

# Quick Start: DataDirect Connect<sup>®</sup> for ADO.NET Data Providers

Progress<sup>®</sup> DataDirect Connect<sup>®</sup> for ADO.NET includes the following products:

- DataDirect Connect for ADO.NET data providers
- DataDirect Connect for ADO.NET Entity Framework data providers

The following basic information enables you to connect with and test your driver immediately after installation. For installation instructions, refer to the [DataDirect Connect for ADO.NET Installation Guide](#). This Quick Start covers the following topics:

[Connecting to a Database](#)

[Testing the Connection](#)

[Tuning the Data Providers for Optimal Performance](#)

[Using Microsoft<sup>®</sup> Enterprise Library Patterns and Practices](#)

## Connecting to a Database

Both ADO.NET data providers and ADO.NET Entity Framework data providers require a connection string to connect to a database.

## Using an ADO.NET Data Provider

Once a DataDirect Connect for ADO.NET data provider is installed, you can connect from your application to your database with a connection string using either the data provider-specific objects or the Common Programming Model. The following steps use the Common Programming Model to add connection code for the DB2 data provider.

- 1 Check the beginning of your application. Ensure that the .NET Framework namespace is present. If you want to use DataDirect Bulk Load, add the DataDirect Common assembly.

```
// Access your data source
using System.Data;
using System.Data.Common;
// Add the Common Assembly to add support for DataDirect Bulk Load
using DDTek.Data.Common;
```

- 2 Add the connection information for your server and exception handling code and close the connection.

```
DbProviderFactory factory = DbProviderFactories.getFactory("DDTek.DB2");
DbConnection Conn = factory.createConnection();
Conn.ConnectionString = "Host=10.30.11.189;port=446;
User ID=john;Password=beAch05;Database=NCANSV5;Collection=ADOVS1"
```

```
try
{
    Conn.Open();
    Console.WriteLine("Connection successful! ");
}
}
```

```
catch (Exception ex)
{
    // Connection failed
    Console.WriteLine(ex.Message);
}
// Close the connection
Conn.Close();
```

## Using an ADO.NET Entity Framework Data Provider

The Entity Framework data providers use the corresponding ADO.NET data provider to communicate with the database server. This means that the functionality defined by the ADO.NET data provider applies to the Entity Framework data provider unless otherwise noted in the DataDirect Connect for ADO.NET User's Guide.

NOTE: Developing with the ADO.NET Entity Framework requires that you use Microsoft .NET Framework Version 3.5 SP1 or higher, and Visual Studio 2008 SP1 or higher with a DataDirect Connect ADO.NET Entity Framework data provider.

The following procedure to connect to a data source assumes that you already have the database schema available.

- 1 Create a new application, such as Windows Console, Windows Forms, or ASP.NET, in Visual Studio.
- 2 In the Solution Explorer, right-click on the project and select **Add / New Item**.
- 3 Select ADO.NET Entity Data Model, and then click **Add**.
- 4 The Entity Data Model Wizard appears. Select **Generate from database**; then, click **Next**.
- 5 On the Choose Your Data Connection window, select an established connection from the drop-down list or click **New Connection...** to create a new connection:
  1. On the Connection Properties window, click **Change**.
  2. On the Choose Data Source window, select **Progress DataDirect** in the Data source list; then, select your data provider in the Data provider drop-down list.
  3. Click **OK**.
  4. Provide the necessary connection information on the Connection Properties window, and then click **OK**.
- 6 The Choose Your Data Connection window displays the connection string information and provides additional choices. When you have made your selections, click **Next**.
  1. If the radio buttons are selectable, select **Yes, include the sensitive data in the connection string**.
  2. In the **Save entity connection settings in App.Config as** field, type a name for main data access class or accept the default.
- 7 Click **Finish**. The model is generated and opened in the Model Browser.

Click the following links for specific information about the General tab of each driver:

[DB2  
SQL Server](#)

[Oracle  
Sybase](#)

## DB2

The following examples provide the minimum required options for a connection string for a user with the BINDADD permissions. Note that both IPv4 and IPv6 formats can be used to identify the host.

DB2 for Linux/UNIX/Windows:

```
"Host=Accounting1;Port=50000;User ID=gene;Password=b0gey;Database=Accounting"
```

DB2 for iSeries:

```
"Host=10.30.11.189;Port=446;User ID=roger;Password=newm4n;Database=NCANSV5;  
Collection=ADOVS1"
```

DB2 for z/OS:

```
"Host=10.30.14.109;Port=448;User ID=fred;Password=sailb0at;Database=S390LOC_7U"
```

The name and description for each connection string option required for a minimum connection to a DB2 server are listed below. Refer to Appendix B "Connection String Option Descriptions" of the [DataDirect Connect for ADO.NET User's Guide](#) for a description of all of the connection string options.

- **Database:** Type the name of the database to which you want to connect.  
NOTE: For DB2 for z/OS, use the name of the DB2 location that you want to access. Your system administrator can determine the name of your DB2 location using the following command:

```
DISPLAY DDF
```

- **Host:** Type either the name or the IP address of the server to which you want to connect. For example, if your network supports named servers, you can specify a server name such as DB2server. Or, you can specify an IP address such as 199.226.224.34.

To connect to a DB2 server running on the local machine, specify localhost as the Host connection string option.

- **Password:** Type the case-sensitive password used to connect to your DB2 database.
- **Port:** Type the port of the DB2 DRDA listener. The initial default is 446
- **User ID:** This field is valid and required only if you are connecting to a DB2 database on Linux/UNIX/Windows. Type the name of the database to which you want to connect.

Specifies the case-sensitive DB2 logon ID to use for this connection

When connecting for the first time, the data provider determines whether bind packages exist on the server. If packages do not exist, the data provider creates them automatically using data provider default values.

**IMPORTANT:** You must have the appropriate privileges for the data provider to create and bind packages with your user ID. These privileges are BINDADD for binding packages, CREATEIN on the collection specified by the Package Collection option, and GRANT EXECUTE on the PUBLIC group for executing the packages. These are typically the permissions of a Database Administrator (DBA). If you do not have these privileges, someone that has a user ID with DBA privileges must create packages by connecting with the data provider.

## Oracle

The minimum required options for the connection string are shown in the following example:

```
"Host=Accounting1;Service Name=sales.us.acme.com;User ID=scott;Password=Tiger"
```

NOTE: In most cases, the Oracle port number is the initial default, 1521. If you are using a different port, you must add the Port connection string option.

The name and description for each connection string option required for a minimum connection are described below. Refer to Appendix B “Connection String Option Descriptions” of the [DataDirect Connect for ADO.NET User's Guide](#) for a description of all of the connection string options.

- **Host:** Type either the name (if your network supports named servers) or the IP address of the server to which you want to connect.
- **Port:** Type the port number of your Oracle listener. Check with your database administrator for the correct number.
- **Password:** Type a case-insensitive password used to connect to your Oracle database. A password is required only if security is enabled on your database. If so, contact your system administrator to get your password.
- **Service Name:** Type the Oracle service name that specifies the database used for the connection. The service name is a string that is the global database name—a name that is comprised of the database name and domain name, for example: sales.us.acme.com.
- **User ID:** Type the case-insensitive default user name used to connect to your Oracle database. A user name is required only if security is enabled on your database. If so, contact your system administrator to get your user name.

## SQL Server

The following sample connection string contains the minimum information necessary to connect to a Microsoft SQL Server server:

```
"Host=Accounting1;User ID=mike;Password=b0gey;Database=Accounting"
```

The following list gives the name and description for each connection string option required for a minimum connection. Refer to Appendix B “Connection String Option Descriptions” of the [DataDirect Connect for ADO.NET User's Guide](#) for a description of all of the connection string options.

NOTE: In most cases, the Microsoft SQL Server port number is the default, 1433. If you are using a different port, you must add the Port connection string option.

- **Database:** Type the name of the database to which you want to connect. If you do not specify a value, the default is the database defined by the system administrator for each user.

Alias: Database Name. If both the Database and Database Name options are specified, the value specified for the Database option is used.

- **Host:** Type either the name or the IP address of the server to which you want to connect. For example, if your network supports named servers, you can specify a server name such as SQLSrvr3. Or, you can specify an IP address such as 199.226.224.34.

To use a Microsoft SQL Server server running on the local machine, specify localhost.

- **Port Number:** Type the port number of your Microsoft SQL Server listener. Check with your database administrator for the correct number.

- **Password:** Type a case-sensitive password used to connect to your Microsoft SQL Server database.
- **User ID:** Type the case-insensitive default user name used to connect to your Microsoft SQL Server database.

## Sybase

The name and description for each connection string option required for a minimum connection to a Sybase server are listed below. Refer to Appendix B “Connection String Option Descriptions” of the [DataDirect Connect for ADO.NET User’s Guide](#) for a description of all of the connection string options.

The following connection string contains the minimum information necessary to connect to a Sybase server:

```
"Host=Accounting1;Port=4100;User ID=jimb;Password=r0ger;Database=Accounting"
```

- **Database:** Type the name of the database to which you want to connect. If you do not specify a value, the default is the database defined by the system administrator for each user.  
  
Alias: Database Name
- **Host:** Type either the name or the IP address of the server to which you want to connect. To use a Sybase server running on the local machine, specify localhost.
- **Database Name:** Type the name of the database to which you want to connect by default. If you do not specify a value, the default is the database defined by the system administrator for each user.
- **Port:** Type the port number of your Sybase listener. Check with your database administrator for the correct number.
- **Password:** Specify a case-sensitive password used to connect to your Sybase database.
- **User ID:** Type the case-insensitive default user name used to connect to your Sybase database.

## Testing the Connection

You can test the connection using a small C# application in Visual Studio.

- 1 Create a small Windows test application (refer to the Microsoft .NET Framework documentation).
- 2 Add a button to the default form that is created for you.
- 3 Follow the steps in the “Connecting to a Database” section above, making sure that you modify the connection string values for your environment. Add the code shown in Step 2 of that procedure to the button’s click event.
- 4 Save the test application.
- 5 Click the button in the test application. A message should appear to indicate a successful connection.

## Tuning the Drivers for Optimal Performance

The data providers have connection options that directly affect performance. To tune the data provider for optimal performance, run the Progress DataDirect Performance Wizard, which is installed along with the product and is available in the Visual Studio Tools menu.

The Wizard leads you step-by-step through a series of questions about your application. Based on your answers, the Wizard provides the optimal settings for performance-related connection options:

## Starting the Performance Tuning Wizard

To start the Performance Tuning Wizard:

- 1 In the Visual Studio menu bar, select **Tools**.
- 2 Select **Progress DataDirect / Run Progress DataDirect Performance Wizard**.

You can also start the Performance Tuning Wizard by right-clicking an existing connection in the Server Explorer.

## Tuning Performance with the Performance Tuning Wizard

After you start the Performance Tuning Wizard, a Welcome window appears. Click **Next** to start the process and select a data provider. When you have answered all questions for a data provider, results are displayed in the Result window. You can use the results in the following ways:

- Copy the generated values for connection string options into an existing connection string.
- Create a new connection with the Performance Tuning Wizard connection string values.
- Generate a new application pre-configured with a connection string optimized for your environment. The Performance Tuning Wizard provides options to select the type of application and the version of ADO.NET code that you want to use.

## Using Microsoft® Enterprise Library Patterns and Practices

Using Microsoft® Enterprise Library patterns can simplify application development by wrapping common tasks, such as data access, into portable code that makes it easier to move your application from one database to another.

To use features of the Enterprise Library with your data provider, download Microsoft Enterprise Library 5.0 from the Microsoft Web site.

NOTE: Enterprise Library 5.0 requires Windows 7, Windows Vista SP2, or Windows Server 2003 SP2. If you are using the data providers on Windows XP SP2, you can use Enterprise Library 4.1 (October 2008).

The classes in the Data Access Application Blocks (DAABs) provide access to the most frequently used features of ADO.NET. Applications can use the DAABs for tasks such as passing data through application layers and returning changed data back to the database. Using DAABs eliminates the need to keep writing the same data access tasks for each new or revised application, so you can spend your time more productively.

The DataDirect Connect® for ADO.NET data providers can be used with Data Access Application Blocks. Refer to the technical brief, "Using the Microsoft Enterprise Library Data Access Application Block (DAAB) in Your Application" at <http://web.datadirect.com/resources/ado-net/daab/> for more information.

Applications that use the standard Logging Application Block (LAB) and design patterns can quickly display the SQL generated as part of the ADO.NET Entity Framework data provider. In particular, the LAB is useful if you need to filter logging messages based on category or priority, if you need to format the messages, or if you need to change the destination of the message without changing the application code. The DB2 data provider includes a set of LAB customizations that are useful for the ADO.NET Entity Framework.

© 2010. Progress Software Corporation. All rights reserved.