBPROP_MULTIPLEPARAMSETS	VARIANT_FALSE	R
DBPROP_MULTIPLERESULTS	DBPROPVAL_MR_SUPPORTED	R
DBPROP_MULTIPLESTORAGEOBJECTS	VARIANT_TRUE	R
DBPROP_MULTITABLEUPDATE	VARIANT_FALSE	R
DBPROP_NULLCOLLATION	DBPROPVAL_NC_LOW	R
DBPROP_OLEOBJECTS	DBPROPVAL_OO_BLOB	R
DBPROP_ORDERBYCOLUMNSINSELECT	VARIANT_FALSE	R

Table 24. Properties Supported by the IBM OLE DB Provider for DB2: Data Source Information (DBPROPSET_DATASOURCEINFO) (continued)

Properties	Default Value	R/W
DBPROP_OUTPUTPARAMETERAVAILABILITY	DBPROPVAL_OA_ATEXECUTE	R
DBPROP_PERSISTENTIDTYPE	DBPROPVAL_PT_NAME	R
DBPROP_PREPAREABORTBEHAVIOR	DBPROPVAL_CB_DELETE	R
DBPROP_PROCEDURETERM	"STORED PROCEDURE"	R
DBPROP_PROVIDERFRIENDLYNAME	"IBM OLE DB Provider for DB2"	R
DBPROP_PROVIDERNAME	"IBMDADB2.DLL"	R
DBPROP_PROVIDEROLEDBVER	"02.7"	R
DBPROP_PROVIDEREVER	N/A	R
DBPROP_QUOTEIDENTIFIERCASE	DBPROPVAL_IC_SENSITIVE	R
DBPROP_ROWSETCONVERSIONSONCOMMAND	VARIANT_TRUE	R
DBPROP_SCHEMATERM	"SCHEMA"	R
DBPROP_SCHEMAUSAGE	DBPROPVAL_SU_DML_STATEMENTS DBPROPVAL_SU_TABLE_DEFINITION DBPROPVAL_SU_INDEX_DEFINITION DBPROPVAL_SU_PRIVILEGE_DEFINITION	R
DBPROP_SQLSUPPORT	DBPROPVAL_SQL_ODBC_EXTENDED DBPROPVAL_SQL_ESCAPECLAUSES DBPROPVAL_SQL_ANSI92_ENTRY	R
DBPROP_SERVERNAME	N/A	R
DBPROP_STRUCTUREDSTORAGE	DBPROPVAL_SS_ISEQUENTIALSTREAM	R
DBPROP_SUBQUERIES	DBPROPVAL_SQ_CORRELATEDSUBQUERIES DBPROPVAL_SQ_COMPARISON DBPROPVAL_SQ_EXISTS DBPROPVAL_SQ_IN DBPROPVAL_SQ_QUANTIFIED	R
DBPROP_SUPPORTEDTXNDDL	DBPROPVAL_TC_ALL	R
DBPROP_SUPPORTEDTXNISOLEVELS	DBPROPVAL_TI_CURSORSTABILITY DBPROPVAL_TI_READCOMMITTED DBPROPVAL_TI_READUNCOMMITTED DBPROPVAL_TI_SERIALIZABLE	R
DBPROP_SUPPORTEDTXNISORETAIN	DBPROPVAL_TR_COMMIT_DC DBPROPVAL_TR_ABORT_NO	R
DBPROP_TABLETERM	"TABLE"	R
DBPROP_USERNAME	N/A	R

Table 25. Properties Supported by the IBM OLE DB Provider for DB2: Initialization (DBPROPSET_DBINIT)

Properties	Default Value	R/W
DBPROP_AUTH_PASSWORD	N/A	R/W
DBPROP_INIT_TIMEOUT (1)	0	R/W
DBPROP_AUTH_PERSIST _SENSITIVE_AUTHINFO	VARIANT_FALSE	R/W
DBPROP_AUTH_USERID	N/A	R/W
DBPROP_INIT_DATASOURCE	N/A	R/W
DBPROP_INIT_HWND	N/A	R/W
DBPROP_INIT_MODE	DB_MODE_READWRITE	R/W
DBPROP_INIT_OLEDBSERVICES	0xFFFFFFFF	R/W
DBPROP_INIT_PROMPT	DBPROMPT_NOPROMPT	R/W
DBPROP_INIT_PROVIDERSTRING	N/A	R/W

Table 26. Properties Supported by the IBM OLE DB Provider for DB2: Rowset (DBPROPSET_ROWSET)

Properties	Default Value	R/W
DBPROP_ABORTPRESERVE	VARIANT_FALSE	R
DBPROP_ACCESSORDER	DBPROPVAL_AO_RANDOM	R
DBPROP_BLOCKINGSTORAGEOBJECTS	VARIANT_FALSE	R
DBPROP_BOOKMARKS	VARIANT_FALSE	R/W
DBPROP_BOOKMARKSKIPPED	VARIANT_FALSE	R
DBPROP_BOOKMARKTYPE	DBPROPVAL_BMK_NUMERIC	R
DBPROP_CACHEDEFERRED	VARIANT_FALSE	R/W
DBPROP_CANFETCHBACKWARDS	VARIANT_FALSE	R/W
DBPROP_CANHOLDROWS	VARIANT_FALSE	R
DBPROP_CANSROLLBACKWARDS	VARIANT_FALSE	R/W
DBPROP_CHANGEINSERTEDROWS	VARIANT_FALSE	R
DBPROP_COMMITPRESERVE	VARIANT_TRUE	R/W
DBPROP_COMMANDTIMEOUT	0	R/W
DBPROP_DEFERRED	VARIANT_FALSE	R
DBPROP_IAccessor	VARIANT_TRUE	R
DBPROP_IColumnsInfo	VARIANT_TRUE	R
DBPROP_IColumnsRowset	VARIANT_TRUE	R/W
DBPROP_IConvertType	VARIANT_TRUE	R
DBPROP_IMultipleResults	VARIANT_FALSE	R/W
DBPROP_IRowset	VARIANT_TRUE	R
DBPROP_IRowChange	VARIANT_FALSE	R/W
DBPROP_IRowsetFind	VARIANT_FALSE	R
DBPROP_IRowsetIdentity	VARIANT_TRUE	R
DBPROP_IRowsetInfo	VARIANT_TRUE	R
DBPROP_IRowsetLocate	VARIANT_FALSE	R/W

Table 26. Properties Supported by the IBM OLE DB Provider for DB2: Rowset (DBPROPSET_ROWSET) (continued)

Properties	Default Value	R/W
DBPROP_IRowsetScroll	VARIANT_FALSE	R/W
DBPROP_IRowsetUpdate	VARIANT_FALSE	R
DBPROP_ISequentialStream	VARIANT_TRUE	R
DBPROP_ISupportErrorInfo	VARIANT_TRUE	R
DBPROP_LITERALBOOKMARKS	VARIANT_FALSE	R
DBPROP_LITERALIDENTITY	VARIANT_TRUE	R
DBPROP_LOCKMODE	DBPROPVAL_LM_SINGLEROW	R/W
DBPROP_MAXOPENROWS	32767	R
DBPROP_MAXROWS	0	R/W
DBPROP_NOTIFICATIONGRANULARITY	DBPROPVAL_NT_SINGLEROW	R/W
DBPROP_NOTIFICATION PHASES	DBPROPVAL_NP_OKTODO DBPROPBAL_NP_ABOUTTODO DBPROPVAL_NP_SYNCHAFTER DBPROPVAL_NP_FAILEDTODO DBPROPVAL_NP_DIDEVENT	R
DBPROP_NOTIFYROWSETRELEASE	DBPROPVAL_NP_OKTODO DBPROPVAL_NP_ABOUTTODO	R
DBPROP_NOTIFYROWSETFETCHPOSITIONCHANGE	DBPROPVAL_NP_OKTODO DBPROPVAL_NP_ABOUTTODO	R
DBPROP_NOTIFYCOLUMNSET	DBPROPVAL_NP_OKTODO DBPROPVAL_NP_ABOUTTODO	R
DBPROP_NOTIFYROWDELETE	DBPROPVAL_NP_OKTODO DBPROPVAL_NP_ABOUTTODO	R
DBPROP_NOTIFYROWINSERT	DBPROPVAL_NP_OKTODO DBPROPVAL_NP_ABOUTTODO	R
DBPROP_ORDEREDBOOKMARKS	VARIANT_FALSE	R
DBPROP_OTHERINSERT	VARIANT_FALSE	R
DBPROP_OTHERUPDATEDELETE	VARIANT_FALSE	R/W
DBPROP_OWNINGINSERT	VARIANT_FALSE	R
DBPROP_OWNINGUPDATEDELETE	VARIANT_FALSE	R
DBPROP_QUICKRESTART	VARIANT_FALSE	R/W
DBPROP_REMOVEDELETED	VARIANT_FALSE	R/W
DBPROP_ROWTHREADMODEL	DBPROPVAL_RT_FREETHREAD	R
DBPROP_SERVERCURSOR	VARIANT_TRUE	R
DBPROP_SERVERDATAONINSERT	VARIANT_FALSE	R
DBPROP_UNIQUEROWS	VARIANT_FALSE	R/W
DBPROP_UPDATABILITY	0	R/W

Table 27. Properties Supported by the IBM OLE DB Provider for DB2: DB2 Rowset (DBPROPSET_DB2ROWSET)

Properties	Default Value	R/W
DBPROP_ISLONGMINLENGTH	32000	R/W

Table 28. Properties Supported by the IBM OLE DB Provider for DB2: Session (DBPROPSET_SESSION)

Properties	Default Value	R/W
DBPROP_SESS_AUTOCOMMITISOLEVELS	DBPROPVAL_TI_CURSORSTABILITY	R/W

Note:

1. The timeout is applicable only when using the TCP/IP protocol to connect to the server. The timeout is enforced only during the TCP/IP sock connect. If the sock connect completes before the specified timeout expires, the timeout will no longer be enforced for the rest of the initialization process. If the client-reroute feature is used then the timeout will be doubled. In general, when client reroute is enabled, the connection timeout behavior is dictated by client reroute.

Connections to data sources using the IBM OLE DB Provider

The following examples show how to connect to a DB2 data source using the IBM OLE DB Provider for DB2

The following examples show how to connect to a DB2 data source using the IBM OLE DB Provider for DB2.

Example 1: Visual Basic application using ADO

```
Dim db As ADODB.Connection
Set db = New ADODB.Connection
db.Provider = "IBMDADB2"
db.CursorLocation = adUseClient
...
```

Example 2: C/C++ application using IDataInitialize and Service Component

```
hr = CoCreateInstance (
    CLSID_MSDAINITIALIZE,
    NULL,
    CLSCTX_INPROC_SERVER,
    IID_IDataInitialize,
    (void**)&IDataInitialize);

hr = pIDataInitialize->CreateDBInstance(
    CLSID_IBMDADB2, // ClassID of IBMDADB2
    NULL,
    CLSCTX_INPROC_SERVER,
    NULL,
    IID_IDBInitialize,
    (IUnknown**)&pIDBInitialize);
```

ADO applications

ADO connection string keywords

To use ADO connection string keywords, you must specify the value of the keyword in `keyword=value`. And each key value pair is delimited by semicolon. Table 1 further illustrates the keywords, values and meaning.

To specify ADO (ActiveX Data Objects) connection string keywords, specify the keyword using the keyword=*value* format in the provider (connection) string. Delimit multiple keywords with a semicolon (;).

The following table describes the keywords supported by the IBM OLE DB Provider for DB2:

Table 29. Keywords supported by the IBM OLE DB Provider for DB2

Keyword	Value	Meaning
DSN	Name of the database alias	The DB2 database alias in the database directory.
UID	User ID	The user ID used to connect to the DB2 server.
PWD	Password of UID	Password for the user ID used to connect to the DB2 server.

Other CLI configuration keywords also affect the behavior of the IBM OLE DB Provider.

Connections to data sources with Visual Basic ADO applications

IBMDADB2 provider name must be specified to use the IBM OLE DB Provider for DB2.

To connect to a DB2 data source using the IBM OLE DB Provider for DB2, specify the IBMDADB2 provider name.

Updatable scrollable cursors in ADO applications

An ADO application that wants to access updatable scrollable cursors can set the cursor location to either `adUseClient` or `adUseServer`. Setting the cursor location to `adUseServer` causes the cursor to materialize on the server.

The IBM OLE DB Provider for DB2 natively supports read-only, forward-only, read-only scrollable, and updatable scrollable cursors. An ADO application that wants to access updatable scrollable cursors can set the cursor location to either `adUseClient` or `adUseServer`. Setting the cursor location to `adUseServer` causes the cursor to materialize on the server.

Limitations for ADO applications

ADO applications have limitations with calling stored procedures, inserting a new row using a server-side scrollable cursor and no support for default parameter values.

The limitations for ADO applications are:

- ADO applications calling stored procedures must have their parameters created and explicitly bound. The `Parameters.Refresh` method for automatically generating parameters is not supported for DB2 Server for VSE & VM.
- There is no support for default parameter values.
- When inserting a new row using a server-side scrollable cursor, use the `AddNew()` method with the `Fieldlist` and `Values` arguments. This is more efficient than calling `AddNew()` with no arguments following `Update()` calls for

each column. Each AddNew() and Update() call is a separate request to the server and therefore, is less efficient than a single call to AddNew().

- Newly inserted rows are not updatable with a server-side scrollable cursor.
- Tables with long or LOB data are not updatable when using a server-side scrollable cursor.

IBM OLE DB Provider support for ADO methods and properties

The IBM OLE DB Provider supports the following ADO methods and properties.

The IBM OLE DB Provider supports the following ADO methods and properties:

Table 30. Command Methods

Method/Property	OLE DB Interface/Property	IBM OLE DB Support
Cancel	ICommand	Yes
CreateParameter		Yes
Execute		Yes

Table 31. Command Properties

Method/Property	OLE DB Interface/Property	IBM OLE DB Support
ActiveConnection	(ADO specific)	
Command Text	ICommandText	Yes
Command Timeout	ICommandProperties::SetProperties DBPROP_COMMANDTIMEOUT	Yes
CommandType	(ADO specific)	
Prepared	ICommandPrepare	Yes
State	(ADO specific)	

Table 32. Command Collections

Method/Property	OLE DB Interface/Property	IBM OLE DB Support
Parameters	ICommandWithParameter DBSCHEMA _PROCEDURE_PARAMETERS	Yes
Properties	ICommandProperties IDBProperties	Yes

Table 33. Connection Methods

Method/Property	OLE DB Interface/Property	IBM OLE DB Support
BeginTrans CommitTrans RollbackTrans	ITransactionLocal	Yes (but not nested) Yes (but not nested) Yes (but not nested)
Execute	ICommand IOpenRowset	Yes

Table 33. Connection Methods (continued)

Method/Property	OLE DB Interface/Property	IBM OLE DB Support
Open	IDBCreateSession IDBInitialize	Yes
OpenSchema adSchemaColumnPrivileges adSchemaColumns adSchemaForeignKeys adSchemaIndexes adSchemaPrimaryKeys adSchemaProcedureParam adSchemaProcedures adSchemaProviderType adSchemaStatistics adSchemaTablePrivileges adSchemaTables	IDBSchemaRowset	Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes
Cancel		Yes

Table 34. Connection Properties

Method/Property	OLE DB Interface/Property	IBM OLE DB Support
Attributes adXactCommitRetaining adXactRollbackRetaining	ITransactionLocal	Yes Yes
CommandTimeout	ICommandProperties DBPROP_COMMAND_TIMEOUT	Yes
ConnectionString	(ADO specific)	
ConnectionTimeout	IDBProperties DBPROP_INIT_TIMEOUT	No
CursorLocation: adUseClient adUseNone adUseServer	(Use OLE DB Cursor Service) (Not Used)	Yes No Yes
DefaultDataBase	IDBProperties DBPROP_CURRENTCATALOG	No
IsolationLevel	ITransactionLocal DBPROP_SESS _AUTOCOMMITISOLEVELS	Yes

Table 34. Connection Properties (continued)

Method/Property	OLE DB Interface/Property	IBM OLE DB Support
Mode	IDBProperties	No
adModeRead	DBPROP_INIT_MODE	Yes
adModeReadWrite		No
adModeShareDenyNone		No
adModeShareDenyRead		No
adModeShareDenyWrite		No
adModeShareExclusive		No
adModeUnknown		No
adModeWrite		No
Provider	ISourceRowset::GetSourceRowset	Yes
State	(ADO specific)	
Version	(ADO specific)	

Table 35. Connection Collection

Method/Property	OLE DB Interface/Property	IBM OLE DB Support
Errors	IErrorRecords	Yes
Properties	IDBProperties	Yes

Table 36. Error Properties

Method/Property	OLE DB Interface/Property	IBM OLE DB Support
Description	IErrorRecords	Yes
NativeError		Yes
Number		Yes
Source		Yes
SQLState		Yes
HelpContext		No
HelpFile		No

Table 37. Field Methods

Method/Property	OLE DB Interface/Property	IBM OLE DB Support
AppendChunk	ISequentialStream	Yes
GetChunk		Yes

Table 38. Field Properties

Method/Property	OLE DB Interface/Property	IBM OLE DB Support
Actual Size	IAccessor IRowset	Yes

Table 38. Field Properties (continued)

Method/Property	OLE DB Interface/Property	IBM OLE DB Support
Attributes DataFormat DefinedSize Name NumericScale Precision Type	IColumnInfo	Yes Yes Yes Yes Yes Yes
OriginalValue	IRowsetUpdate	Yes (Cursor Service)
UnderlyingValue	IRowsetRefresh IRowsetResynch	Yes (Cursor Service) Yes (Cursor Service)
Value	IAccessor IRowset	Yes

Table 39. Field Collection

Method/Property	OLE DB Interface/Property	IBM OLE DB Support
Properties	IDBProperties IRowsetInfo	Yes

Table 40. Parameter Methods

Method/Property	OLE DB Interface/Property	IBM OLE DB Support
AppendChunk	ISequentialStream	Yes
Attributes Direction Name NumericScale Precision Scale Size Type	ICommandWithParameter DBSCHEMA _PROCEDURE_PARAMETERS	Yes No Yes Yes Yes Yes Yes
Value	IAccessor ICommand	Yes

Table 41. Parameter Collection

Method/Property	OLE DB Interface/Property	IBM OLE DB Support
Properties		Yes

Table 42. RecordSet Methods

Method/Property	OLE DB Interface/Property	IBM OLE DB Support
AddNew	IRowsetChange	Yes
Cancel		Yes

Table 42. RecordSet Methods (continued)

Method/Property	OLE DB Interface/Property	IBM OLE DB Support
CancelBatch	IRowsetUpdate::Undo	Yes (Cursor Service)
CancelUpdate		Yes (Cursor Service)
Clone	IRowsetLocate	Yes
Close	IAccessor IRowset	Yes
CompareBookmarks		No
Delete	IRowsetChange	Yes
GetRows	IAccessor IRowset	Yes
Move	IRowset IRowsetLocate	Yes
MoveFirst	IRowset IRowsetLocate	Yes
MoveNext	IRowset IRowsetLocate	Yes
MoveLast	IRowsetLocate	Yes
MovePrevious	IRowsetLocate	Yes
NextRecordSet	IMultipleResults	Yes
Open	ICommand IOpenRowset	Yes
Requery	ICommand IOpenRowset	Yes
Resync	IRowsetRefresh	Yes (Cursor Service)
Supports	IRowsetInfo	Yes
Update UpdateBatch	IRowsetChange IRowsetUpdate	Yes Yes (Cursor Service)

Table 43. RecordSet Properties

Method/Property	OLE DB Interface/Property	IBM OLE DB Support
AbsolutePage	IRowsetLocate IRowsetScroll	Yes Yes ¹
AbsolutePosition	IRowsetLocate IRowsetScroll	Yes Yes ¹
ActiveConnection	IDBCreateSession IDBInitialize	Yes

Table 43. RecordSet Properties (continued)

Method/Property	OLE DB Interface/Property	IBM OLE DB Support
BOF	(ADO specific)	
Bookmark	IAccessor IRowsetLocate	Yes
CacheSize	cRows in IRowsetLocate IRowset	Yes
CursorType adOpenDynamic adOpenForwardOnly adOpenKeySet adOpenStatic	ICommandProperties	No Yes Yes Yes
EditMode	IRowsetUpdate	Yes (Cursor Service)
EOF	(ADO specific)	
Filter	IRowsetLocate IRowsetView IRowsetUpdate IViewChapter IViewFilter	No
LockType	ICommandProperties	Yes
MarshallOption		No
MaxRecords	ICommandProperties IOpenRowset	Yes
PageCount	IRowsetScroll	Yes ¹
PageSize	(ADO specific)	
Sort	(ADO specific)	
Source	(ADO specific)	
State	(ADO specific)	
Status	IRowsetUpdate	Yes (Cursor Service)
Note:		
1. The values to be returned are approximations. Deleted rows will not be skipped.		

Table 44. RecordSet Collection

Method/Property	OLE DB Interface/Property	IBM OLE DB Support
Fields	IColumnInfo	Yes
Properties	IDBProperties IRowsetInfo::GetProperties	Yes

Compilation and linking of C/C++ applications and the IBM OLE DB Provider

C/C++ applications that use the constant CLSID_IBMDADB2 must include the `ibmdadb2.h` file, which can be found in the `SQLLIB\include` directory.

C/C++ applications that use the constant CLSID_IBMDADB2 must include the `ibmdadb2.h` file, which can be found in the `SQLLIB\include` directory. These applications must define `DBINITCONSTANTS` before the include statement. The following example shows the correct sequence of C/C++ preprocessor directives:

```
#define DBINITCONSTANTS
#include "ibmdadb2.h"
```

Connections to data sources in C/C++ applications using the IBM OLE DB Provider

To connect to a DB2 data source using the IBM OLE DB Provider for DB2 in a C/C++ application, you can use one of the two OLE DB core interfaces, `IDBPromptInitialize` or `IDataInitialize`, or you can call the COM API `CoCreateInstance`.

To connect to a DB2 data source using the IBM OLE DB Provider for DB2 in a C/C++ application, you can use one of the two OLE DB core interfaces, `IDBPromptInitialize` or `IDataInitialize`, or you can call the COM API `CoCreateInstance`. The `IDataInitialize` interface is exposed by the OLE DB Service Component, and the `IDBPromptInitialize` is exposed by the Data Links Component.

COM+ distributed transaction support and the IBM OLE DB Provider

OLE DB applications running in a Microsoft Component Services (COM+) environment on Windows 2000 or XP can use the `ITransactionJoin` interface.

OLE DB applications running in a Microsoft Component Services (COM+) environment on Windows 2000 or XP can use the `ITransactionJoin` interface to participate in distributed transactions with multiple DB2 for Linux, UNIX, and Windows, host, and System i[®] database servers as well as other resource managers that comply with the COM+ specifications.

Prerequisites

To use the COM+ distributed transaction support offered by the IBM OLE DB Provider for DB2, ensure that your server meets the following prerequisites.

Note: These requirements are only for the Windows-based computers where DB2 clients are installed.

- Windows 2000 with Service Pack 3 or later
- Windows XP

Enablement of COM+ support in C/C++ database applications

To run a C or C++ application in COM+ transactional mode, you can create the `IBMDADB2` data source instance using `CoCreateInstance`, get a session object, and use `JoinTransaction`.

To run a C or C++ application in COM+ transactional mode, you can create the IBM DADB2 data source instance using `CoCreateInstance`, get a session object, and use `JoinTransaction`. See the description of how to connect a C or C++ application to a data source for more information.

To run an ADO application in COM+ transactional mode, see the description of how to connect a C or C++ application to a data source.

To use a component in an COM+ package in transactional mode, set the `Transactions` property of the component to one of the following values:

- "Required"
- "Required New"
- "Supported"

For information about these values, see the COM+ documentation.

Chapter 3. IBM Data Server Provider for .NET

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

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

- DB2 for Linux, UNIX, and Windows, Version 9.1, Version 9.5, Version 9.7, and Version 9.8
- DB2 Universal Database Version 8 for Windows, UNIX, and Linux
- DB2 for z/OS Version 8, Version 9, and Version 10 , through DB2 Connect
- DB2 Universal Database for AS/400 Version 5 Release 3 and later, through DB2 Connect
- IBM Informix Version 11.10, and Version 11.50

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

System Requirements for the IBM Data Server Provider for .NET

Before using the DB2 Install program to install the IBM Data Server Provider for .NET, you must already have the .NET Framework installed on your computer. If the .NET Framework is not installed, the DB2 Install program will not install the IBM Data Server Provider for .NET.

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

Using the IBM Data Server Provider for .NET Documentation

The IBM Data Server Provider for .NET reference documentation contains detailed provider-specific information. Many of the IBM Data Server Provider for .NET classes inherit or implement members from other .NET Framework classes or interfaces. The provider documentation includes a summary of all members within each of these classes. For more detailed information about a specific inherited member, see the appropriate topic in the Microsoft .NET Framework SDK.

Namespaces

Namespace	Description
IBM.Data.DB2	The .NET Framework data provider for IBM data servers.
IBM.Data.DB2Types	Provides classes for native data types in IBM data servers.
IBM.Data.Informix	The .NET Framework data provider for Informix data servers.

IBM.Data.DB2 Namespace

The IBM.Data.DB2 namespace is the .NET Framework Data Provider for IBM data servers. The IBM Data Server Provider for .NET extends support for the ADO.NET interface and delivers high-performing, secure access to data. This provider is sometimes called the Common DB2 .NET Provider.

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

- DB2 for Linux, UNIX, and Windows Version 9.1, Version 9.5, Version 9.7, and Version 9.8
- DB2 for z/OS Version 8, Version 9, and Version 10 , through DB2 Connect
- DB2 Universal Database for AS/400 Version 5 Release 3 and later, through DB2 Connect

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

System Requirements for the IBM Data Server Provider for .NET

Before using the DB2 Install program to install the IBM Data Server Provider for .NET, you must already have the .NET Framework installed on your computer. If the .NET Framework is not installed, the DB2 Install program will not install the IBM Data Server Provider for .NET.

For DB2 for i V5R4 and earlier, the following fix is required on the server: APAR ii13348.

Using the IBM Data Server Provider for .NET Documentation

The IBM Data Server Provider for .NET reference documentation contains detailed provider-specific information. Many of the IBM Data Server Provider for .NET classes inherit or implement members from other .NET Framework classes or interfaces. The provider documentation includes a summary of all members within each of these classes. For more detailed information about a specific inherited member, see the appropriate topic in the Microsoft .NET Framework SDK.

Developing IBM Data Server Provider for .NET Applications

The IBM Data Server Provider for .NET provides functionality for connecting to a database, executing commands, and retrieving results. Those results can be processed directly, or placed in an ADO.NET DataSet for further processing while in a disconnected state. While in the DataSet, data can be exposed to the user, combined with other data from multiple sources, or passed remotely between tiers. Any processing performed on the data while in the DataSet can then be reconciled to the database.

To use the IBM Data Server Provider for .NET, you must add an `imports` or `using` statement for the IBM.Data.DB2 namespace to your application .DLL. For example:

```
[Visual Basic]
Imports IBM.Data.DB2
[C#]
using IBM.Data.DB2;
```

You also must include a reference to `IBM.Data.DB2.dll` in your application's project. In Visual Studio you can do this either from the References section for your project in the Solution Explorer, or from the **Project > Add Reference** menu option. If you are compiling a program from the command line for Framework 2.0 or later, your **csc** or **vbc** commands should include the following option:












```
/r:<install dir>\bin\netf20\IBM.Data.DB2.dll
```






To establish a connection to one of the supported data servers, you will need to construct a `DB2Connection` object and provide it with a valid DB2 .NET connection string. See the `DB2Connection.ConnectionString` property for information about the supported keywords.

For detailed information about how to best use the `IBM.Data.DB2` namespace, see the documentation on the following IBM Data Server Provider for .NET classes:





- `DB2DataAdapter`
- `DB2Command`
- `DB2Connection`
- `DB2DataReader`

Classes





Class	Description
 <code>DB2BulkCopy</code>	Facilitates the copying of rows from one data source to another.
 <code>DB2BulkCopyColumnMapping</code>	Represents a column mapping from the data source table to the destination table.
 <code>DB2BulkCopyColumnMappingCollection</code>	Represents a collection of column mappings from the data source table to the destination table.
 <code>DB2Command</code>	Represents an SQL statement or stored procedure to execute against a database.
 <code>DB2CommandBuilder</code>	Automatically generates single-table commands used to reconcile changes made to a <code>DataSet</code> with the associated database.
 <code>DB2Connection</code>	Represents an open connection to a database.
 <code>DB2ConnectionStringBuilder</code>	Facilitates generic and IBM Data Server Provider for .NET-specific approaches to generating valid connection strings.
 <code>DB2DataAdapter</code>	Represents a set of data commands and a connection to a database that are used to fill the <code>DataSet</code> and update the database.
 <code>DB2DataReader</code>	Provides a way of reading a forward-only stream of data rows from a database.
 <code>DB2DataSourceEnumerator</code>	Provides a way to discover visible IBM family data sources.
 <code>DB2Error</code>	Collects information relevant to a warning or error returned by the database.

Class	Description
 DB2ErrorCollection	Collects all errors generated by the DB2DataAdapter.
 DB2Exception	The exception that is generated when an error is returned by an IBM data server.
 DB2Factory	Represents a set of methods for creating instances of the System.Data.Common data source classes for the data provider.
 DB2InfoMessageEventArgs	Provides data for the InfoMessageevent.
 DB2Parameter	Represents a parameter to a DB2Command and optionally, its mapping to a DataColumn.
 DB2ParameterCollection	Represents a collection of parameters relevant to a DB2Command as well as their mappings to columns in a DataSet.
 DB2Permission	Enables the IBM Data Server Provider for .NET to ensure that a user has a security level adequate to access an IBM database.
 DB2PermissionAttribute	Associates a security action with a custom security attribute.
 DB2Record	Represents a read-only record.
 DB2ResultSet	Provides multi-directional scrolling through a bindable stream of data rows from a database. In addition, DB2ResultSet instances can also update, delete, and insert rows.
 DB2RowsCopiedEventArgs	Provides data for the DB2RowsCopied event.
 DB2RowUpdatedEventArgs	Provides data for the RowUpdated event.
 DB2RowUpdatingEventArgs	Provides data for the RowUpdating event.
 DB2Transaction	Represents an SQL transaction to be made at a database.
 DB2UpdatableRecord	Represents a new row to be created in a DB2ResultSet instance.
 DB2XmlAdapter	Represents a means of retrieving from a database and filling an XPathDocument instance.

Delegates

Delegate	Description
 DB2InfoMessageEventHandler	Represents the method that will handle the InfoMessageevent of a DB2Connection object.
 DB2RowsCopiedEventHandler	Represents the method that will handle the DB2RowsCopied event of a DB2BulkCopy object.
 DB2RowUpdatedEventHandler	Represents the method that will handle the RowUpdated event of a DB2DataAdapter object.
 DB2RowUpdatingEventHandler	Represents the method that will handle the RowUpdating event of a DB2DataAdapter object.

Enumerations

Enumeration	Description
 DB2BulkCopyOptions	Specify options to use with DB2BulkCopy. These options are bit flags, which enable you to combine them in bit masks.
 DB2CursorType	Specify cursor options to use with DB2ResultSet. These options are bit flags, which enable you to combine them in bit masks.
 DB2ResultSetOptions	Specify result set options to use with DB2ResultSet. These options are bit flags, which enable you to combine them in bit masks.
 DB2Type	Specifies the data type of a field, property, or DB2Parameter.

Reference

“IBM.Data.DB2Types Namespace”

The IBM.Data.DB2Types namespace provides classes and structures that represent DB2 data types for the .NET Framework Data Provider.

Chapter 3, “IBM Data Server Provider for .NET,” on page 71

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

IBM.Data.DB2Types Namespace

The IBM.Data.DB2Types namespace provides classes and structures that represent DB2 data types for the .NET Framework Data Provider.

The following table shows mappings between DB2Type data types, DB2 data types, Informix data types, Microsoft .NET Framework types, and DB2Types classes and structures.

Table 45. Mappings between data types, classes and structures

Category	DB2Types Classes and Structures	DB2Type Data Type	DB2 Data Type	Informix Data Type	.NET Data Type
Numeric	DB2Int16	SmallInt	SMALLINT	BOOLEAN, SMALLINT	Int16
	DB2Int32	Integer	INT	INTEGER, INT, SERIAL	Int32
	DB2Int64	BigInt, BigSerial	BIGINT	BIGINT, BIGSERIAL, INT8, SERIAL8	Int64
	DB2Real, DB2Real370	Real	REAL	REAL, SMALLFLOAT	Single
	DB2Double	Double	DOUBLE PRECISION	DECIMAL (≤31), DOUBLE PRECISION	Double
	DB2Double	Float	FLOAT	DECIMAL (32), FLOAT	Double
	DB2Decimal	Decimal	DECIMAL	MONEY	Decimal
	DB2Decimal Float	DecimalFloat	DECFLOAT (16 34)		Decimal
DB2Decimal	Numeric	DECIMAL	DECIMAL (≤31), NUMERIC	Decimal	

6. These data types are not supported as parameters in DB2 .NET common language runtime routines.

7. A DB2ParameterClass.ParameterName property of the type DB2Type.Xml can accept variables of the following types: String, byte[], DB2Xml, and XmlReader.


8. These data types are applicable only to DB2 for z/OS.

9. This data type is only supported for DB2 for z/OS Version 9 and later releases and for DB2 for Linux, UNIX, and Windows Version 9.5 and later releases.

Table 45. Mappings between data types, classes and structures (continued)

Category	DB2Types Classes and Structures	DB2Type Data Type	DB2 Data Type	Informix Data Type	.NET Data Type
Date/Time	DB2Date	Date	DATE	DATETIME (date precision)	Datetime
	DB2Time	Time	TIME	DATETIME (time precision)	TimeSpan
	DB2TimeStamp	Timestamp	TIMESTAMP	DATETIME (time and date precision)	DateTime
	DB2TimeStampOffset	Timestamp WithTimeZone	TIMESTAMP WITH TIME ZONE	N/A	DateTime Offset
XML	DB2Xml	XmlIfxType.Xml	XML		Byte[]
Character data	DB2String	Char	CHAR	CHAR	String
	DB2String	VarChar	VARCHAR	VARCHAR	String
	DB2String	LongVarChar	LONG VARCHAR	LVARCHAR	String
Binary data	DB2Binary	Binary	CHAR FOR BIT DATA		Byte[]
	DB2Binary	Binary	BINARY		Byte[]
	DB2Binary	VarBinary	VARBINARY		Byte[]
	DB2Binary	Long VarBinary	LONG VARCHAR FOR BIT DATA		Byte[]
Graphic data	DB2String	Graphic	GRAPHIC		String
	DB2String	VarGraphic	VARGRAPHIC		String
	DB2String	Long VarGraphic	LONG VARGRAPHIC		String
LOB data	DB2Clob	Clob	CLOB	CLOB, TEXT	String
	DB2Blob	Blob	BLOB	BLOB, BYTE	Byte[]
	DB2Clob	DbClob	DBCLOB		String
Row ID	DB2RowId	RowId	ROWID		Byte[]

Enumerations

Enumeration	Description
 DB2RoundingMode	Specify the rounding mode to use with DB2DecimalFloat objects.

Appendix A. Overview of the DB2 technical information

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

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

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

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

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

Documentation feedback

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

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

DB2 technical library in hardcopy or PDF format

The following tables describe the DB2 library available from the IBM Publications Center at www.ibm.com/e-business/linkweb/publications/servlet/pbi.wss. English and translated DB2 Version 10.1 manuals in PDF format can be downloaded from www.ibm.com/support/docview.wss?rs=71&uid=swg27009474.

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

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

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

Table 46. DB2 technical information

Name	Form Number	Available in print	Last updated
<i>Administrative API Reference</i>	SC27-3864-00	Yes	April, 2012
<i>Administrative Routines and Views</i>	SC27-3865-01	No	January, 2013
<i>Call Level Interface Guide and Reference Volume 1</i>	SC27-3866-01	Yes	January, 2013
<i>Call Level Interface Guide and Reference Volume 2</i>	SC27-3867-01	Yes	January, 2013
<i>Command Reference</i>	SC27-3868-01	Yes	January, 2013
<i>Database Administration Concepts and Configuration Reference</i>	SC27-3871-01	Yes	January, 2013
<i>Data Movement Utilities Guide and Reference</i>	SC27-3869-01	Yes	January, 2013
<i>Database Monitoring Guide and Reference</i>	SC27-3887-01	Yes	January, 2013
<i>Data Recovery and High Availability Guide and Reference</i>	SC27-3870-01	Yes	January, 2013
<i>Database Security Guide</i>	SC27-3872-01	Yes	January, 2013
<i>DB2 Workload Management Guide and Reference</i>	SC27-3891-01	Yes	January, 2013
<i>Developing ADO.NET and OLE DB Applications</i>	SC27-3873-01	Yes	January, 2013
<i>Developing Embedded SQL Applications</i>	SC27-3874-01	Yes	January, 2013
<i>Developing Java Applications</i>	SC27-3875-01	Yes	January, 2013
<i>Developing Perl, PHP, Python, and Ruby on Rails Applications</i>	SC27-3876-00	No	April, 2012
<i>Developing RDF Applications for IBM Data Servers</i>	SC27-4462-00	Yes	January, 2013
<i>Developing User-defined Routines (SQL and External)</i>	SC27-3877-01	Yes	January, 2013
<i>Getting Started with Database Application Development</i>	GI13-2046-01	Yes	January, 2013

Table 46. DB2 technical information (continued)

Name	Form Number	Available in print	Last updated
<i>Getting Started with DB2 Installation and Administration on Linux and Windows</i>	GI13-2047-00	Yes	April, 2012
<i>Globalization Guide</i>	SC27-3878-00	Yes	April, 2012
<i>Installing DB2 Servers</i>	GC27-3884-01	Yes	January, 2013
<i>Installing IBM Data Server Clients</i>	GC27-3883-00	No	April, 2012
<i>Message Reference Volume 1</i>	SC27-3879-01	No	January, 2013
<i>Message Reference Volume 2</i>	SC27-3880-01	No	January, 2013
<i>Net Search Extender Administration and User's Guide</i>	SC27-3895-01	No	January, 2013
<i>Partitioning and Clustering Guide</i>	SC27-3882-01	Yes	January, 2013
<i>Preparation Guide for DB2 10.1 Fundamentals Exam 610</i>	SC27-4540-00	No	January, 2013
<i>Preparation Guide for DB2 10.1 DBA for Linux, UNIX, and Windows Exam 611</i>	SC27-4541-00	No	January, 2013
<i>pureXML Guide</i>	SC27-3892-01	Yes	January, 2013
<i>Spatial Extender User's Guide and Reference</i>	SC27-3894-00	No	April, 2012
<i>SQL Procedural Languages: Application Enablement and Support</i>	SC27-3896-01	Yes	January, 2013
<i>SQL Reference Volume 1</i>	SC27-3885-01	Yes	January, 2013
<i>SQL Reference Volume 2</i>	SC27-3886-01	Yes	January, 2013
<i>Text Search Guide</i>	SC27-3888-01	Yes	January, 2013
<i>Troubleshooting and Tuning Database Performance</i>	SC27-3889-01	Yes	January, 2013
<i>Upgrading to DB2 Version 10.1</i>	SC27-3881-01	Yes	January, 2013
<i>What's New for DB2 Version 10.1</i>	SC27-3890-01	Yes	January, 2013
<i>XQuery Reference</i>	SC27-3893-01	No	January, 2013

Table 47. DB2 Connect-specific technical information

Name	Form Number	Available in print	Last updated
<i>DB2 Connect Installing and Configuring DB2 Connect Personal Edition</i>	SC27-3861-00	Yes	April, 2012

Table 47. DB2 Connect-specific technical information (continued)

Name	Form Number	Available in print	Last updated
DB2 Connect Installing and Configuring DB2 Connect Servers	SC27-3862-01	Yes	January, 2013
DB2 Connect User's Guide	SC27-3863-01	Yes	January, 2013

Displaying SQL state help from the command line processor

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

Procedure

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

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

where *sqlstate* represents a valid five-digit SQL state and *class code* represents the first two digits of the SQL state.

For example, ? 08003 displays help for the 08003 SQL state, and ? 08 displays help for the 08 class code.

Accessing different versions of the DB2 Information Center

Documentation for other versions of DB2 products is found in separate information centers on ibm.com[®].

About this task

For DB2 Version 10.1 topics, the *DB2 Information Center* URL is <http://publib.boulder.ibm.com/infocenter/db2luw/v10r1>.

For DB2 Version 9.8 topics, the *DB2 Information Center* URL is <http://publib.boulder.ibm.com/infocenter/db2luw/v9r8/>.

For DB2 Version 9.7 topics, the *DB2 Information Center* URL is <http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/>.

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

For DB2 Version 9.1 topics, the *DB2 Information Center* URL is <http://publib.boulder.ibm.com/infocenter/db2luw/v9/>.

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

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

A locally installed DB2 Information Center must be updated periodically.

Before you begin

A DB2 Version 10.1 Information Center must already be installed. For details, see the “Installing the DB2 Information Center using the DB2 Setup wizard” topic in *Installing DB2 Servers*. All prerequisites and restrictions that applied to installing the Information Center also apply to updating the Information Center.

About this task

An existing DB2 Information Center can be updated automatically or manually:

- Automatic updates update existing Information Center features and languages. One benefit of automatic updates is that the Information Center is unavailable for a shorter time compared to during a manual update. In addition, automatic updates can be set to run as part of other batch jobs that run periodically.
- Manual updates can be used to update existing Information Center features and languages. Automatic updates reduce the downtime during the update process, however you must use the manual process when you want to add features or languages. For example, a local Information Center was originally installed with both English and French languages, and now you want to also install the German language; a manual update will install German, as well as, update the existing Information Center features and languages. However, a manual update requires you to manually stop, update, and restart the Information Center. The Information Center is unavailable during the entire update process. In the automatic update process the Information Center incurs an outage to restart the Information Center after the update only.

This topic details the process for automatic updates. For manual update instructions, see the “Manually updating the DB2 Information Center installed on your computer or intranet server” topic.

Procedure

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

1. On Linux operating systems,
 - a. Navigate to the path where the Information Center is installed. By default, the DB2 Information Center is installed in the `/opt/ibm/db2ic/V10.1` directory.
 - b. Navigate from the installation directory to the `doc/bin` directory.
 - c. Run the `update-ic` script:

```
update-ic
```
2. On Windows operating systems,
 - a. Open a command window.
 - b. Navigate to the path where the Information Center is installed. By default, the DB2 Information Center is installed in the `<Program Files>\IBM\DB2 Information Center\Version 10.1` directory, where `<Program Files>` represents the location of the Program Files directory.
 - c. Navigate from the installation directory to the `doc\bin` directory.
 - d. Run the `update-ic.bat` file:

```
update-ic.bat
```

Results

The DB2 Information Center restarts automatically. If updates were available, the Information Center displays the new and updated topics. If Information Center updates were not available, a message is added to the log. The log file is located in `doc\eclipse\configuration` directory. The log file name is a randomly generated number. For example, `1239053440785.log`.

Manually updating the DB2 Information Center installed on your computer or intranet server

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

About this task

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

1. Stop the *DB2 Information Center* on your computer, and restart the Information Center in stand-alone mode. Running the Information Center in stand-alone mode prevents other users on your network from accessing the Information Center, and allows you to apply updates. The Workstation version of the DB2 Information Center always runs in stand-alone mode. .
2. Use the Update feature to see what updates are available. If there are updates that you must install, you can use the Update feature to obtain and install them

Note: If your environment requires installing the *DB2 Information Center* updates on a machine that is not connected to the internet, mirror the update site to a local file system by using a machine that is connected to the internet and has the *DB2 Information Center* installed. If many users on your network will be installing the documentation updates, you can reduce the time required for individuals to perform the updates by also mirroring the update site locally and creating a proxy for the update site.

If update packages are available, use the Update feature to get the packages. However, the Update feature is only available in stand-alone mode.

3. Stop the stand-alone Information Center, and restart the *DB2 Information Center* on your computer.

Note: On Windows 2008, Windows Vista (and higher), the commands listed later in this section must be run as an administrator. To open a command prompt or graphical tool with full administrator privileges, right-click the shortcut and then select **Run as administrator**.

Procedure

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

1. Stop the *DB2 Information Center*.
 - On Windows, click **Start > Control Panel > Administrative Tools > Services**. Then right-click **DB2 Information Center** service and select **Stop**.
 - On Linux, enter the following command:
`/etc/init.d/db2icdv10 stop`
2. Start the Information Center in stand-alone mode.
 - On Windows:
 - a. Open a command window.

- b. Navigate to the path where the Information Center is installed. By default, the *DB2 Information Center* is installed in the *Program_Files\IBM\DB2 Information Center\Version 10.1* directory, where *Program_Files* represents the location of the Program Files directory.
- c. Navigate from the installation directory to the *doc\bin* directory.
- d. Run the *help_start.bat* file:


```
help_start.bat
```
- On Linux:
 - a. Navigate to the path where the Information Center is installed. By default, the *DB2 Information Center* is installed in the */opt/ibm/db2ic/V10.1* directory.
 - b. Navigate from the installation directory to the *doc/bin* directory.
 - c. Run the *help_start* script:


```
help_start
```

The systems default Web browser opens to display the stand-alone Information Center.

3. Click the **Update** button (🔄). (JavaScript must be enabled in your browser.) On the right panel of the Information Center, click **Find Updates**. A list of updates for existing documentation displays.
4. To initiate the installation process, check that the selections you want to install, then click **Install Updates**.
5. After the installation process has completed, click **Finish**.
6. Stop the stand-alone Information Center:
 - On Windows, navigate to the *doc\bin* directory within the installation directory, and run the *help_end.bat* file:


```
help_end.bat
```

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


```
help_end
```

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


```
/etc/init.d/db2icdv10 start
```

Results

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

DB2 tutorials

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

Before you begin

You can view the XHTML version of the tutorial from the Information Center at <http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/>.

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

DB2 tutorials

To view the tutorial, click the title.

"pureXML®" in pureXML Guide

Set up a DB2 database to store XML data and to perform basic operations with the native XML data store.

DB2 troubleshooting information

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

DB2 documentation

Troubleshooting information can be found in the *Troubleshooting and Tuning Database Performance* or the Database fundamentals section of the *DB2 Information Center*, which contains:

- Information about how to isolate and identify problems with DB2 diagnostic tools and utilities.
- Solutions to some of the most common problem.
- Advice to help solve other problems you might encounter with your DB2 database products.

IBM Support Portal

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

Access the IBM Support Portal at http://www.ibm.com/support/entry/portal/Overview/Software/Information_Management/DB2_for_Linux,_UNIX_and_Windows

Terms and conditions

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

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

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

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

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

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

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

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

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

Appendix B. Notices

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

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

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

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

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

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 Japan

The following paragraph does not apply to the United Kingdom or any other country/region where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions; therefore, this statement may not apply to you.

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

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

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

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

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

IBM Canada Limited
U59/3600
3600 Steeles Avenue East
Markham, Ontario L3R 9Z7
CANADA

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

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

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

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

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

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

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating

platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

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

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

Trademarks

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

The following terms are trademarks or registered trademarks of other companies

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

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

Index

Special characters

.NET

- application deployment 2
- application development software 3
- C# applications
 - compiler options 25
 - database connections 7
 - link options 25
 - result sets 13
 - SQL statements 11
 - stored procedures 14
 - Windows 23
- IBM Data Server Provider
 - System.Diagnostics.Trace 17
- pureQuery
 - optimizing queries 18
- Visual Basic applications
 - compiler options 23
 - database connections 7
 - link options 23
 - result sets 13
 - SQL statements 11
 - stored procedures 14
 - Windows 22

A

- ActiveX Data Object (ADO) specification
 - IBM Data Server Provider for .NET 4
- ADO applications
 - connection string keywords 61
 - IBM OLE DB Provider support for ADO methods and properties 62
 - limitations 61
 - stored procedures 61
 - updatable scrollable cursors 61
- ADO.NET applications
 - common base classes 6
 - developing 1
- ADORecordset objects 32
- application development
 - IBM Data Server Provider for .NET 4
 - IBM Database Add-Ins for Visual Studio 4
- applications
 - ADO
 - limitations 61
 - updatable scrollable cursors 61
 - connecting to data sources
 - IBM OLE DB Provider 68
 - IBM OLE DB Provider 42
 - Visual Basic 61

C

- C/C++ language
 - applications
 - IBM OLE DB Provider 68
- C# .NET
 - applications
 - building (Windows) 23

C# .NET (continued)

- applications (continued)
 - compiler options 25
 - link options 25
- COM
 - distributed transaction support 68
- COM+ applications
 - connection pooling 7
- connection keywords
 - ODBC .NET Data Provider 33
 - OLE DB .NET Data Provider 26
- connection pooling
 - IBM Data Server Provider for .NET 7
 - OLE DB .NET Data Provider 31
- cursors
 - IBM OLE DB Provider 45
- scrollable
 - ADO applications 61
- updatable
 - ADO applications 61

D

- data types
 - ADO.NET database applications 9
 - mappings
 - OLE DB and DB2 45
- DB2 Information Center
 - updating 81, 82
 - versions 80
- documentation
 - overview 77
 - PDF files 77
 - printed 77
 - terms and conditions of use 84

E

- Enterprise Library data access module 21

H

- help
 - SQL statements 80

I

- IBM Data Server Provider for .NET
 - calling stored procedures 14
 - common base classes 6
 - connecting to databases 7
 - connection pooling 7
 - data types 9
 - database system requirements 5
 - Microsoft Entity Framework 18
 - overview 1, 4, 6, 22, 71
 - reading result sets 13
 - simultaneously accessing result sets 16
 - SQL statements 11

IBM Database Add-Ins for Visual Studio
overview 4

IBM OLE DB Provider

ADO

applications 61

methods 62

properties 62

application types 42

C/C++ applications 68

COM support 68, 69

connecting to data sources 60, 61

consumers 41

cursors 45, 61

data conversion

DB2 types to OLE DB types 49

OLE DB types to DB2 types 47

installing 41

LOBs 42

OLE DB components 53

OLE DB interfaces 53

OLE DB properties 55

OLE DB services 45

overview 41

providers 41

restrictions 52

schema rowsets 42

system requirements 41

threading 42

versions 41

Visual Basic applications 61

IBM.Data.DB2 namespace 72

IBM.Data.IfxTypesDB2Types namespace 75

L

large objects (LOBs)

IBM OLE DB Provider 42

M

Microsoft Entity Framework

IBM Data Server Provider for .NET 18

Microsoft OLE DB Provider for ODBC

OLE DB support 53

Microsoft Transaction Server (MTS)

COM distributed transaction support 68

support in DB2 69

N

notices 87

O

ODBC .NET Data Provider

overview 1, 33

restrictions 33

OLE DB

connections to data sources using IBM OLE DB

Provider 60

data conversion

from DB2 to OLE DB types 49

from OLE DB to DB2 types 47

data types

mappings to DB2 data types 45

OLE DB (*continued*)

IBM OLE DB Provider support overview 53

properties supported by OLE DB .NET Data Provider 55

services automatically enabled 45

table functions 41

OLE DB .NET Data Provider

ADORRecordset objects 32

OLE applications

connection pooling 31

time columns 31

overview 1, 26

restrictions 27

P

problem determination

information available 84

tutorials 84

properties

OLE DB 55

R

result sets

reading

IBM Data Server Provider for .NET 13

S

SAMPLE database

connecting

ODBC .NET Data Provider 33

OLE DB .NET Data Provider 26

schemas

rowsets 42

SQL statements

executing

IBM Data Server Provider for .NET 11

help

displaying 80

system requirements

IBM OLE DB Provider for DB2 41

SystemDiagnosticsTrace

tracing

IBM Data Server Provider for .NET 17

T

terms and conditions

publications 84

threads

IBM OLE DB Provider for DB2 41, 42

Tracing IBM Data Server Provider

System.Diagnostics.Trace 17

troubleshooting

online information 84

tutorials 84

trusted contexts

connection string keywords 8

tutorials

list 84

problem determination 84

pureXML 84

troubleshooting 84

U

updates

DB2 Information Center 81, 82

V

Visual Basic

applications 61

cursor considerations 61

data control support 61

Visual Basic .NET

applications

building 22

compiler options 23

link options 23



Printed in USA

SC27-3873-01



Spine information:

IBM DB2 10.1 for Linux, UNIX, and Windows

Developing ADO.NET and OLE DB Applications

