Foreword by Juval Löwy, President, IDesign

Data Binding with Windows Forms 2.0

Programming Smart Client Data Applications with .NET

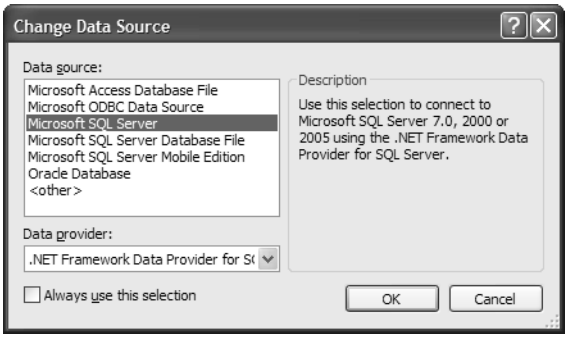
Microsoft® .net Development Series

Brian Noyes

string contains sensitive information (specifically, a username and password). If so, it gives you the option to leave that information out of the connection string so that the sensitive information doesn't get embedded in your configuration file. You can also view the resulting connection string by clicking on the plus sign next to the Connection String group header. This displays the connection string that will be used in a selectable text box. You can select the string (in case you need to copy it to the Clipboard), but you cannot edit the string directly.

2. If you click the *New Connection* button, what you see depends on whether you have configured any connections before. The first time you add a connection, you are prompted to select a data source and provider (see Figure 5.5). You also get the same dialog if you click the *Change* button from the Add Connection dialog (see Figure 5.6). The Add Connection dialog lets you create a new connection based on any of the available providers, and once you do, it too will be added to the list of data connections in Server Explorer.

   The Add Connection dialog is new in Visual Studio 2005, but it's very similar to the one that existed in previous versions. The items presented in this dialog change based on the provider selected. Figure 5.6 shows the settings for the managed SQL Server provider. To configure a SQL Server connection, you specify a server name—this can be a SQL Server instance name on the network or an IP address. If you are referring to the local machine's default instance of SQL Server, you can use one of three shorthand addresses: localhost, (local), or just the


Change Data Source

Data source:
Microsoft Access Database File
Microsoft ODBC Data Source
Microsoft SQL Server
Microsoft SQL Server Database File
Microsoft SQL Server Mobile Edition
Oracle Database
<other>

Description
Use this selection to connect to Microsoft SQL Server 7.0, 2000 or 2005 using the .NET Framework Data Provider for SQL Server.

Data provider: .NET Framework Data Provider for S(

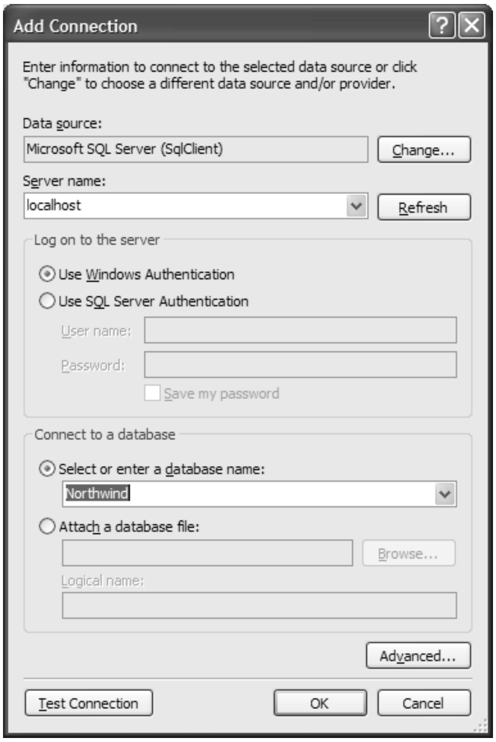
FIGURE 5.5: Data Source and Provider Selection

dot (.) character. You also provide authentication information and specify the database name. This configures the connection string that is used to connect to the database.

If you had selected a SQL Server database file as the data source in Figure 5.5, the dialog would be different and would only let you specify a path to the database file and authentication information. This is the way to specify a data connection for a SQL Server 2005 Express Edition database.

3. Choose whether to save the connection information in your application configuration file (Figure 5.7). Doing so lets you easily modify the connection string when you deploy your application without needing to change any of the source code.

   Visual Studio adds code to the table adapter definitions to read in the connection string from the .config file if it can be found; otherwise, the


Add Connection

Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.

Data source: Microsoft SQL Server (SqlClient)

Server name: localhost

Log on to the server

(● Use Windows Authentication) ○ Use SQL Server Authentication

User name:

Password:

☐ Save my password

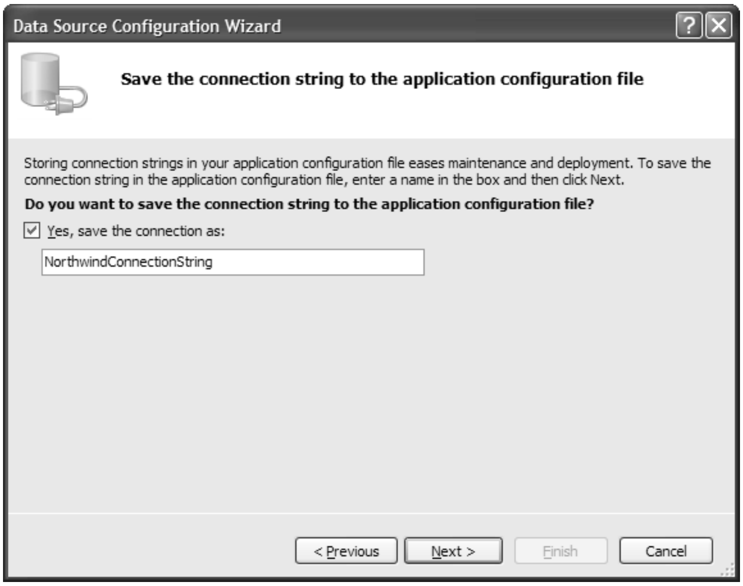
Connect to a database

(● Select or enter a database name:) Northwind

○ Attach a database file:

Logical name:

FIGURE 5.6: Add Connection Dialog



Save the connection string to the application configuration file

Storing connection strings in your application configuration file eases maintenance and deployment. To save the connection string in the application configuration file, enter a name in the box and then click Next.

Do you want to save the connection string to the application configuration file?

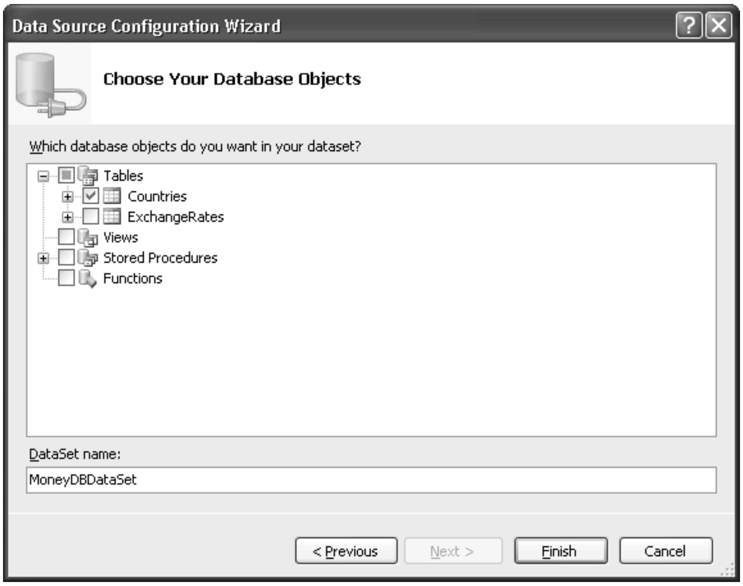
☑ Yes, save the connection as:

NorthwindConnectionString

FIGURE 5.7: Save Connection String Step

table adapter tries to use a hard-coded default (the one specified at design time through this process).

4. In the next window you select the objects from the database that you want to include in the typed data set that will be generated as the output of this wizard process (see Figure 5.8). As mentioned earlier in the book, you can include tables, views, stored procedures, or functions in the data set simply by checking the boxes next to them in the tree of database objects. This is analogous to dragging these objects onto the data set designer surface from Server Explorer. At the bottom of the dialog you can specify the type name for the generated typed data set class.
5. When you click the *Finish* button, the typed data set definition will be added to your project as an XML Schema Definition (XSD) file with an associated typed data set definition source file that is hidden by default. You can view the actual typed data set code by expanding the Solution Explorer tree underneath the XSD file. Underneath the XSD file is a .Designer.cs file for the data set that contains the autogenerated class definitions for the data set and its associated table adapters. Additionally, the Data Sources window will update to show the objects in your data set.


Data Source Configuration Wizard
Choose Your Database Objects
< Previous
Next >
Finish
Cancel

Which database objects do you want in your dataset?

- Tables
  - [✓] Countries
  - [ ] ExchangeRates
- [ ] Views
- [ ] Stored Procedures
- [ ] Functions

DataSet name:

MoneyDBDataSet

FIGURE 5.8: Choose Database Objects Step

Adding a Web Service Data Source

Windows Forms applications are often just a user interface for front-end functionality that resides on a middle-tier server somewhere out on the network. As the use of the Internet and widely distributed networks grow, a common model for distributing functionality and data is through Web services. As such, consuming data returned from a Web service is a common need for Windows Forms applications, and one way to consume data is to bind controls to it. The Data Sources window makes this much easier by generating all the appropriate code for you based on a Web reference that you add to your project.

As mentioned earlier, if you add a Web reference to your project directly, the object types returned by methods on that Web service will show up in the Data Sources window automatically. If you need to add a Web reference for the purposes of data binding and you haven't already added the reference, you can easily do so by selecting a Web service as the data source type when you are in the Data Source Configuration wizard. This displays the Add Web Reference dialog as the next step in the wizard (see Figure 5.9). Note that this is the same dialog you get when you select *Add Web Reference* from Solution Explorer.

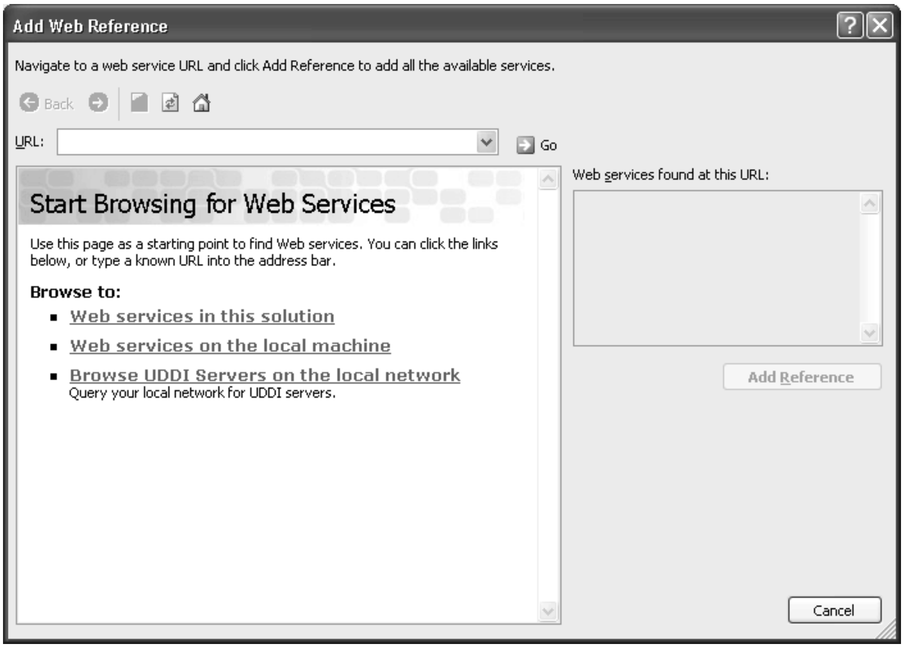

Add Web Reference
Navigate to a web service URL and click Add Reference to add all the available services.
Back
URL:
Go
Start Browsing for Web Services
Use this page as a starting point to find Web services. You can click the links below, or type a known URL into the address bar.
Browse to:
Web services in this solution
Web services on the local machine
Browse UDDI Servers on the local network
Query your local network for UDDI servers.
Web services found at this URL:
Add Reference
Cancel

FIGURE 5.9: Add Web Reference Dialog

You can type in a URL to a Web service or browse for one using the available links in the embedded browser window. When you enter or select a Web service, the browser window displays information about the Web service. If you provide a name for the Web service in the Web Reference Name box after selecting a Web service, this will be used as the namespace that wraps the Web service proxy class and type information within the project. This namespace will be generated as a child namespace of the default project namespace. In addition to the proxy class that lets you make calls to the Web service, class declarations are also generated for all the types that the Web service methods return. It is these type definitions that the Data Sources window lets you use for data binding.

Adding an Object Data Source

In addition to working with database objects and Web service return types, you also often want to perform data binding against objects returned from your business layer or data access layer classes. Note that these include typed data set definitions that reside in a separate assembly, which should be the normal way you design your data access components.

1. To set up data binding to these objects through the designer, select the *Object* data source type in the wizard's first step (see Figure 5.3).
2. Select the object type that you would like to bind to (see Figure 5.10). The wizard displays the objects in a tree view that lets you drill down from the assembly level, through the namespaces, and to the types defined in those namespaces.
3. If the type you want to bind to resides in an assembly that you don't yet have a reference to, click the *Add Reference* button to display the corresponding dialog that lets you add a reference to the assembly to your project. This is the same Add Reference dialog that you use from Solution Explorer. When you select an assembly in it, a normal reference is added to your project.

All public types in a referenced assembly are displayed in the tree of bindable objects, and they can be displayed in the Data Sources window if

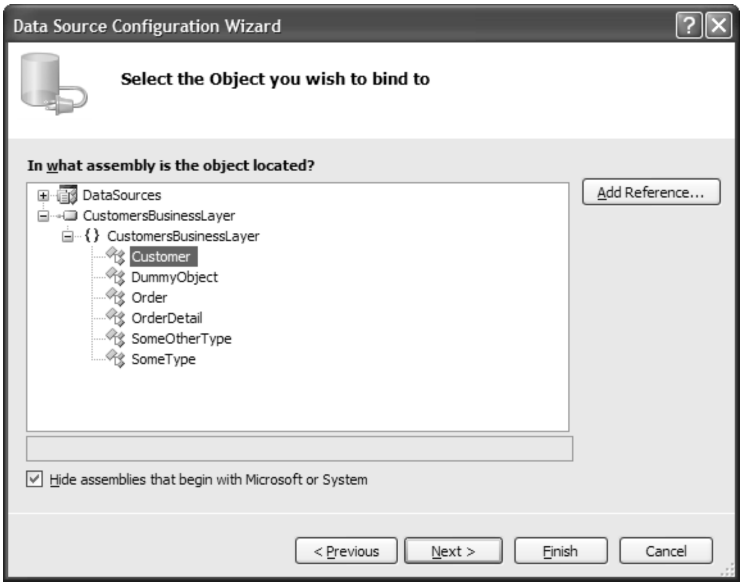

Data Source Configuration Wizard
Select the Object you wish to bind to
In what assembly is the object located?
DataSources
CustomersBusinessLayer
{} CustomersBusinessLayer
Customer
DummyObject
Order
OrderDetail
SomeOtherType
SomeType
Add Reference...
Hide assemblies that begin with Microsoft or System
< Previous
Next >
Finish
Cancel

FIGURE 5.10: Object Type Selection Step