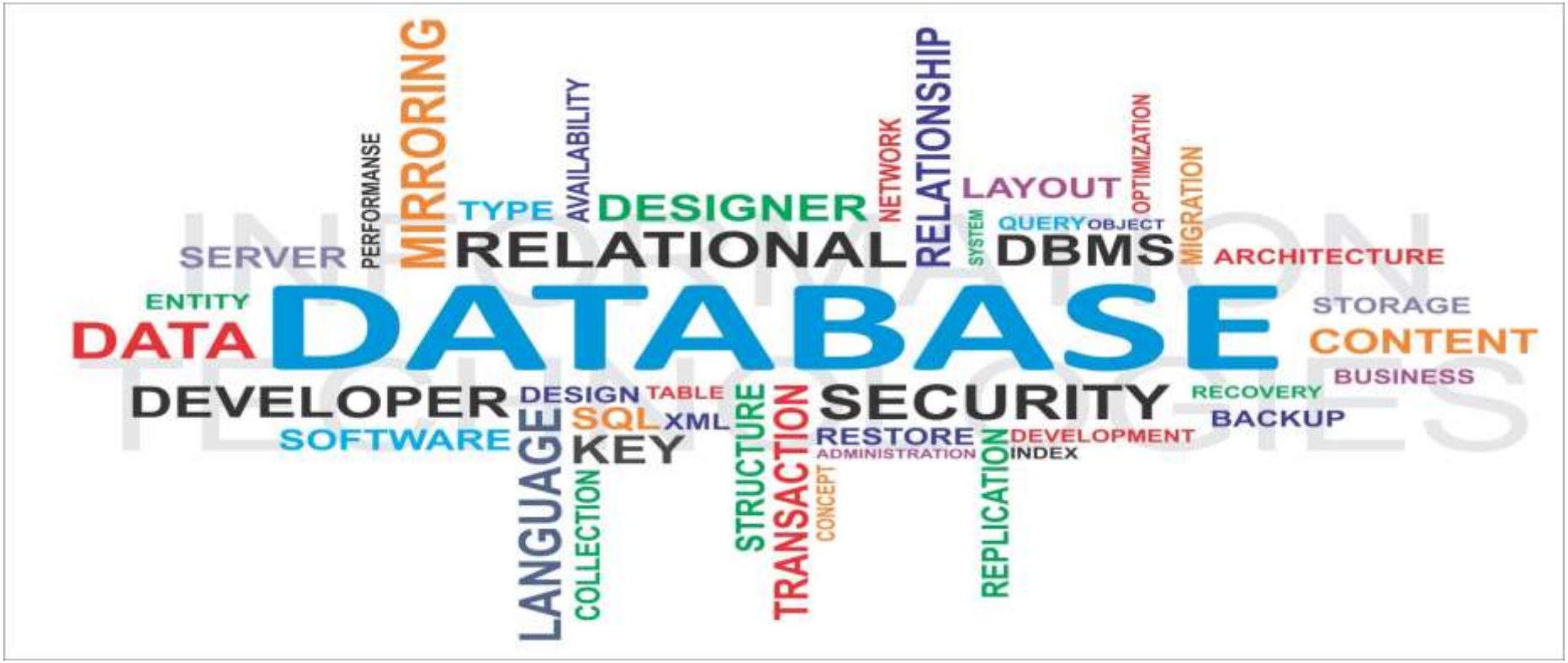
ADO.NET


PERFORMANSE
MIRRORING
AVAILABILITY
TYPE
DESIGNER
NETWORK
RELATIONSHIP
LAYOUT
OPTIMIZATION
MIGRATION
SYSTEM
QUERY
OBJECT
SERVER
RELATIONAL
DBMS
ARCHITECTURE
ENTITY
DATA
DATABASE
STORAGE
CONTENT
BUSINESS
DEVELOPER
DESIGN
TABLE
SECURITY
RECOVERY
BACKUP
SOFTWARE
SQL
XML
KEY
RESTORE
DEVELOPMENT
ADMINISTRATION
INDEX
LANGUAGE
COLLECTION
STRUCTURE
TRANSACTION
CONCEPT
REPLICATION

ActiveX Data Objects

- ADO.NET has a number of classes that :
  - Retrieve Data
  - Manipulate Data
  - Update Data
- VB,C#, C++, J#

What is ADO.NET?

ADO.NET provides a set of classes for working with data. ADO.NET provides:

- **An evolutionary, more flexible successor to ADO**
- **A system designed for disconnected environments**
- **A programming model with advanced XML support**
- **A set of classes, interfaces, structures, and enumerations that manage data access from within the .NET Framework**

ADO vs. ADO.NET

- ADO works great, but:
  - Requires COM and Windows
  - Recordsets don’t travel well over the Internet
  - Connected behavior is hard to work with
- Requires more code
  - ADO.NET solves these problems
  - Uses XML under the covers for all data transport
  - No special code needed to marshal across the Internet

Disconnected?

- ADO.NET offers the capability of working with databases in a disconnected manner.
- An entire database table can be retrieved to a local computer/temp file if it is a network database.
- A connection could also be constant

"I'll be right there. Let me just take care of this user. He's about halfway through a three-hour download."

Web-Centric Applications

- Download the data and process it at a local level.
- If changes are made, the connection can be remade and the changes posted.
- The database could be LAN or Internet based.

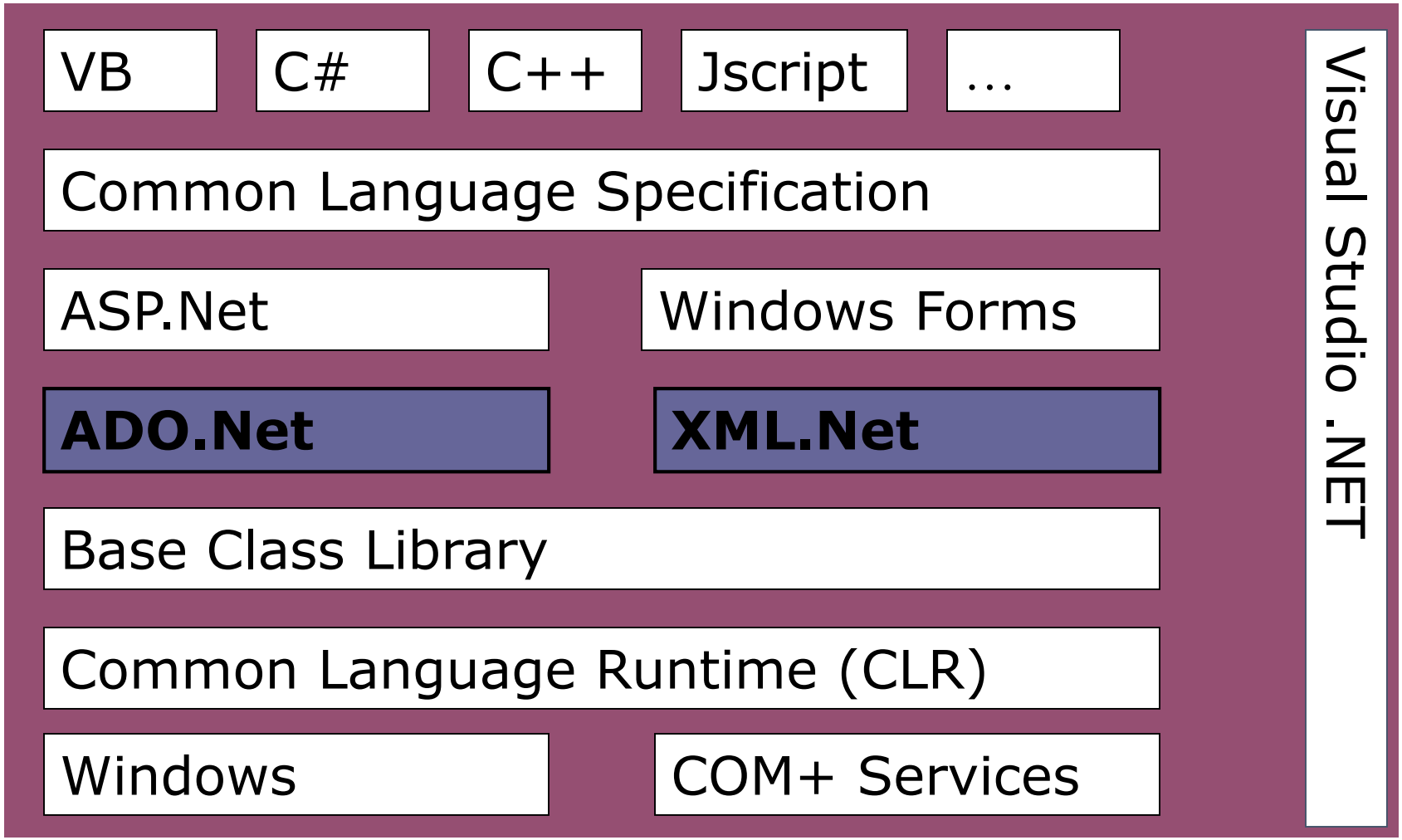
What is ADO.Net?

- The data access classes for the .Net framework
- Designed for highly efficient data access
- Support for XML and disconnected record sets

And the .Net framework?

- A standard cross language interface
- Encapsulation of services, classes and data types
- Uses XML for data representation

Where does ADO sit?

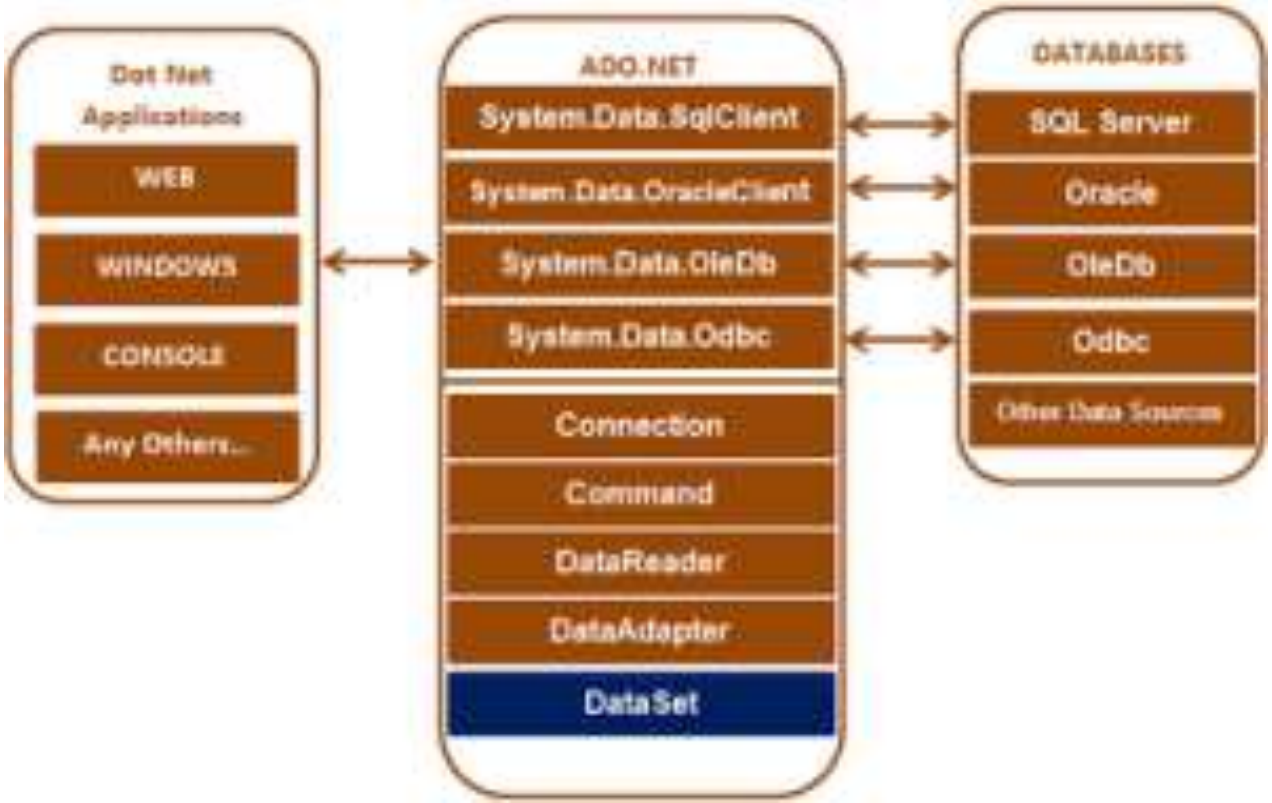


ADO / ADO.Net Comparisons

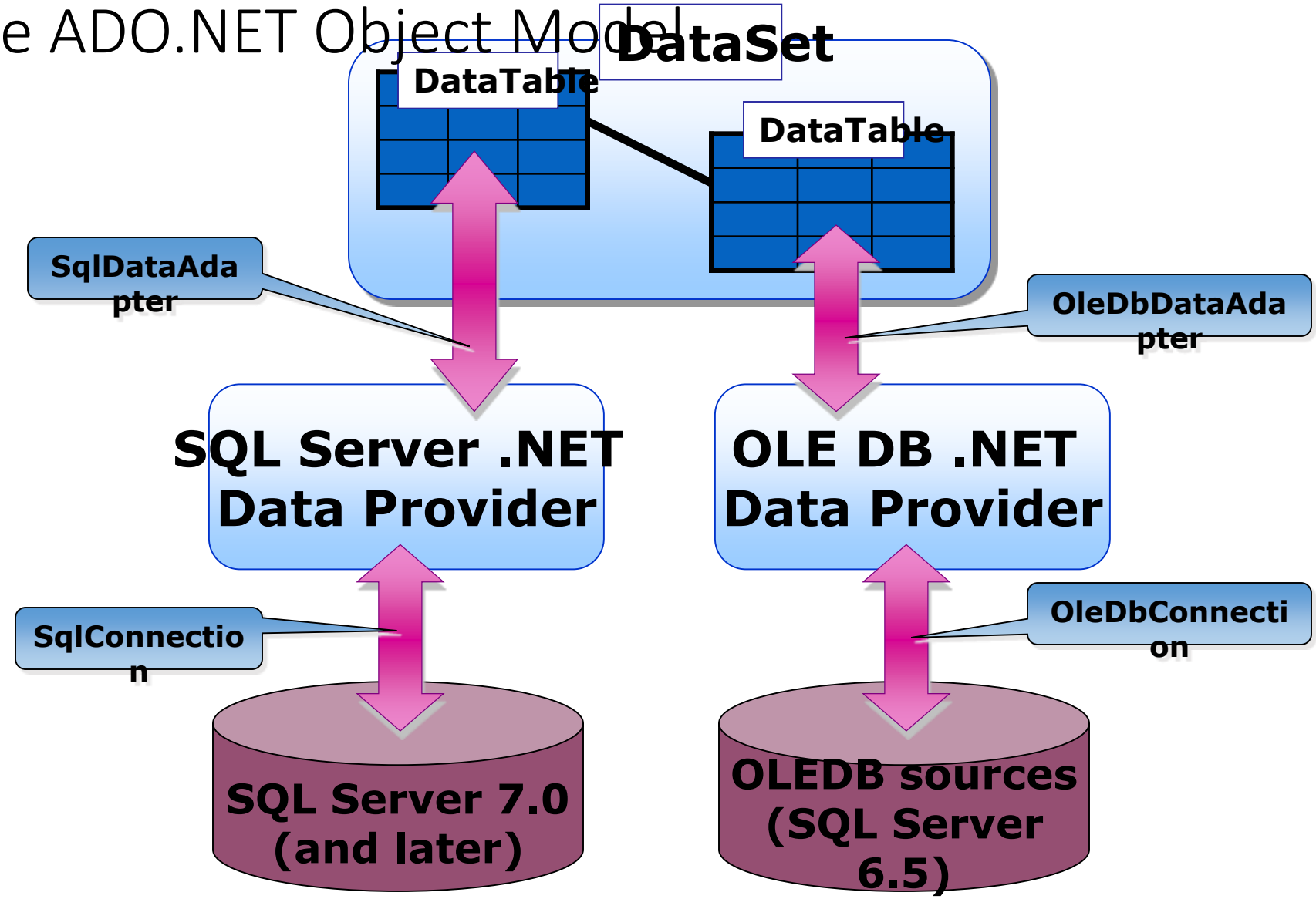
Feature	ADO	ADO.Net
In memory data storage	Recordset object Mimics single table	Dataset object Contains DataTables
Data Reads	Sequential	Sequential or non-sequential
Data Sources	OLE/DB via the Connection object	Managed provider calls the SQL APIs

ADO / ADO.Net Comparisons

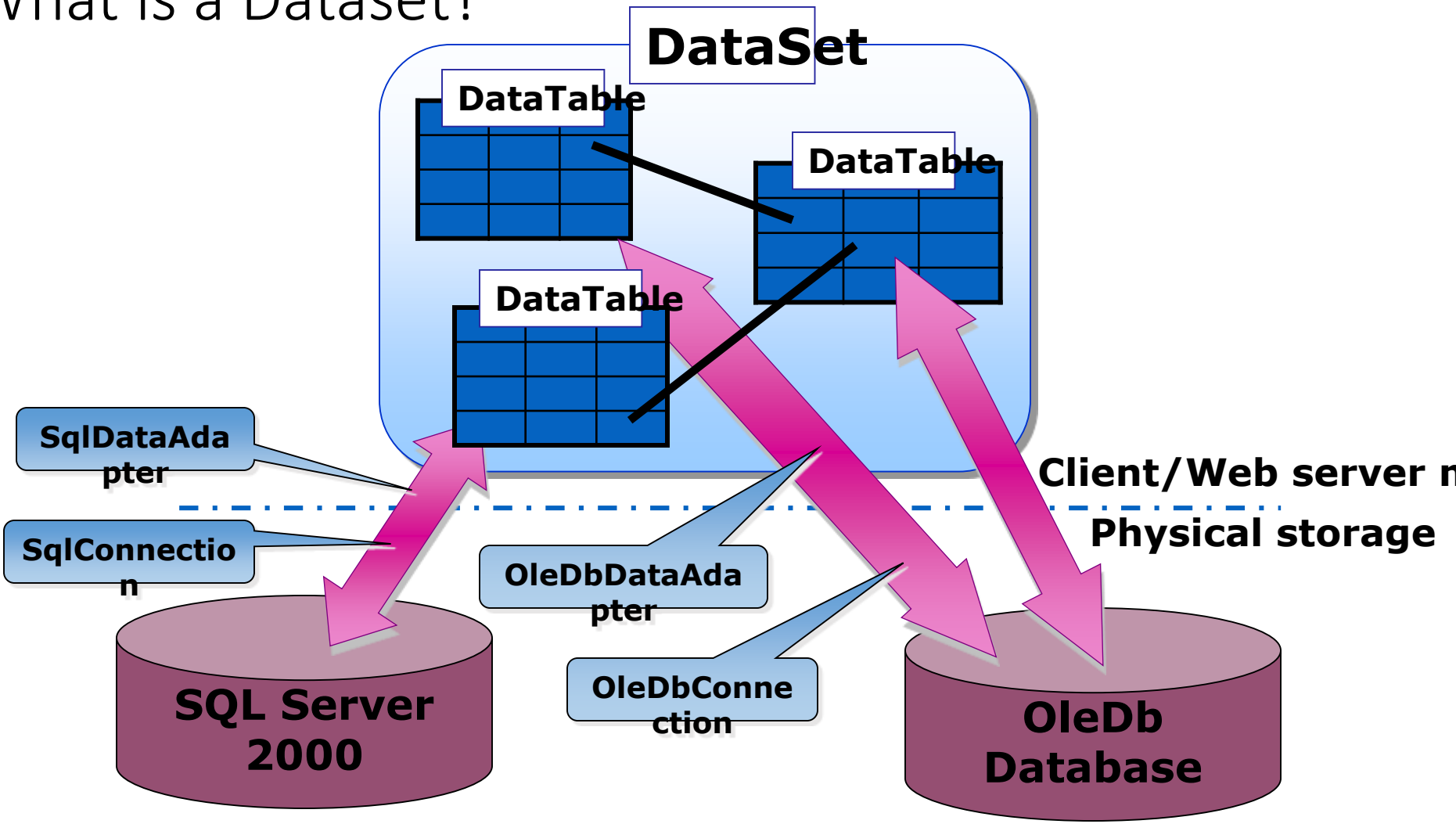
Feature	ADO	ADO.Net
Disconnected data	Limited support, suitable for R/O	Strong support, with updating
Passing datasets	COM marshalling	DataSet support for XML passing
Scalability	Limited	Disconnected access provides scalability


Dot Net
Applications
WEB
WINDOWS
CONSOLE
Any Others...
ADO.NET
System.Data.SqlClient
System.Data.OracleClient
System.Data.OleDb
System.Data.Odbc
Connection
Command
DataReader
DataAdapter
DataSet
DATABASES
SQL Server
Oracle
OleDb
Odbc
Other Data Sources

The ADO.NET Object Model

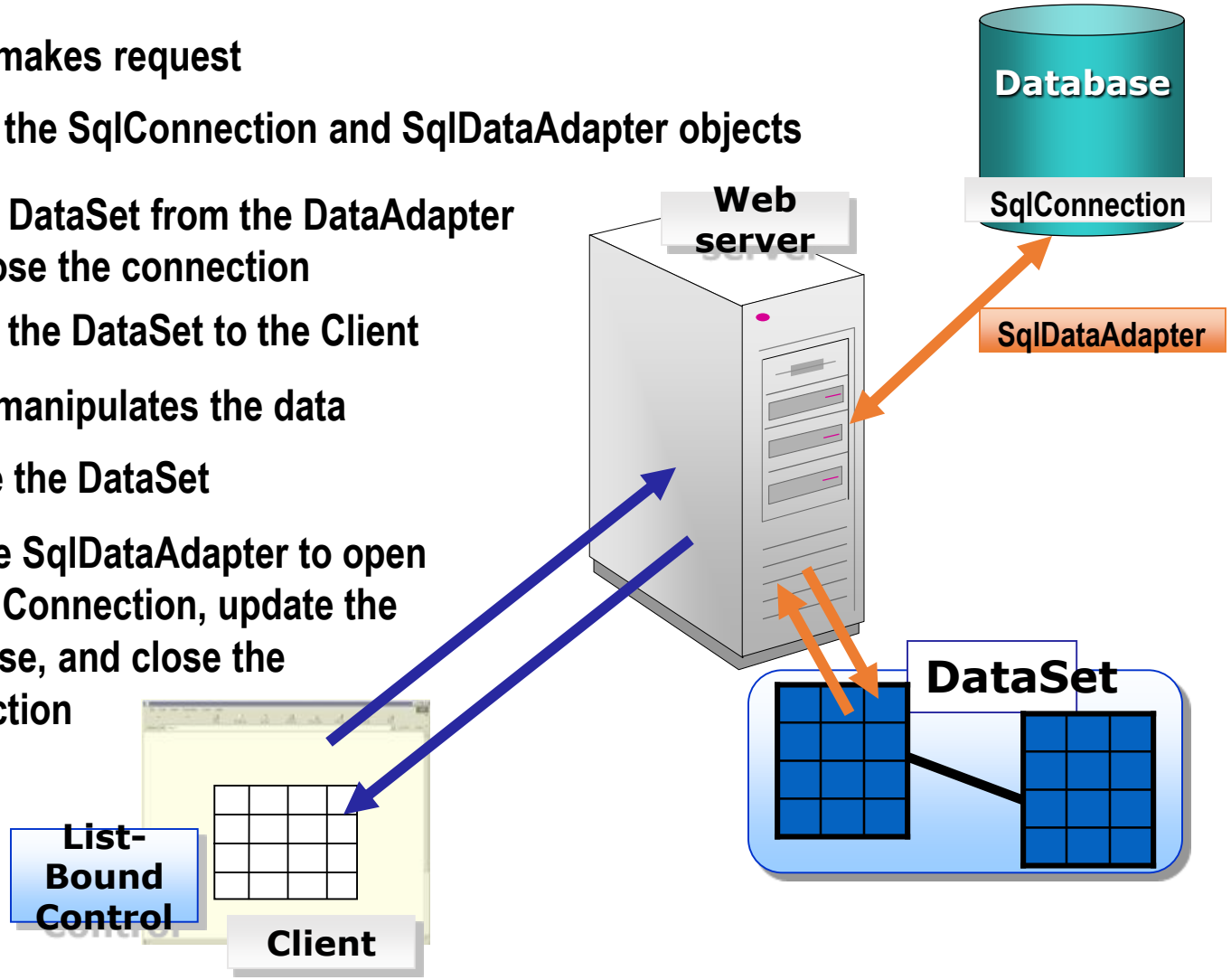


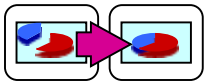
What is a Dataset?



Accessing Data with ADO.NET

1. Client makes request
2. Create the SqlConnection and SqlDataAdapter objects
3. Fill the DataSet from the DataAdapter and close the connection
4. Return the DataSet to the Client
5. Client manipulates the data
6. Update the DataSet
7. Use the SqlDataAdapter to open the SqlConnection, update the database, and close the connection





Data Providers

- MS SQL Server 7.0+
- Oracle
- OLE DB (old SQL & Access- Jet 4.0)
- Open Database Connectivity (ODBC)- earlier Visual Studio, Access Driver, ODBC for Oracle

* Version 1.0 does not include ODBC

4 Core Classes of ADO.NET

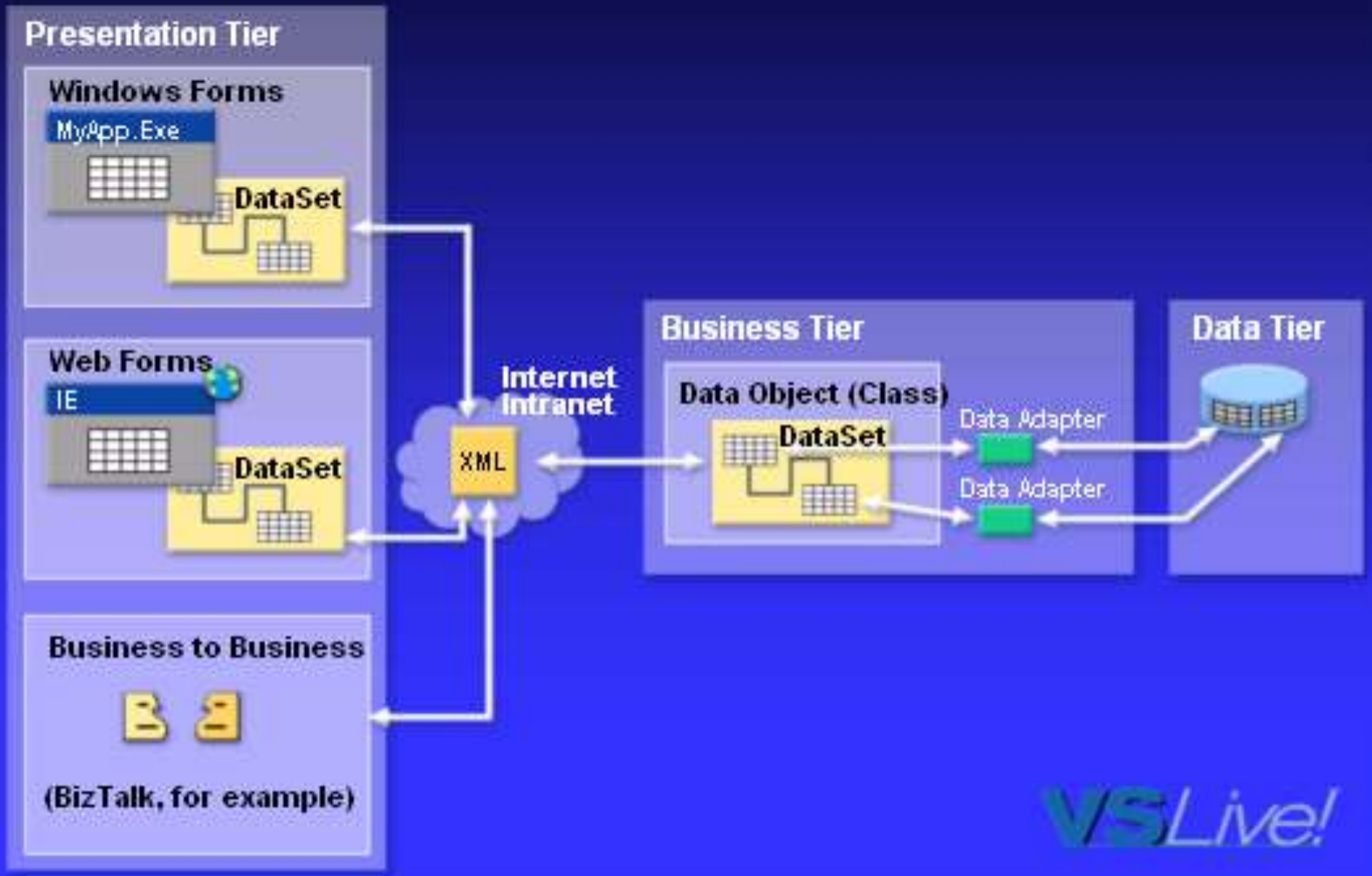
- Connection-Connect to database
- Command-SQL statement to retrieve data
- DataReader-Sequential access to the data source
- DataAdapter-Populate a dataset

  & Update the database

Ot  Terms

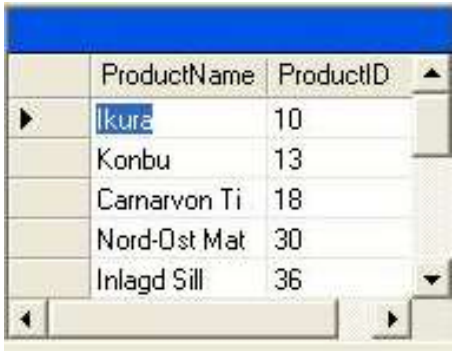
- Fill : The OleDbDataAdapter method *Fill* retrieves information from the database associated with OleDbConnection and places this information in the DataSet.
- DataGrid: A *DataGrid* is the area which will be filled with data from the database. The DataGrid method SetDataBinding binds a DataGrid to a data source.

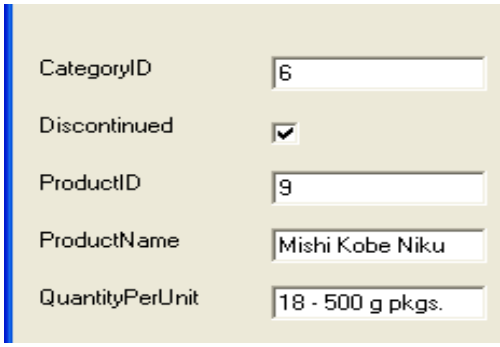
ADO .NET Architecture



Choices?

- Using ADO.NET we can either display information in a:
  - DataGrid
  - Individual Controls





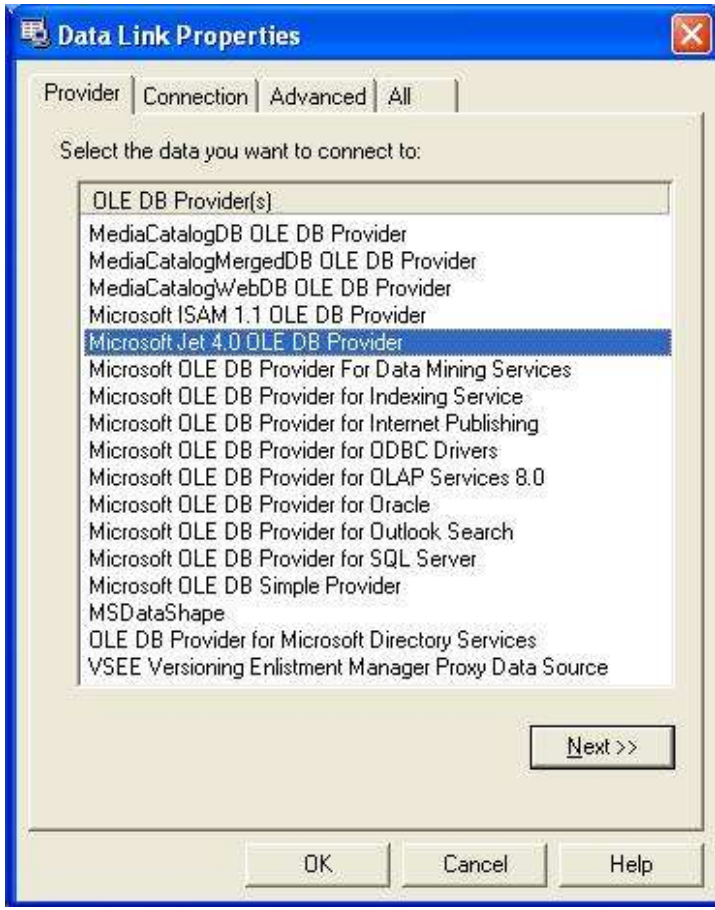
Let’s Connect to a Database

Adding a Connection

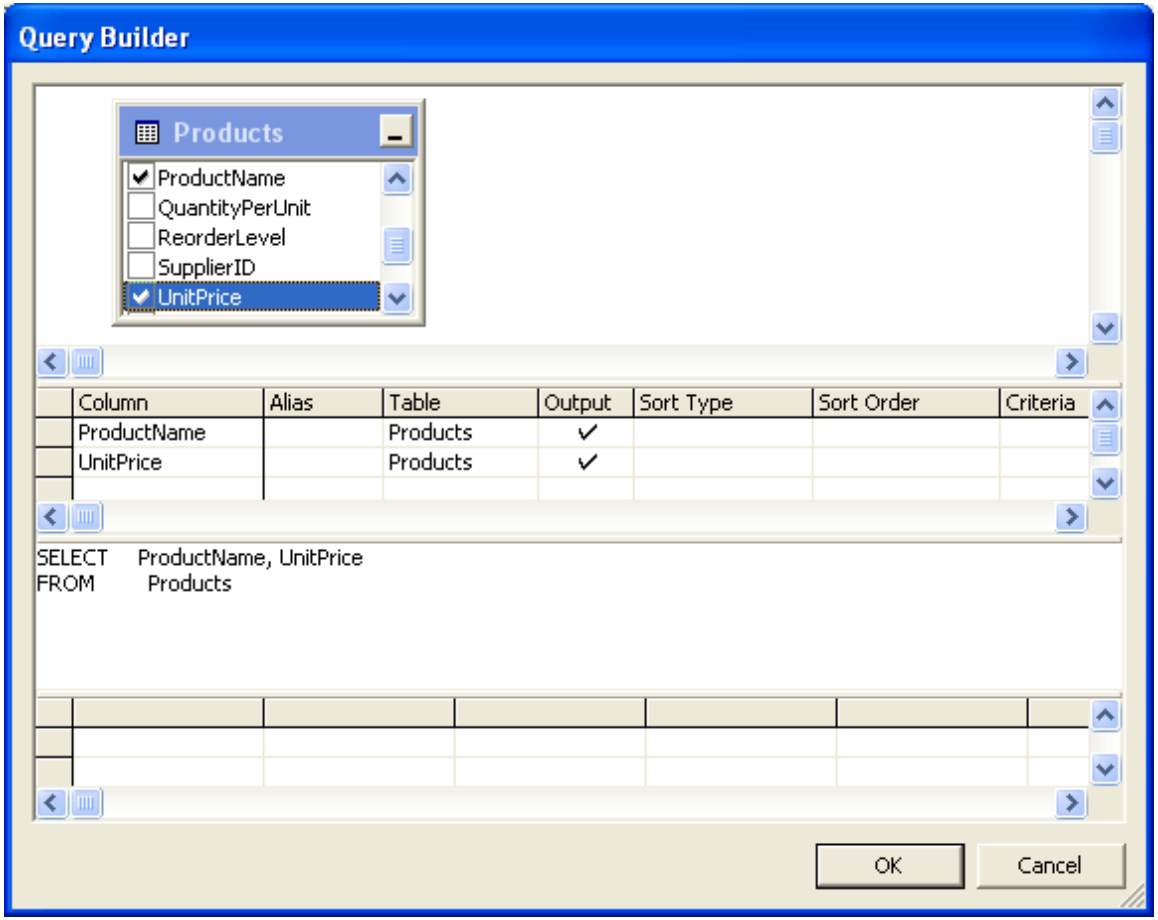
The *ADD CONNECTION* option is built into ADO.NET to create a database connection in the *DATA LINK PROPERTIES* window. The *DATA ADAPTER CONFIGURATION WIZARD* is used to set up an OleDbDataAdapter which generates queries to the connected database.

Connecting to the Database

- Dragging an OleDbDataAdapter from the Toolbox to the form of displays the Data Adapter Configuration Wizard.
- Clicking Next on the welcome screen displays the Choose Your Data Connection window.
- Clicking the New Connection button pops up the Data Link Properties form. Click the Provider tab, choose Microsoft Jet 4.0 OLE DB Provider



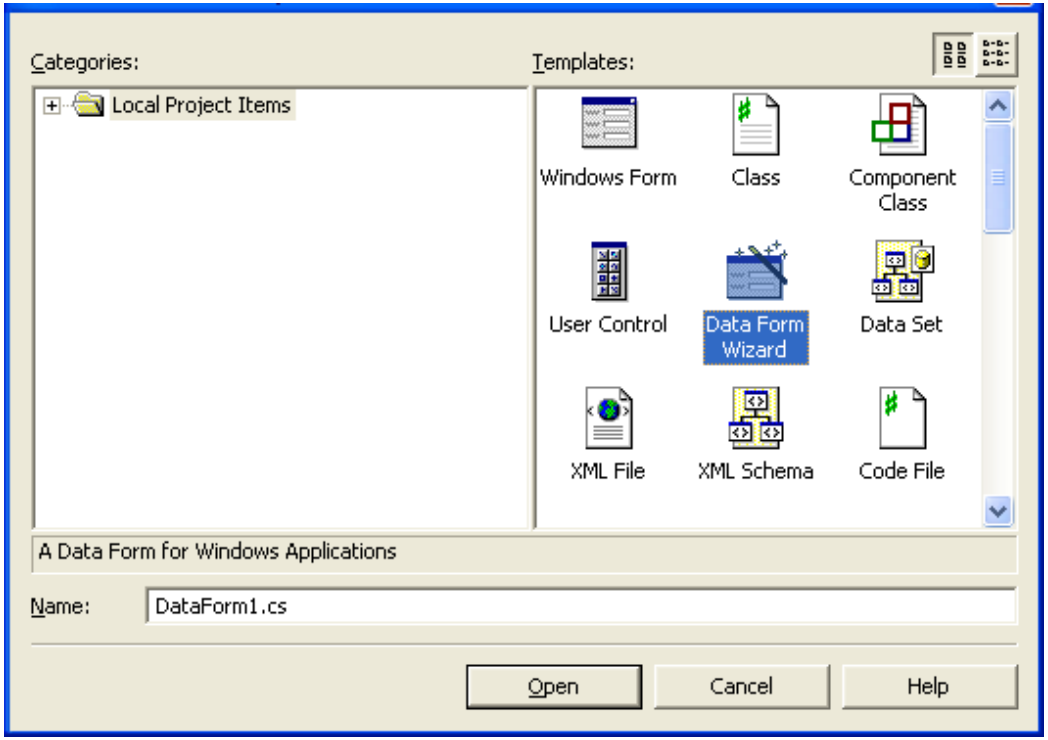
SQL Commands:Creating a Query



Time to try it!

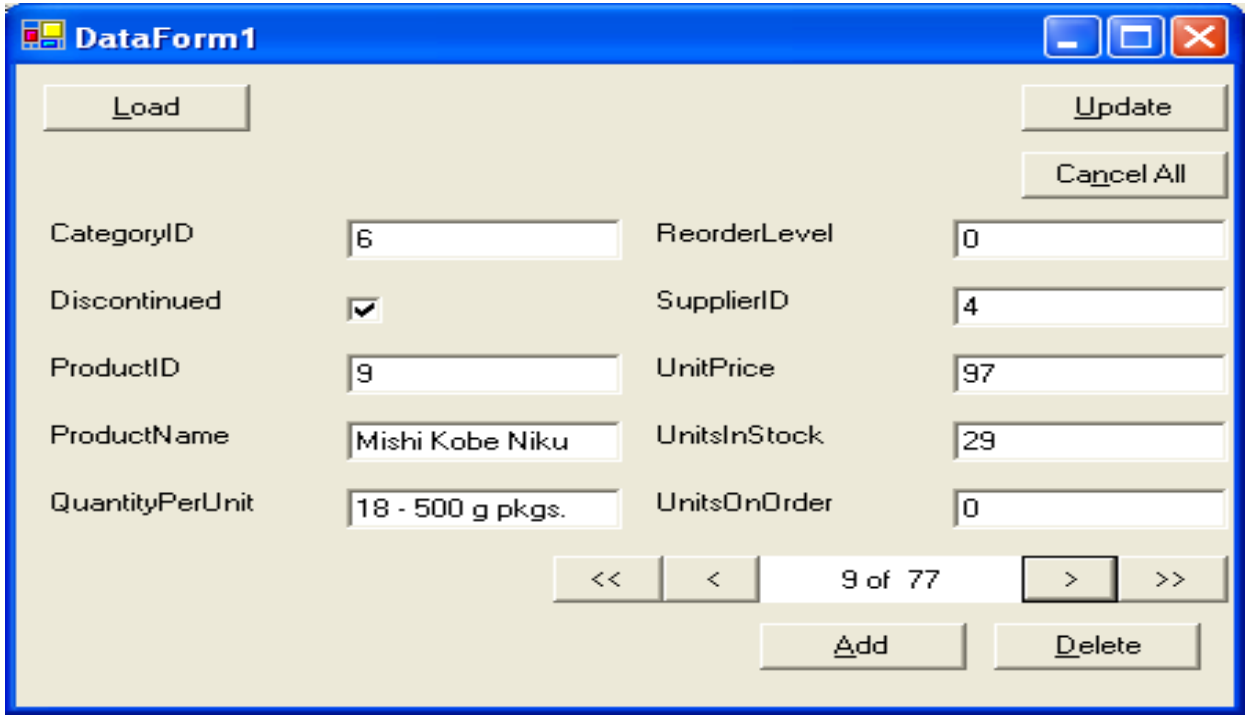
Northwind Database

Using a Data Form Wizard



Choosing Tables

Fill the Form



Using Namespaces

- Use the Imports or using statement to import namespaces

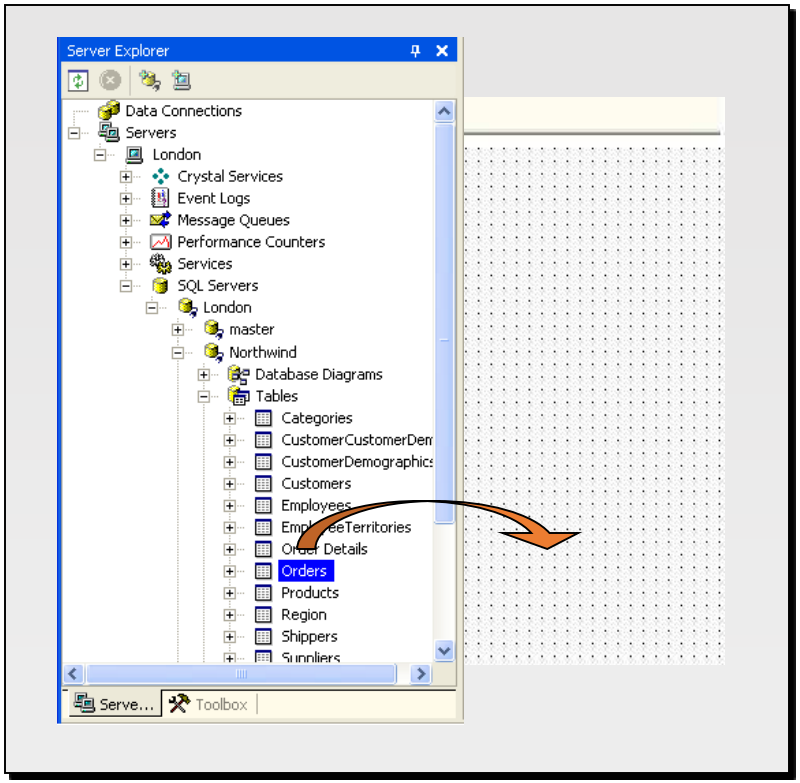
```
Imports System.Data
Imports System.Data.SqlClient
```

```
using System.Data;
using System.Data.SqlClient;
```

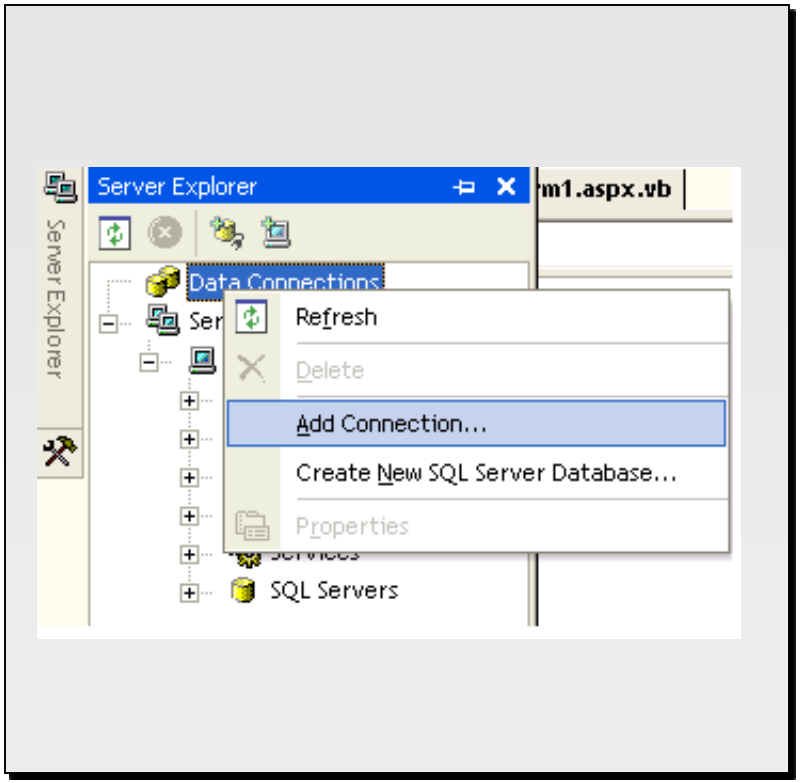
- Namespaces used with ADO.NET include:
  - **System.Data**
  - **System.Data.SqlClient**
  - **System.Data.OleDb**

Using Server Explorer to Generate a Connection

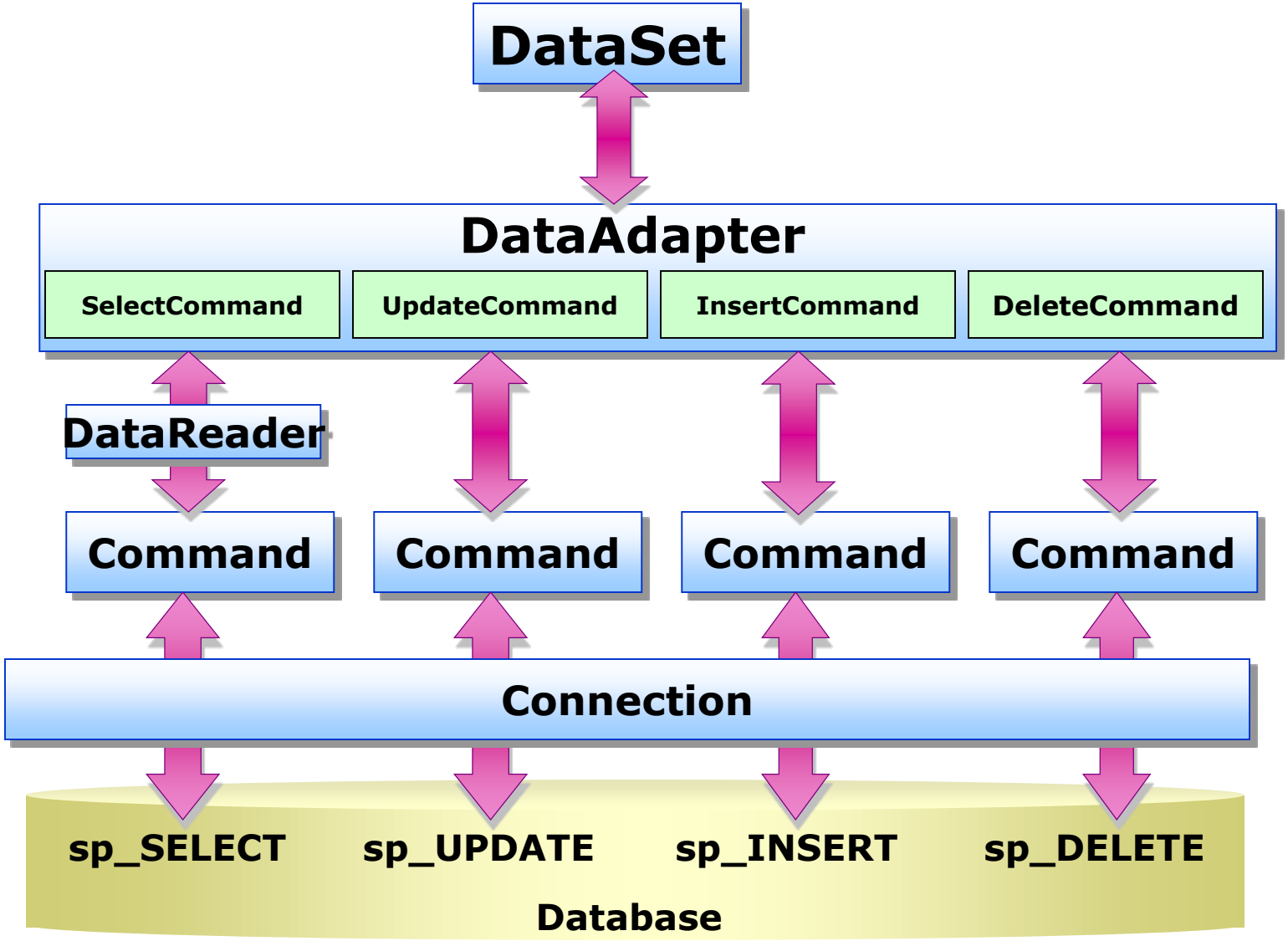
- Create a new data connection by dragging a Table from Server Explorer



- **Create a new data connection using the Data Links dialog box**



The DataAdapter Object Model



Creating a DataAdapter

- **Store the query in a DataAdapter**

```
Dim da As New SqlDataAdapter _
        ("select * from Authors", conn)
```

```
SqlDataAdapter da = new SqlDataAdapter
        ("select * from Authors",conn);
```

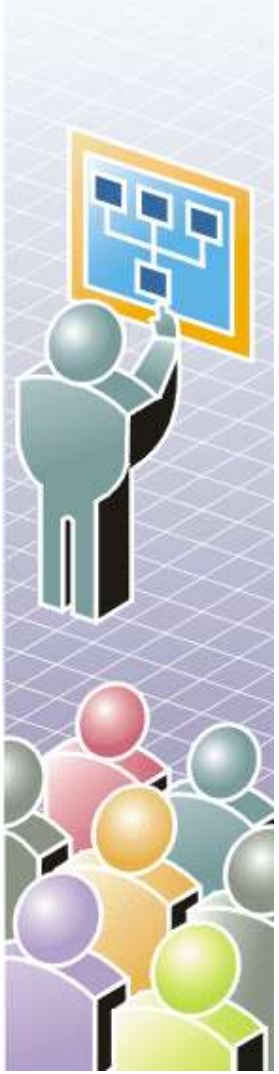
- **The DataAdapter constructor sets the SelectCommand property**

```
da.SelectCommand.CommandText
da.SelectCommand.Connection
```

```
da.SelectCommand.CommandText;
da.SelectCommand.Connection;
```

- **Set the InsertCommand, UpdateCommand, and DeleteCommand properties if needed**

Demonstration: Connecting to a Database



- Expand Server Explorer to a table in a SQL Server database
- Drag and Drop Data Access

Generating a DataSet

- You can generate a DataSet…
  - …through the UI…
    - Creates a **DataSet** that allows you to access data as an object
  - …or through code…

- and then fill…

```
Dim ds As New DataSet()
```

```
DataSet ds = new DataSet();
```

```
DataAdapter1.Fill(ds)
DataAdapter2.Fill(ds)
```

```
DataAdapter1.Fill(ds);
DataAdapter2.Fill(ds);
```

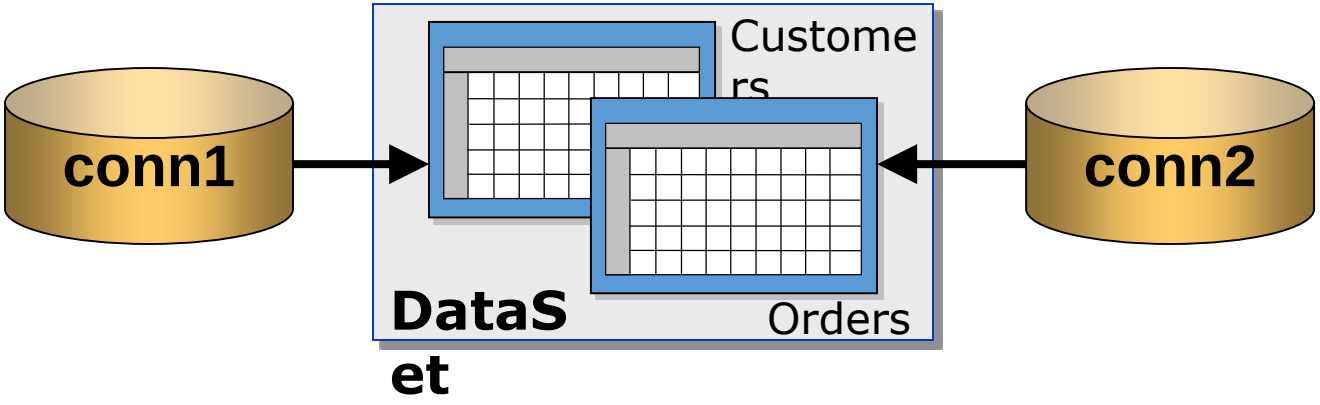
Storing Multiple Tables

- **Add the first table**

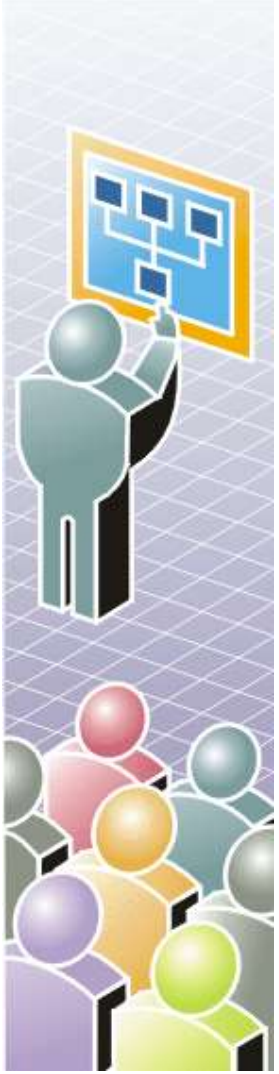
```
daCustomers = New SqlDataAdapter _
   ("select * from Customers", conn1)
daCustomers.Fill(ds, "Customers")
```

- **Add the subsequent table(s)**

```
daOrders = New SqlDataAdapter _
   ("select * from Orders", conn2)
daOrders.Fill(ds, "Orders")
```



Demonstration: Generating a DataSet



- Create a typed DataSet from a DataAdapter
- Add a second DataTable from a different DataAdapter
- Show the schema of DataSet

What are List-Bound Controls?

- **Controls that connect to a data source and display the data**
- **List-bound controls include the following:**
  - **DropDownList**
  - **ListBox**
  - **CheckBoxList**
  - **RadioButtonList**
  - **DataGrid**
  - **DataList**
  - **Repeater**

Displaying DataSet Data in List-Bound Controls

Property	Description
DataSource	▪ The **DataSet** containing the data
DataMember	▪ The DataTable in the **DataSet**
DataTextField	▪ The field in the **DataTable** that is displayed
DataValueField	▪ The field in the **DataTable** that becomes the value of the selected item in the list

- Fill the DataSet, then call the DataBind method

```
DataAdapter1.Fill(ds)
lstEmployees.DataBind()
```

```
DataAdapter1.Fill(ds);
lstEmployees.DataBind();
```