

Crystal Reports®

For Visual Studio .NET®

**Reporting Off ADO.NET Datasets**

---

© 2001 Crystal Decisions, Inc. Crystal Decisions, Crystal Reports, and the Crystal Decisions logo are registered trademarks or trademarks of Crystal Decisions, Inc. Microsoft and Visual Studio are registered trademarks of Microsoft Corporation in the U.S. and/or other countries. All other trademarks are the property of their respective owners.

Version 3.0

Crystal Decisions, Inc.  
895 Emerson Street  
Palo Alto, CA 94301

# Reporting Off ADO.NET Datasets

## Introduction

Crystal Reports for Visual Studio .NET supports reports that access ADO.NET datasets.

You can create an ADO.NET dataset from a variety of sources. Whatever the source is, before you can report off an ADO.NET dataset, you must first generate an object for the dataset. Subsequently, you can use a Report Expert in Crystal Reports for Visual Studio .NET to create a new report based on the data description provided by the dataset object.

A dataset object generated with the Visual Studio ADO.NET Dataset Designer contains only the data description, and not the actual data. Consequently, when working with a report connected to such a dataset object, you cannot browse field data in the Crystal Report Designer at design time.

To have the report display the actual data at runtime, you must first push the data into the dataset object using the ADO.NET Object Model, and then pass the populated dataset to the Report Engine using the Report Engine Object Model. You must then bind the report to the corresponding Web or Windows Forms Viewer before building and running the application.

### **In this walkthrough, you will create a report in Crystal Reports for Visual Studio .NET and report off an Access database via ADO.NET:**

1. Create a Web Form with a Web Forms Viewer.
2. Use the ADO.NET Dataset Designer to generate an ADO.NET dataset object from an Access database.
3. Use the Standard Report Expert in Crystal Reports for Visual Studio .NET to create a new report in the same project based on the data description provided by that dataset object.
4. Use the ADO.NET Object Model to push the data into the dataset object.
5. Use the Report Engine Object Model to pass the populated dataset object to the Report Engine, so that the report can display the actual data at runtime.
6. Bind the report to a Web Forms Viewer by report object in the application source file.

**Note** You can also bind the report to a Windows Forms Viewer and host it in a Windows project. For more information, refer to the online help for Crystal Reports for Visual Studio .NET.

7. Run the Web application.

## Creating a Web Form with a Web Forms Viewer

Since the different languages in Visual Studio .NET (including Visual C#, Managed C++, and Visual Basic .NET) all support Web Forms programming, you can choose the language you prefer to develop your Web Forms page. However, for the purpose of this walkthrough, you will be creating a Web project in Visual C#.

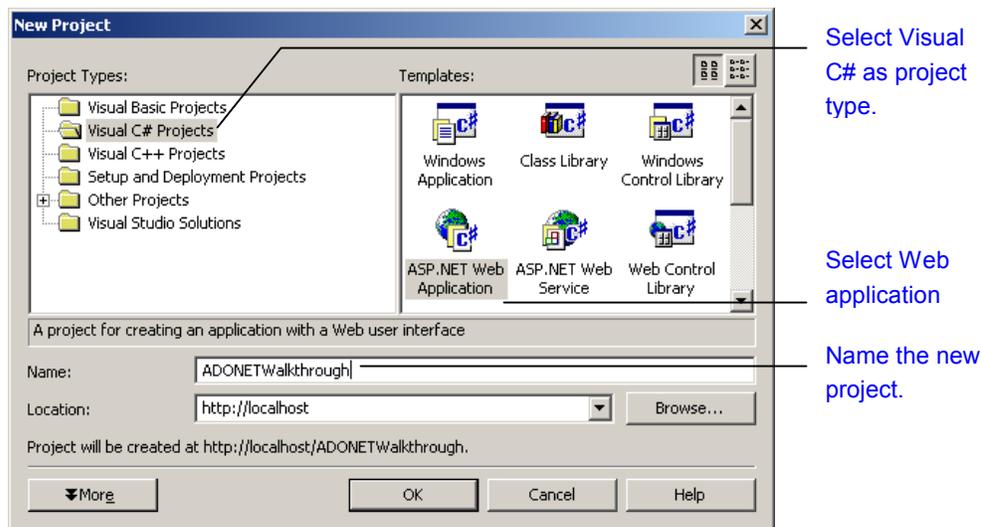
1. In Visual Studio .NET, select **File | New | Project**. This invokes the **New Project** dialog box.

2. In the **New Project** dialog box, select **Visual C# Projects** in the left pane.
3. Select **ASP.NET Web Application** in the right pane.

In general, you may choose to create your project on any Web server of your choice. For the purpose of this example, you will create the project in the default project location that is the local IIS Web server.

**Note** You can refer to the local IIS Web server by its machine name or by the name "localhost". For example, if the local host machine name is "angelskier", you can refer to it as http://angelskier or http://localhost. For the rest of this walkthrough, you will refer to the local host machine as http://localhost.

4. Enter " http://localhost/ADONETWalkthrough " in the **Location** field. Click OK. This creates the Web Application project, ADONETWalkthrough, at http://localhost/ADONETWalkthrough. (By default, in Visual Studio .NET, Web Application projects created on http://localhost is placed under c:\Inetpub\wwwroot, assuming c: is the Web server installation drive.)



After you click **OK** in the **New Project** dialog box, Visual Studio creates a Web Forms page (with the default name "WebForm1.aspx") in the project ADONETWalkthrough.

The window for the Web Forms page consists of two tabs: **Design** and **HTML**. The **Design** tab displays the Web Form visually, and supports programming using the Visual Studio Integrated Development Environment (IDE). The **HTML** tab contains the HTML source that supports ASP.NET.

Upon creating the Web Forms page, WebForm1.aspx, Visual Studio opens it in the **Design** tab.

With the Design tab of WebForm1.aspx active, the **Web Forms** tab of the Visual Studio Toolbox displays a set of controls for the Web Form, including the Web Forms Viewer control (labeled as **CrystalReportViewer** in the **Web Forms** tab).

You can now drag and drop the Web Forms Viewer control from the Toolbox into the Design tab of WebForm1.aspx.

## Generating an ADO.NET Dataset Object

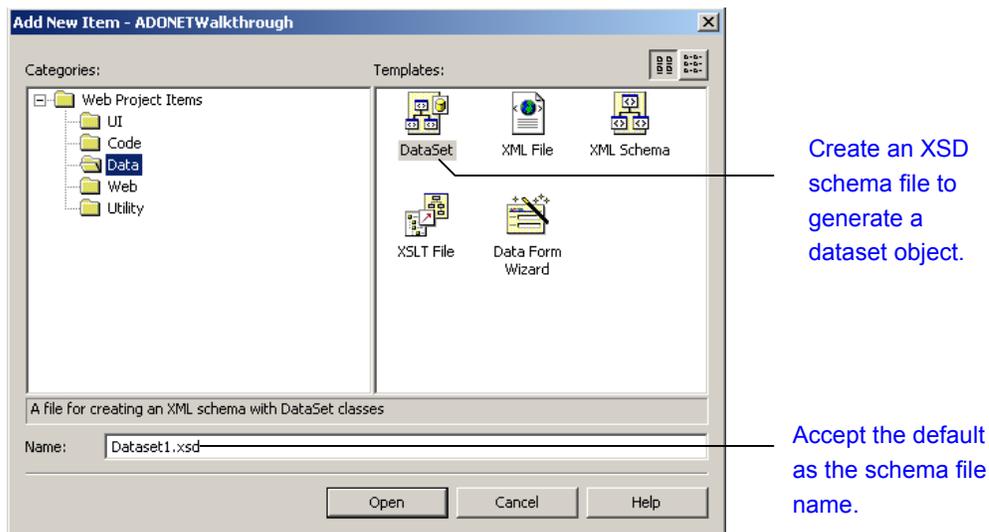
An ADO.NET dataset object is a collection of dataset classes created in memory. You can create an ADO.NET dataset object from numerous sources, including a database.

### Creating a Dataset Object from a Database

You can create an ADO.NET dataset from various data sources (including Access, Oracle, and SQL Server databases) by using the Visual Studio ADO.NET Dataset Designer. The ADO.NET Dataset Designer shows a graphic view of the database with its tables, its fields, and their corresponding types. You can drag tables from the Server Explorer to a schema file in the DataSet tab of the Designer. In this walkthrough, you will create a schema file for the project, ADONETWalkthrough, based on the Access sample database, xtreme.mdb, which has been automatically installed during setup at c:\Program Files\Microsoft Visual Studio .NET\Crystal Reports\Samples\Database (assuming c: is the drive where Visual Studio .NET has been installed).

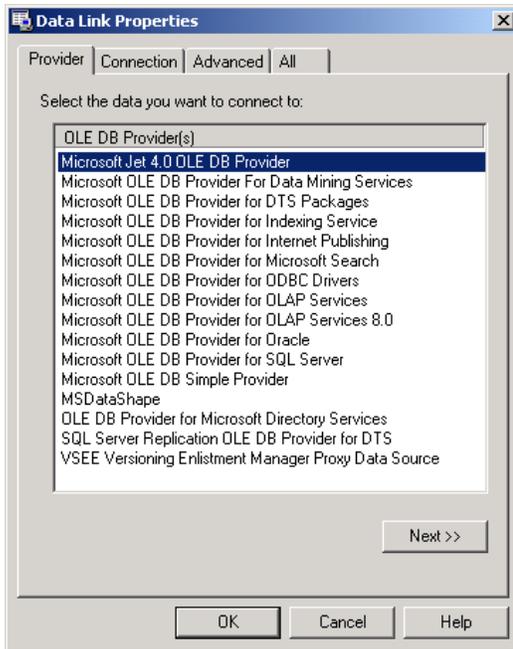
#### To create a dataset object from xtreme.mdb

1. Create a new schema file in ADONETWalkthrough:
  - a. In the **Solution Explorer**, right-click the project name, point to **Add**, and click **Add New Item**.
  - b. In the **Categories** area of the **Add New Item** dialog box, expand **Web Project Items** and select **Data**.
  - c. In the **Templates** area, select **DataSet**.
  - d. Accept the default name **Dataset1.xsd**.



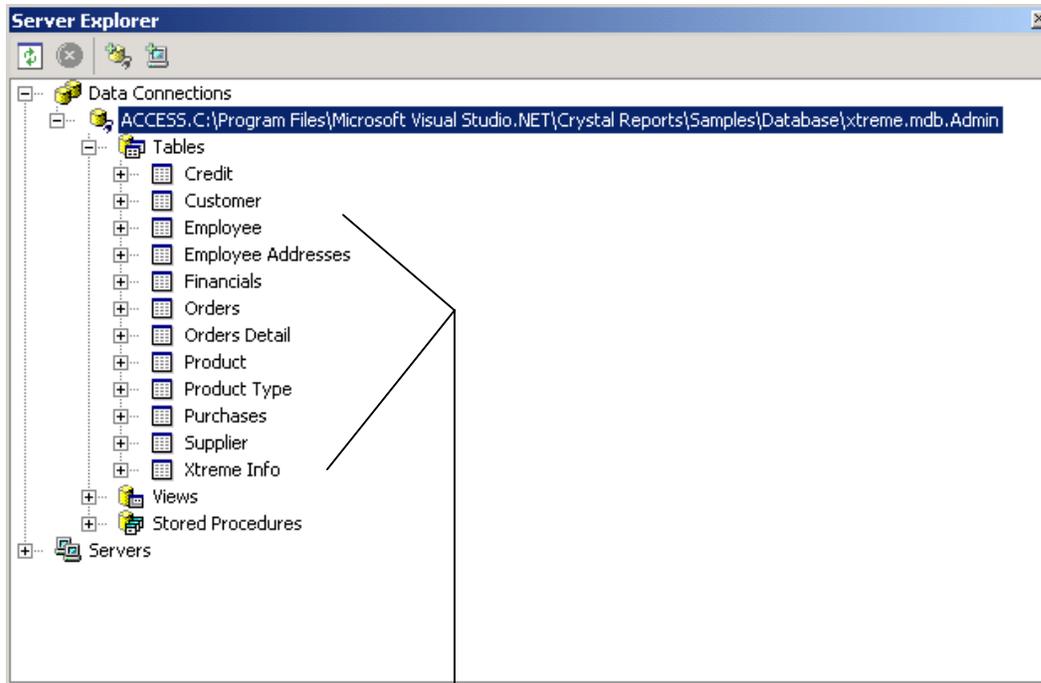
This creates a new schema file (Dataset1.xsd) that will be used to generate a strongly typed dataset. The schema file will be displayed in the ADO.NET Dataset Designer.

2. Specify where the database is located:
  - a. Select **Server Explorer** from the **View** menu if the Server Explorer has not yet been opened.
  - b. In the Server Explorer, right-click **Data Connections** and select **Add Connection**. This invokes the Data Link Properties dialog box.
  - c. In the **Data Link Properties** dialog box, click the **Provider** tab and select a provider (for example, Microsoft Jet 4.0 OLE DB Provider).



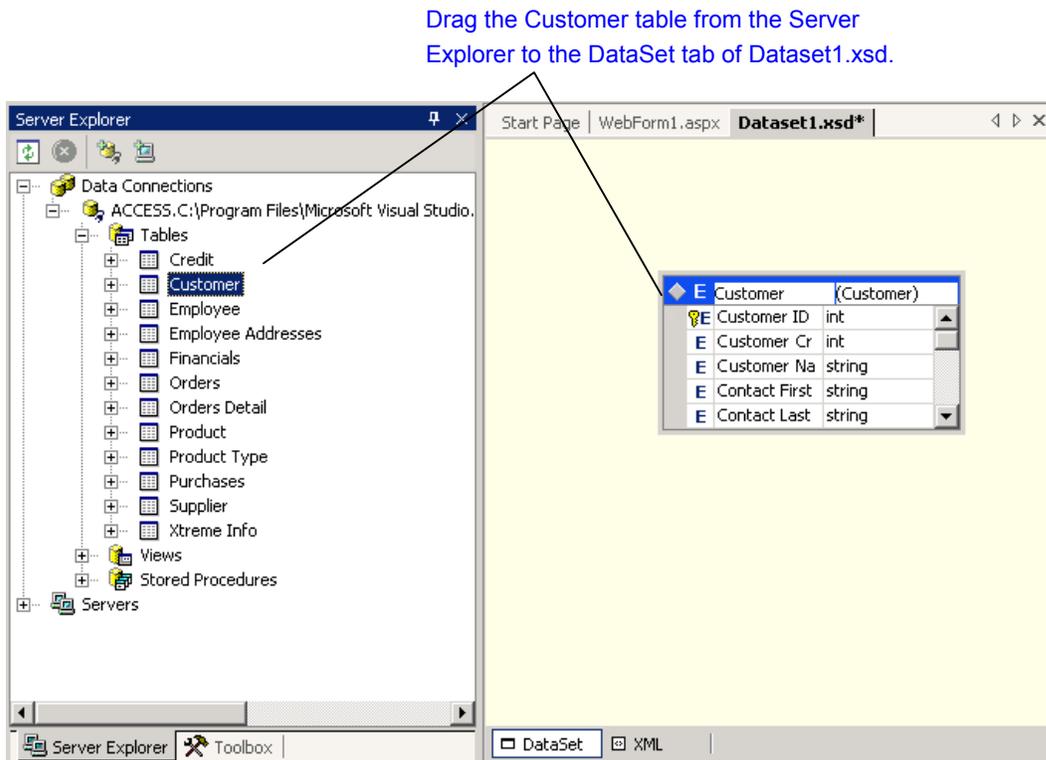
- d. Click the **Connection** tab and specify the location of your database. Enter any server and logon information where necessary. For the purpose of this walkthrough, you will access the sample database xtreme.mdb which has been installed by default on c:\Program Files\Microsoft Visual Studio .NET\Crystal Reports\Samples\Database.
- e. Click **OK**.

Your database, its tables, and its fields now appear in the Server Explorer under the Data Connections node.



List of tables and fields in the database xtreme.mdb.

3. In the Solution Explorer, double-click **Dataset1.xsd**, if it is not already the active view. Dataset1.xsd should now be displayed in the **DataSet** tab of the ADO.NET Dataset Designer.
4. To build a schema for your dataset, drag the desired tables from the **Server Explorer** to the **DataSet** tab of Dataset1.xsd. For the purpose of this walkthrough, select and drag the Customer table from the Server Explorer to the **DataSet** tab.



5. Right-click **References** in the Solution Explorer, and add CrystalDecisions.CrystalReports.Engine as reference.
6. Click the **Save Dataset1.xsd** icon to save the Dataset1.xsd file.
7. On the **Build** menu, click **Build** to generate the dataset object for the project. In this scenario, the dataset object will contain the data description for the Customer table.

**Note** The generated dataset object contains only a description of the database based on the schema in Dataset1.xsd; it does not contain the actual data.

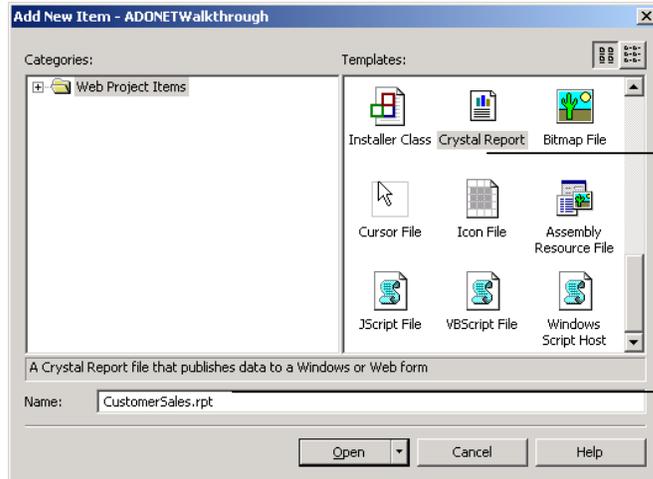
## Connecting a Report to an ADO.NET Dataset Object

An ADO.NET dataset object provides a data description from which you can add tables to your Crystal report. Use one of Crystal's Report Experts (e.g., Standard Report Expert) to add tables from the ADO.NET dataset object.

Below, you will create a new report in ADONETWalkthrough, and use the Standard Report Expert to connect to the dataset object that has been created above based on the Customer table in xtreme.mdb:

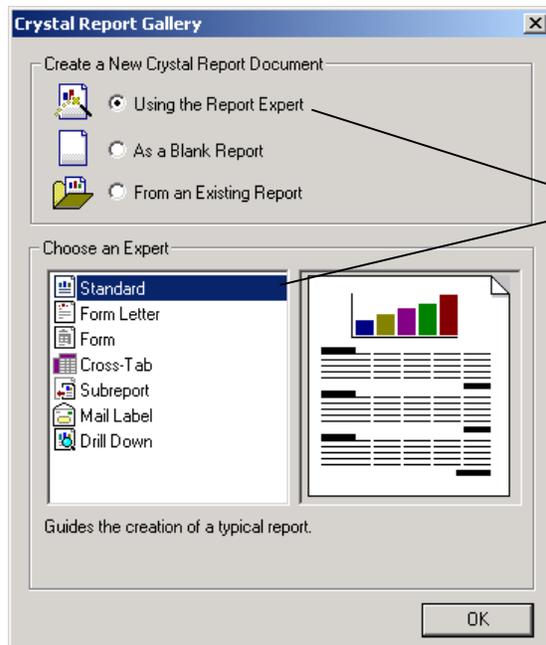
1. In the Solution Explorer, right-click ADONETWalkthrough, and select **Add | Add New Item** from the shortcut menu. This invokes the **Add New Item – ADONETWalkthrough** dialog box.

- In the **Add New Item – ADONETWalkthrough** dialog box, under **Categories**, with **Utility** highlighted in the left pane, select **Crystal Report** in the right pane. Enter "CustomerSales.rpt" in the **Name** field. Click the **Open** button. This invokes the Crystal Report Designer that will help you create and design CustomerSales.rpt.



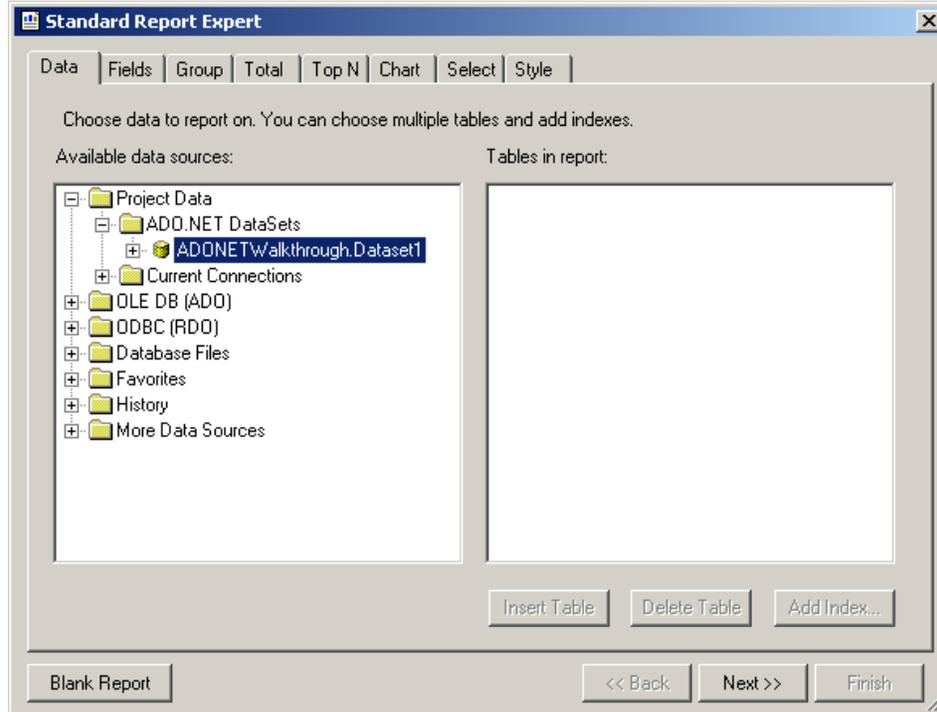
Add a new report item to the project.

- When you create a report, the Crystal Report Designer opens with the **Crystal Report Gallery**. Under **Create a New Crystal Report Document**, choose **Using the Report Expert**. Under **Choose an Expert**, choose **Standard**. This creates CustomerSales.rpt using the Standard Report Expert.



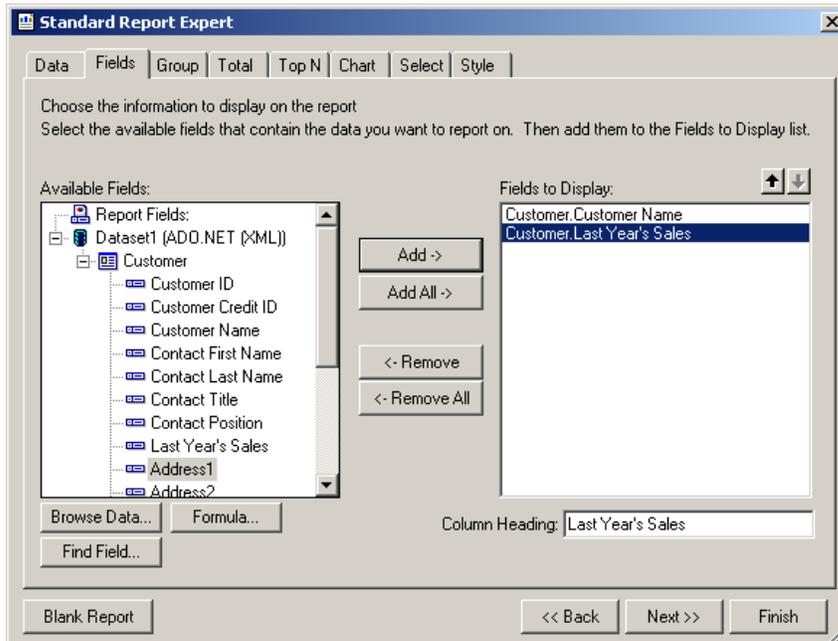
Choose the Standard Report Expert to create a report.

4. The Standard Report Expert opens, expand **Project Data | ADO.NET Datasets**.



5. Double-click the dataset object, **ADONETWalkthrough.Dataset1**. This displays the tables included in this dataset object; in this scenario, you will find the Customer table.
6. Add the Customer table to CustomerSales.rpt by highlighting it and clicking **Insert Table**. Then click **Next**.
7. In the **Fields** tab, under **Available Fields**, highlight the Customer Name field and click **Add**.

8. Similarly, add the Last Year's Sales field to CustomerSales.rpt.



9. Click **Finish**. The Crystal Report Designer displays CustomerSales.rpt.

## Setting Up an ADO.NET Report in a Viewer

Since a dataset object generated with the ADO.NET Dataset Designer contains only the data description, and not the actual data, your report cannot display any data. In order to have the report display the actual data, you should fill the dataset object with data before you bind the report to a Web Forms Viewer or a Windows Forms Viewer. You should do this in the corresponding source file for the Web or Windows Form.

## Setting Up an ADO.NET Report in a Web Forms Viewer

### To specify xtreme.mdb as the database and bind CustomerSales.rpt to a Web Forms Viewer

1. Highlight **WebForm1.aspx** in the Solution Explorer. Then click the **View Code** icon. This opens the source file for the Web Form, WebForm1.aspx.cs.
2. In WebForm1.aspx.cs, type at the beginning of the namespace definition:
 

```
using System.Data.OleDb;
```
3. At the beginning of the class definition for WebForm1, type:
 

```
public CustomerSales oRpt = null;
```

4. After the call to `base.OnInit(e)` in `OnInit()`, type:

```
oRpt = new CustomerSales ();

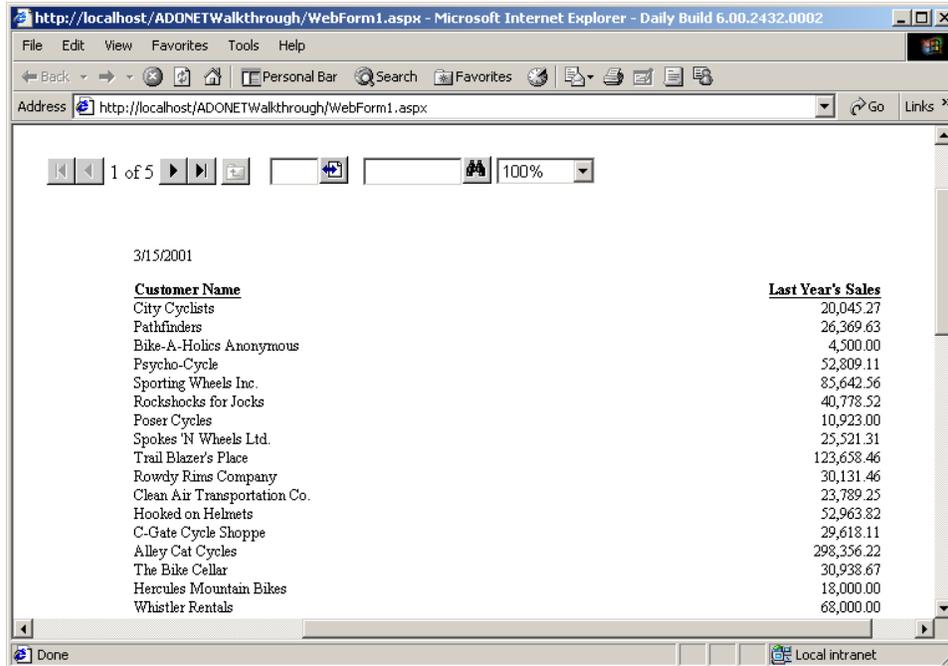
/* Use ADO.NET object model to set up connection information */
OleDbConnection oleConn = new OleDbConnection
("Provider=Microsoft.Jet.OLEDB.4.0;Data Source=C:\\Program
Files\\Microsoft Visual Studio .NET\\Crystal
Reports\\Samples\\Database\\xtreme.mdb");
OleDbDataAdapter oleAdapter = new OleDbDataAdapter("SELECT * FROM
Customer", oleConn);
Dataset1 dataSet = new Dataset1();

/* Connect to, fetch data and disconnect from database */
oleAdapter.Fill (dataSet, "Customer");

/* Use Report Engine object model to pass populated dataset
to report */
oRpt.SetDataSource (dataSet);

/* bind report object with data to the Web Forms Viewer */
CrystalReportViewer1.ReportSource = oRpt;
```

5. Select **Build** from the **Build** menu to compile the application. Then select **Start Without Debugging** from the **Debug** menu to run the application. The Web Form will run on your browser displaying CustomerSales.rpt showing Last Year's Sales for each Customer Name.



3/15/2001

<u>Customer Name</u>	<u>Last Year's Sales</u>
City Cyclists	20,045.27
Pathfinders	26,369.63
Bike-A-Holics Anonymous	4,500.00
Psycho-Cycle	52,809.11
Sporting Wheels Inc.	85,642.56
Rockshocks for Jocks	40,778.52
Poser Cycles	10,923.00
Spokes 'N Wheels Ltd.	25,521.31
Trail Blazer's Place	123,658.46
Rowdy Rims Company	30,131.46
Clean Air Transportation Co.	23,789.25
Hooked on Helmets	52,963.82
C-Gate Cycle Shoppe	29,618.11
Alley Cat Cycles	298,356.22
The Bike Cellar	30,938.67
Hercules Mountain Bikes	18,000.00
Whistler Rentals	68,000.00

**Tip** When specifying fields to select from a database, following the syntax for SQL statements, you must delimit field names with the backward single quotes and not the straight single quotes.

Also, the `Fill` method connects to the specified database, fetches the data, and disconnects from the database. If you want to add more than one table from the database to the report, use a SQL JOIN statement to join the tables together; then specify the single resultant table in the `Fill` method.