

Crystal Reports®

For Visual Studio .NET®

Designing and Viewing a Report in a Windows Application

© 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

Designing and Viewing a Report in a Windows Application

Introduction

Crystal Reports for Visual Studio .NET extends the powerful reporting capability of Crystal Reports on the Microsoft .NET platform. You can use the Crystal Report Designer in Visual Studio .NET to create a new report or modify an existing report. You can then keep the report on a local machine, or publish it as a Report Web Service on a Web server. Depending on whether you are developing a Windows or Web application, you can then host the report with either the Windows Forms Viewer or the Web Forms Viewer, respectively.

This paper explains how to do the following in Visual Studio .NET:

- create a new Crystal report in a project
- add an existing Crystal report to a project
- modify a Crystal report
- host and view a Crystal report using the Windows Forms Viewer

"Appendix A: Class Description for the Windows Forms Viewer Control" lists the set of objects, methods and properties for the Windows Forms Viewer. "Appendix B: Report Binding Methods for the Windows Forms Viewer" lists the different methods of binding a local report to a Windows Forms Viewer.

If you are interested in viewing Crystal reports using the Web Forms Viewer, please refer to the paper "Viewing a Report in a Web Application". For more information on Report Web Services, please refer to the paper "Exposing Reports as Web Services".

For more information and updates, please visit our Web site at:

<http://www.crystaldecisions.com/net>

Setting Up the Sample Database

When you install Visual Studio .NET, the sample database, xtreme.mdb, is automatically installed in C:\Program Files\Microsoft Visual Studio .NET\Crystal Reports\Samples\Database\. The reports used in this walkthrough, OrderByCustomer.rpt and World Sales Report.rpt, assume xtreme.mdb is at this location.

To change the location of xtreme.mdb

If you have installed xtreme.mdb at a location other than the default (i.e., C:\Program Files\Microsoft Visual Studio .NET\Crystal Reports\Samples\Database\), you should do a Set Location for World Sales Report.rpt to point the report to the actual location of the database.

1. In Visual Studio .NET, select **File | Open | File**. The **Open File** dialog box appears.
2. In the **Open File** dialog box, browse to C:\Program Files\Microsoft Visual Studio .NET\Crystal Reports\Samples\Reports\General Business, and specify the file World Sales Report.rpt. This opens World Sales Report.rpt in the Crystal Report Designer.

3. In the Crystal Report Designer, right-click in a report area. Choose **Database | Set Location**. The **Set Location** dialog box appears.
4. In the **Set Location** dialog box, click the down-arrow under **Current Data Source**. Select the data source assumed by World Sales Report.rpt, "C:\Program Files\Microsoft Visual Studio .NET\Crystal Reports\Samples\Database\xtreme.mdb". You will point this assumed location to the actual location of the database.
5. You may use any one of these database technologies to connect the report to the actual data source: OLE DB (ADO), ODBC (RDO), Database Files (i.e., using native drivers), or ADO.NET (XML) (which is under **More Data Sources**). For the purpose of this walkthrough, under **Replace with**, double-click **OLE DB (ADO)**. The **OLE DB (ADO)** dialog box appears.
Note If you are already connected to xtreme.mdb, go to step 9. Or, if you are connected to another database through OLE DB (ADO), select **Make New Connection**.
6. In the **OLE DB (ADO)** dialog box, highlight the OLD DB provider, **Microsoft Jet 4.0 OLEDB Provider**. Then click the **Next** button.
7. Provide the necessary information to access the actual data source: for **Database Name**, click the square button adjacent to the entry box. Select xtreme.mdb according to where you have installed it. Click **Next**.
8. Update any advanced information for your data source, if necessary. Then click **Finish**. You will be returned to the **Set Location** dialog box.
9. Redirect the report to look for the database in the actual location: in the **Set Location** dialog box, under **Replace with**, highlight the actual location of the database. Click **Replace**. You will find this location updated under **Current Data Source**. Then click **Close**.
10. If you have modified fields in xtreme.mdb, right-click in a report area, point to **Database** and select **Verify Database**. A message titled **Verify Database** appears. Click **OK** to fix up unmapped fields in the database.
11. Choose **File | Save World Sales Report.rpt** to save the report with the updated database location.

Creating a Crystal Report

In general, you can create a Crystal report as part of a Visual C#, Managed C++ or Visual Basic project in Visual Studio .NET. In this walkthrough, you will be creating reports in Visual C# projects.

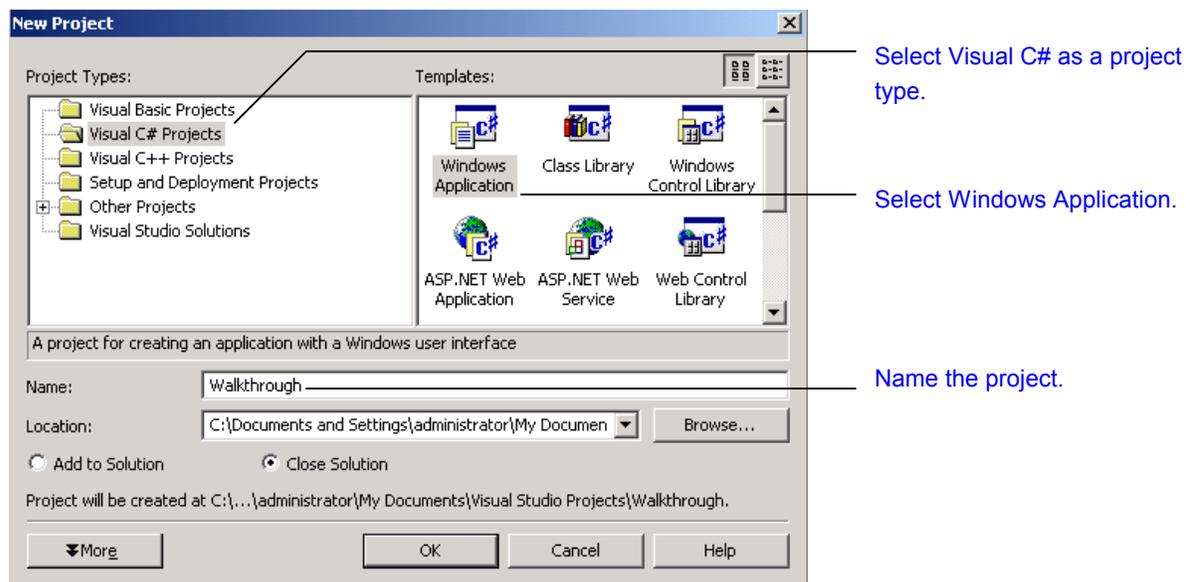
The following scenario illustrates how to create a Crystal report, OrderByCustomer.rpt, in a Visual C# project named Walkthrough. For simplicity, the report accesses a sample database via ActiveX Data Object (ADO), and displays orders grouped according to the customer.

Creating a Crystal Report in a Visual C# Project

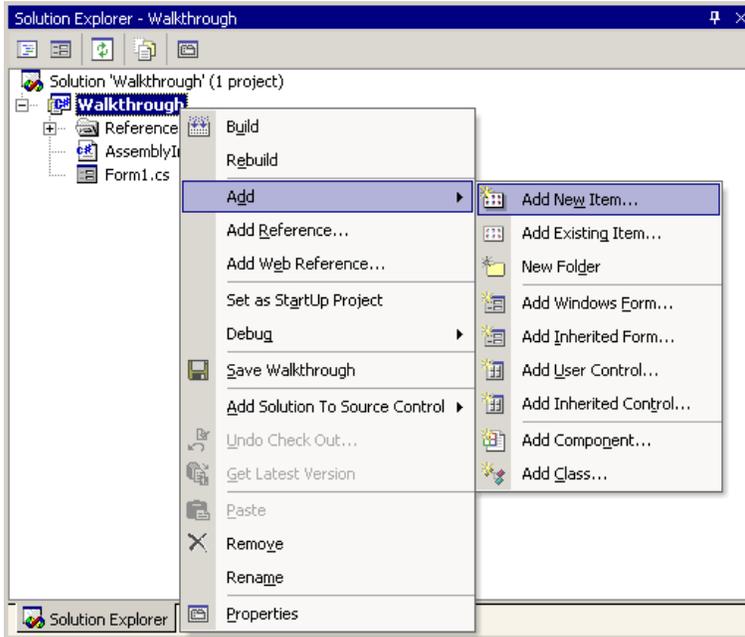
1. In Visual Studio .NET, select **File | New | Project**. The **New Project** dialog box appears.
2. In the **New Project** dialog box, select **Visual C# Projects** in the left pane.
3. Select **Windows Application** in the right pane.

You may log on as a regular user and choose to create your project in any folder of your choice. For the purpose of this example, you will create the project in the default project folder for the user logged on as Administrator, C:\Documents and Settings\Administrator\My Documents\Visual Studio Projects.

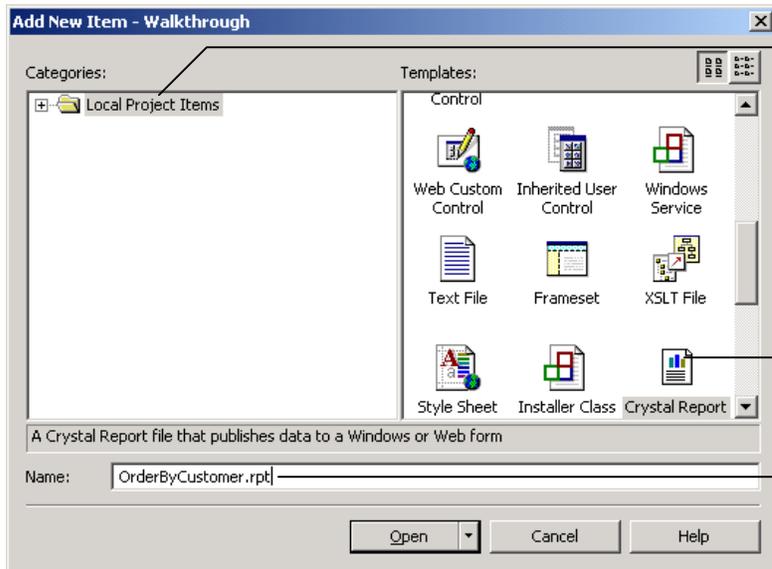
4. Enter "Walkthrough" in the **Name** field. The default project location is displayed in the **Location** field. Click **OK**. This creates the Visual C# Windows project, Walkthrough, in C:\Documents and Settings\Administrator\My Documents\Visual Studio Projects.



- In the Visual Studio Solution Explorer, right-click on the project Walkthrough. Select **Add | Add New Item** from the shortcut menu. The **Add New Item – Walkthrough** dialog box appears.



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



Select Local Project Items.

Select Crystal Report.

Name the new report.

Designing a Crystal Report

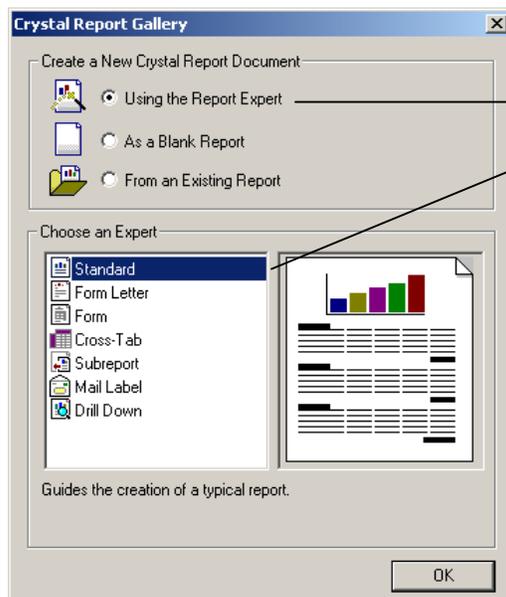
The Crystal Report Designer in Visual Studio .NET closely resembles the Designer of the Report Designer Component that runs on Visual Basic 6.0.

When you create a report, the Crystal Report Designer opens with the Crystal Report Gallery that offers the following options:

- use a Report Expert to guide you through the report creation process
- open a blank report
- create a new report from an existing report

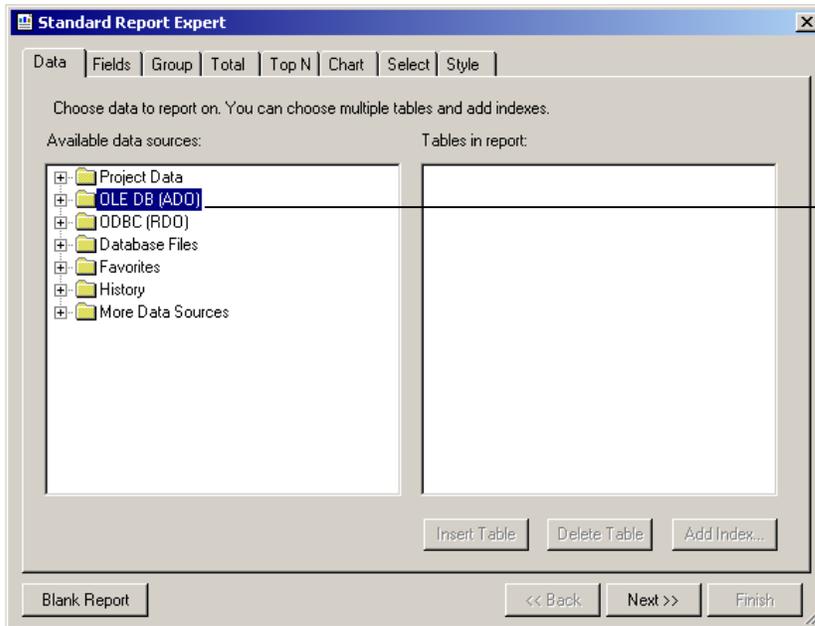
For the purpose of this example, you will use the Standard Report Expert to create a report titled OrderByCustomer. The OrderByCustomer.rpt file will access the xtreme sample database via ADO and display orders grouped according to each customer.

1. In the **Crystal Report Gallery**, under **Create a New Crystal Report Document**, select **Using the Report Expert**, if it is not already selected.
2. Under **Choose an Expert**, select **Standard**, if it is not already selected. Click **OK**. This invokes the Standard Report Expert.



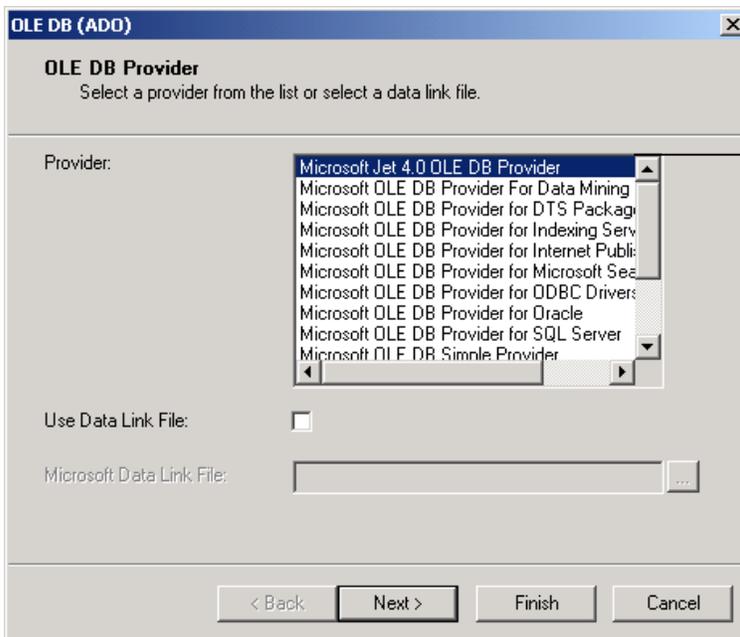
Choose the Standard Report Expert to create a report.

- In the **Data** tab of the **Standard Report Expert**, double-click **OLE DB (ADO)**. This invokes the **OLE DB (ADO)** dialog box.



Choose an ADO data source.

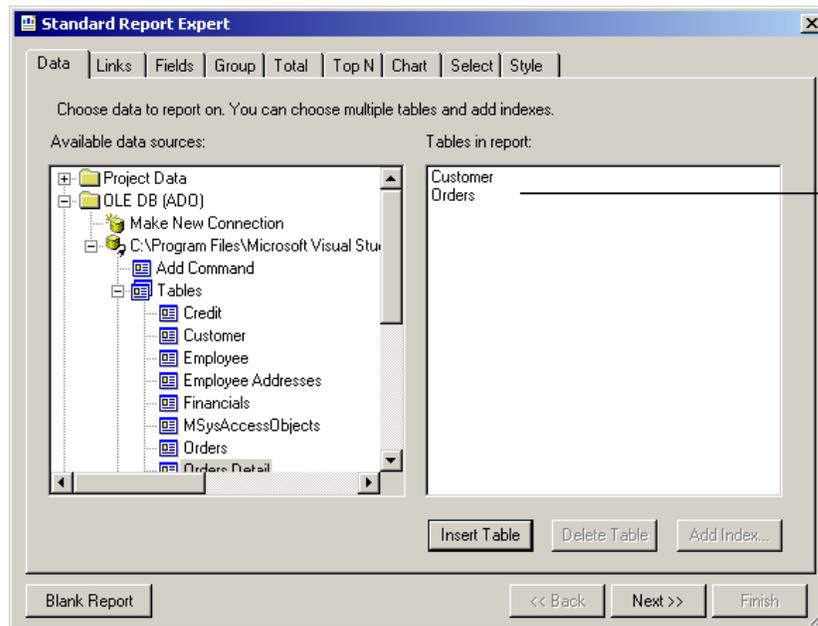
- In the **OLE DB (ADO)** dialog box, select **Microsoft Jet 4.0 OLE DB Provider**. Click **Next**.



Select MS Jet as the OLE DB provider.

- Click the square button adjacent to the entry field for **Database Name**.
- In the **Open** dialog box, select the xtreme database under C:\Program Files\Microsoft Visual Studio .NET\Crystal Reports\Samples\Database. Click **Open**.

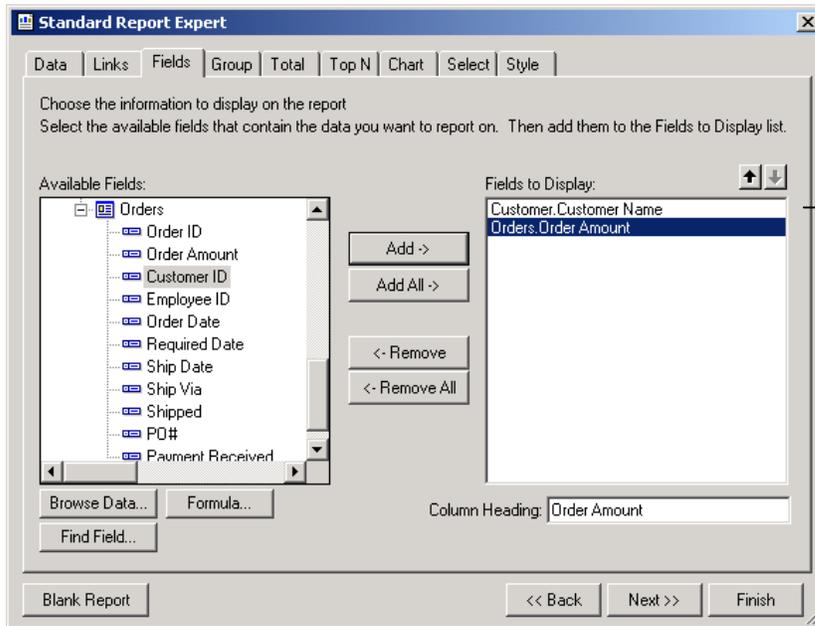
7. In the **OLE DB (ADO)** dialog box, accept **Access** as the **Database Type**. Enter the appropriate **User ID**. Click **Next**.
8. Accept the default advanced information, and click **Finish**.
9. The **Standard Report Expert** shows the new connection to C:\Program Files\Microsoft Visual Studio .NET\Crystal Reports\Samples\Database\xtreme.mdb, with the tables and fields in the database. Expand **Tables**, select the Customer table and click **Insert Table**. Do the same to add the Orders table. Click **Next**.



Add these tables to the report from the sample database.

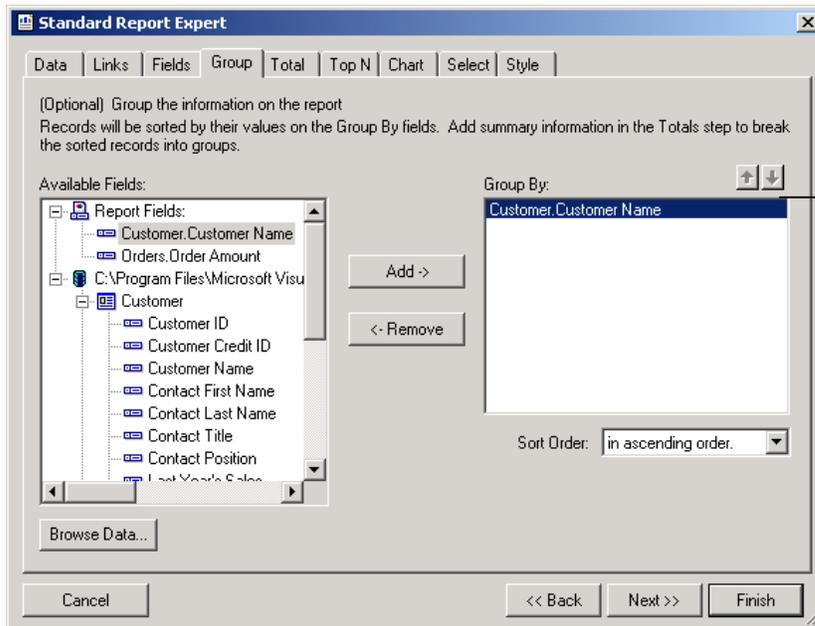
10. In the **Links** tab, click **Next**.

- In the **Fields** tab, add the Customer Name field from the Customer table and the Order Amount field from the Orders table. Click **Next**.



Select the fields to be included in the report.

- In the **Group** tab, under **Available Fields**, select the Customer Name field in the Customer table. Click the **Add** button. Next, click the **Finish** button. You have just created a report that displays orders grouped according to each customer.



Select a field to group data in the report.

- To save OrderByCustomer.rpt with Walkthrough, right-click on the project **Walkthrough** in the Solution Explorer and select **Save Walkthrough** from the shortcut menu.

14. Select **File | Close Solution** to close the solution.

Later in the section "Modifying a Crystal Report", you will further modify the report with the Crystal Report Designer.

Adding an Existing Crystal Report

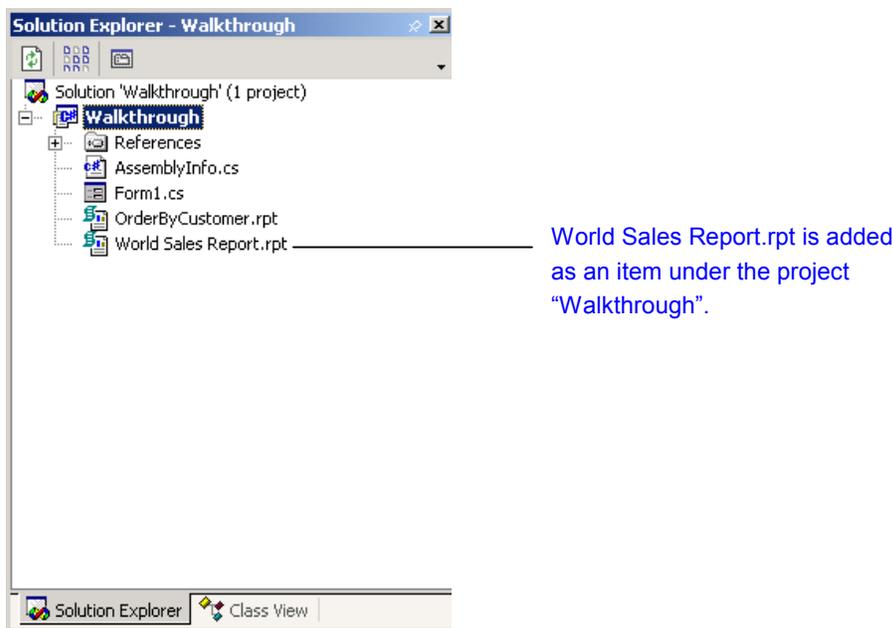
In general, you can add an existing Crystal report to a Visual C#, Managed C++ or Visual Basic project in Visual Studio .NET. In this walkthrough, you will be adding reports to Visual C# projects.

The following scenario illustrates how to add an existing Crystal report, World Sales Report.rpt, to the Visual C# project Walkthrough that has been created in the previous section "Creating a Crystal Report".

1. In Visual Studio .NET, select **File | Open | Project** to open Walkthrough.sln in the folder C:\Documents and Settings\Administrator\My Documents\Visual Studio Projects.
2. In the Visual Studio Solution Explorer, right-click the Visual C# project Walkthrough. Select **Add | Add Existing Item** from the shortcut menu. This invokes the **Add Existing Item – Walkthrough** dialog box.
3. In the **Add Existing Item – Walkthrough** dialog box, specify the location and name of the report you are adding. In this example, you are adding the sample report, World Sales Report.rpt, from the folder C:\Program Files\Microsoft Visual Studio .NET\Crystal Reports\Samples\Reports\General Business (assuming C: is your installation drive for Visual Studio .NET).

NOTE Select **All Files (*.*)** for **Files of Type** in order to display .rpt files in a folder.

World Sales Report.rpt will appear as an item under the Walkthrough project in the Solution Explorer.



To save World Sales Report.rpt with the project Walkthrough, right-click the project Walkthrough in the Solution Explorer and select **Save Walkthrough** from the shortcut menu.

Note World Sales Report.rpt assumes the database, xtreme.mdb, is at C:\Program Files\Microsoft Visual Studio .NET\Crystal Reports\Samples\Database. If you have set it up previously at a different location on your machine, please refer to the previous section "Setting Up the Sample Database" to have it set up properly.

If you would like to modify the report, double-click the report in the Solution Explorer to launch the Crystal Report Designer. For further information on modifying a report with the Crystal Report Designer, please refer to the section "Modifying a Crystal Report" below.

Modifying a Crystal Report

After opening a report in the Crystal Report Designer, you can further modify the report by manipulating its source of data, re-selecting and re-grouping data, and re-formatting its objects and layout. The Crystal Report Designer provides a user interface that closely integrates with the Visual Studio .NET IDE. It includes events that can be directly programmed from within Visual Studio.

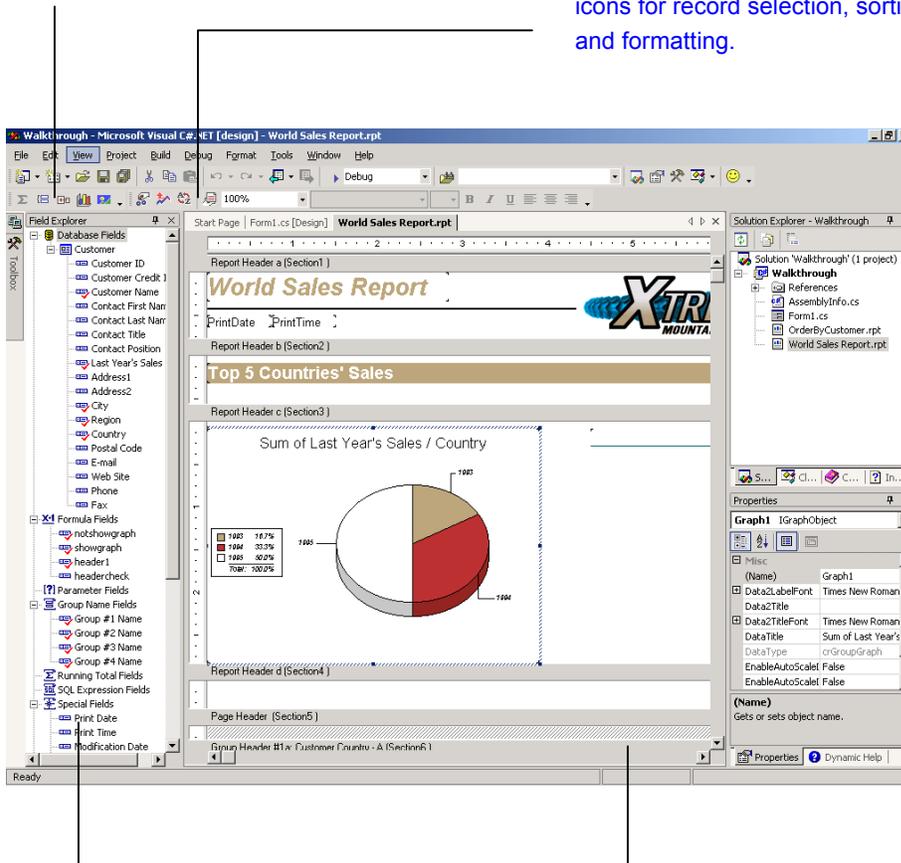
The following three sections describe the layout of the Crystal Report Designer and list the basic report design functions.

The Crystal Report Designer

The graphic below shows the Crystal Report Designer displaying World Sales Report.rpt.

D. Crystal Reports Insert toolbar: icons for inserting, summary fields, groups, subreports, charts, and pictures.

C. Crystal Reports Main toolbar: icons for record selection, sorting, and formatting.



A. Field Explorer: shows all the fields in the displayed report.

B. Main report window: displays World Sales Report.rpt

A. Field Explorer

Enables you to add, modify or delete a field object. You can choose to display the Field Explorer by selecting from the Visual Studio menu bar **View | Other Windows | Document Outline**.

B. Main Report window

Displays the main report in a report window. For a report that contains subreports, there is a **Main Report** window and a separate report window for each subreport when you drill-down on the subreport (by double-clicking it). You can invoke the Report Shortcut Menu by

right-clicking any blank space in a report window (i.e., right-clicking anywhere that is not occupied by an object or a section heading preceding each section in the report).

C. Crystal Reports Main Toolbar

Includes icons for record selection, sorting, section and text object formatting. You may move or size this toolbar. You can also toggle the display of this toolbar by right-clicking on the Visual Studio .NET Standard Toolbar and checking / unchecking the item **Crystal Reports – Main**.

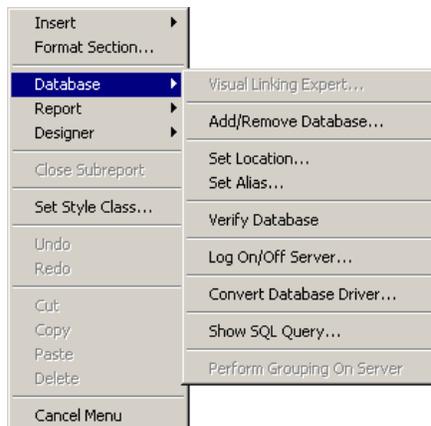
D. Crystal Reports Insert Toolbar

Includes icons to insert summary fields, groups, subreports, charts, and pictures. You may move or size this toolbar. You can also toggle the display of this toolbar by right-clicking on the Visual Studio .NET Standard Toolbar and checking / unchecking the item **Crystal Reports – Insert**.

Crystal Report Designer Functions

Manipulating the Data Source

To make changes to the data source of a report, select **Database** from the Report Shortcut Menu. You can then choose from a sub-menu of options to manipulate a data source including: re-linking tables, adding and removing a database, setting a database location, verifying a database, and converting to a different database driver.



Formatting Objects

Undo/Redo actions

- You can undo (and redo) certain actions. To undo (or redo) an action, select **Undo** (or **Redo**) from the Report Shortcut Menu. You can use Undo on more than one previous action; however, each Undo (or Redo) applies to one action at a time.
- Alternatively, you can use the Undo and Redo buttons on the Visual Studio Standard Toolbar, which apply to multiple actions.

Inserting fields and objects

- To insert a field or object (e.g., text, line, box, chart, and OLE object) into a report, select **Insert** from the Report Shortcut Menu.
- Alternatively, you can insert a text, line or box object from the **Crystal Reports** tab of the Visual Studio Toolbox. You can insert each of these controls by dragging and dropping it, or copying and pasting it into the report. You can also select the desired report section and double-click the control for the desired object, which places the object in the top-left corner of the selected section.
- You can add, modify or delete a field object through the Field Explorer.

Modifying fields and objects

- To format an object, right-click the object and choose **Format** from the shortcut menu. You can also edit, cut, copy, paste or delete the object from the same menu.
- To format a chart object, use the Chart Expert by right-clicking the chart and selecting **Chart Expert** from the shortcut menu. You can also use advanced chart formatting features by pointing to **Format Chart** from the same menu, and selecting the appropriate object.

Formatting the Report Layout

- To insert, delete or format a section in a report, select **Format Section** from the Report Shortcut Menu. Alternatively, you can select **Insert | Section** from the same menu. This invokes the Section Expert.
- To modify the layout of a report, select **Report | Style Expert**.
- To change page margins, select **Designer | Page Setup** from the Report Shortcut Menu. You can also turn on or off the ruler and grid in the report window.

Changing the Report Design

- Select **Report** from the Report Shortcut Menu. You can then select from a sub-menu of options to modify record selection, grouping, sorting, and miscellaneous local report options. You can invoke the Report Expert to help you reformat a report. You can also save a report separately from the project.
- To set miscellaneous global report options, select **Designer | Default Settings** from the Report Shortcut Menu.

Setting Up a Printer

Select **Designer | Printer Setup** from the Report Shortcut Menu. There, you can specify the printer, printer options, paper size, and orientation to be used when printing a report.

Report Design Formatting Example

Open OrderByCustomer.rpt in the Crystal Report Designer if you have not done so, by double-clicking it in the Solution Explorer. In this example, you will format OrderByCustomer.rpt by:

- suppressing the Group Header section
- suppressing the Details section
- removing the Order Amount text object in the Page Header section

To do so:

1. In the Crystal Report Designer, right-click in the Group Header (i.e., the gray area that is labeled "Group Header #1:Customer.Customer Name - A (Section 6)"). Select **Suppress (No Drill-down)** from the shortcut menu.
2. Do the same for the Details section.
3. Select the text object "Order Amount" in the Page Header section and press delete.
4. Save OrderByCustomer.rpt by clicking the **Save OrderByCustomer.rpt** icon on the Visual Studio Standard Toolbar.

What is the Windows Forms Viewer?

Windows Forms in Visual Studio .NET

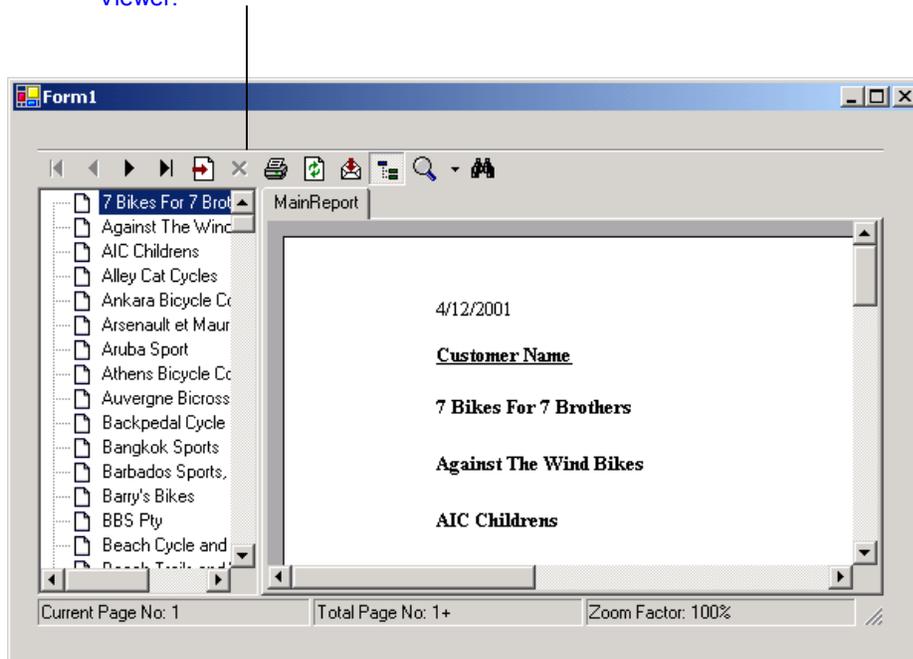
A Windows Form in Visual Studio .NET is similar to a form in earlier versions of Visual Basic. In Visual Studio .NET, you can create a Windows application using any of the supported languages (i.e., Visual C# and Visual Basic). Upon creating a Windows application, Visual Studio .NET automatically opens a Windows Form for you in the project. You can then drag and drop controls from the Visual Studio Toolbox to the Design view of the Windows Form, and program in the Code view of the Windows Form.

After you have designed a Windows Form, you can proceed to build it. If the project builds successfully, the Windows Form will be invoked automatically.

The Windows Forms Viewer

The Windows Forms Viewer is a Windows Form control that allows you to host and view a Crystal report in a Visual Studio .NET Windows application. Similar to other Windows Form controls, adding a Windows Forms Viewer to a Windows Form requires the simple action of dragging a Windows Forms Viewer control from the Visual Studio Toolbox and dropping it in the Design View of the Windows Form. You can then specify the Crystal report to host in the Windows Forms Viewer.

Windows application hosting a report with a Windows Forms Viewer.



Similar to Visual Basic controls, Windows Form controls in the same Windows Form can interact with each other. In particular, the Report Object Model supports interaction between a Windows Form control and a viewer control.

Hosting and Viewing a Crystal Report Using the Windows Forms Viewer

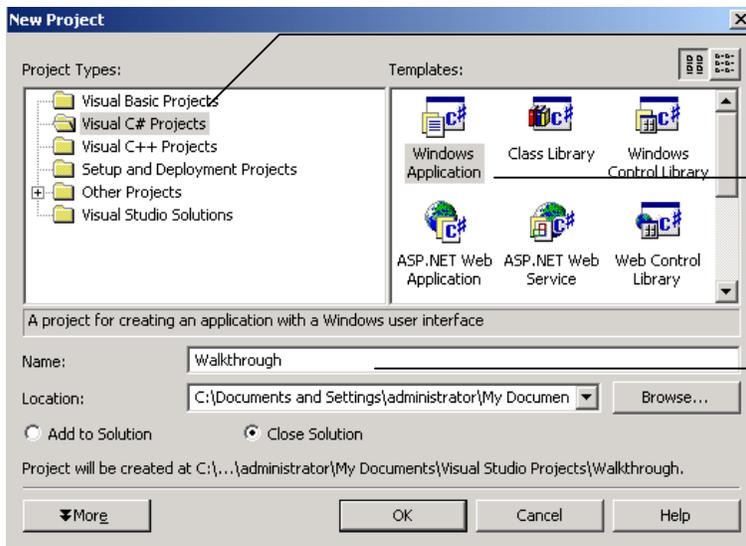
In the walkthrough below, you will create a Visual C# Windows application to host a Crystal report with a Windows Forms Viewer. The Crystal report you will host is OrderByCustomer.rpt that has been created in the section "Creating a Crystal Report" and modified in "Modifying a Crystal Report".

Creating a Windows Form

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 **Windows Application** in the right pane.

You may choose to create your project in any folder of your choice. For this example, you will create the project, WindowsFormWalkthrough, in the same folder used above, C:\Documents and Settings\Administrator\My Documents\Visual Studio Projects.

4. Enter "WindowsFormWalkthrough" in the **Name** field. Accept the default location "C:\Documents and Settings\Administrator\My Documents\Visual Studio Projects" in the **Location** field. Click **OK**. This creates a Visual C# Windows project named WindowsFormWalkthrough in C:\Documents and Settings\Administrator\My Documents\Visual Studio Projects.



Select Visual C# as a project type.

Select Windows Application.

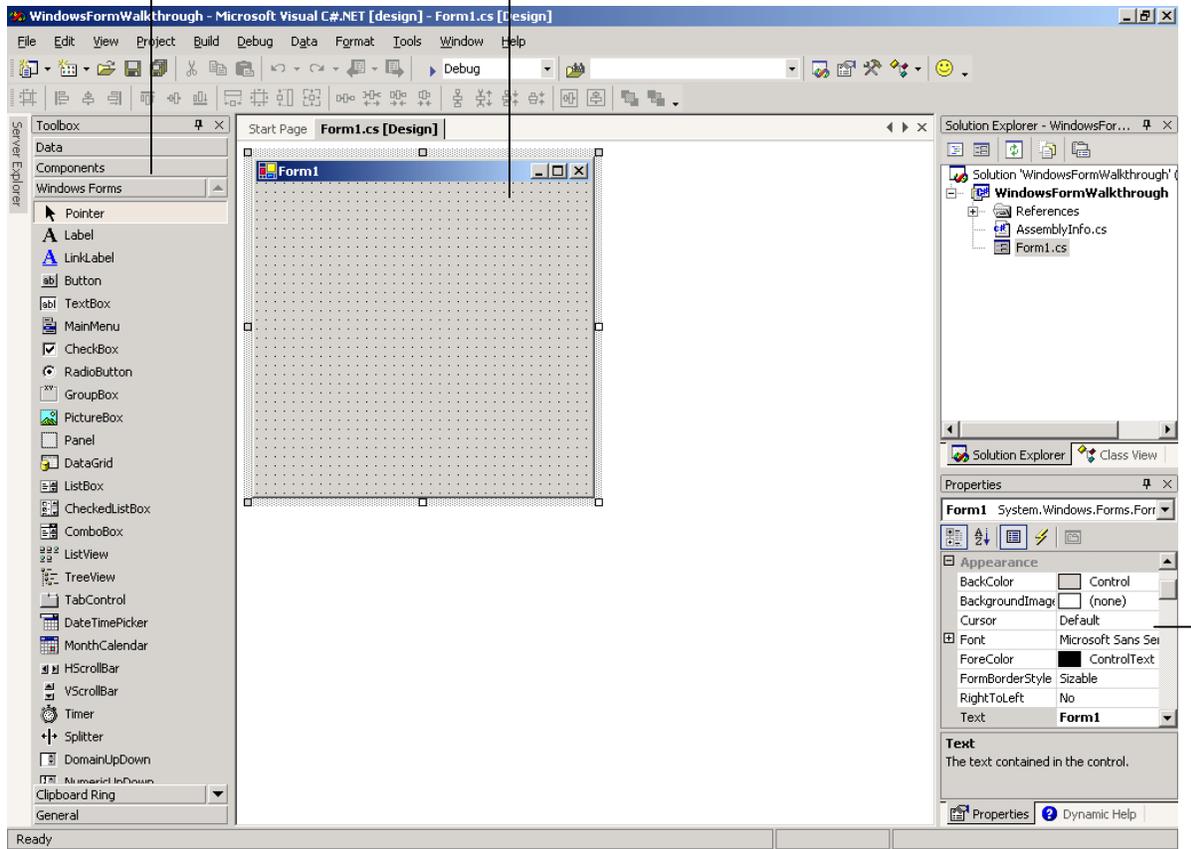
Name the project.

Upon clicking OK in the **New Project** dialog, Visual Studio creates a Windows Form (with the default name "Form1.cs") in the project, WindowsFormWalkthrough. Visual Studio opens the Windows Form in its Design View. The Visual Studio Toolbox also displays a set of controls for the Windows Form.

The Windows Forms tab of the toolbar displays a set of Windows Form controls.

The Windows Form in the Design View.

The Properties window displays the properties for the selected item in the Design View.

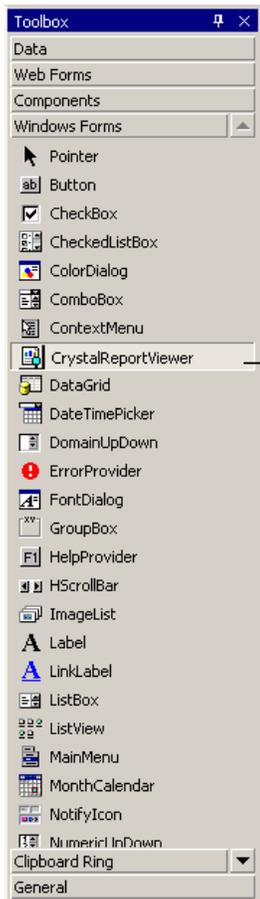


Inserting the Windows Forms Viewer

1. In Visual Studio .NET, when a Windows Form is opened, the **Windows Forms** tab of the Toolbox displays a list of Windows Form controls, including the Windows Forms Viewer. The Windows Forms Viewer control is labeled as **CrystalReportViewer** in the Toolbox.

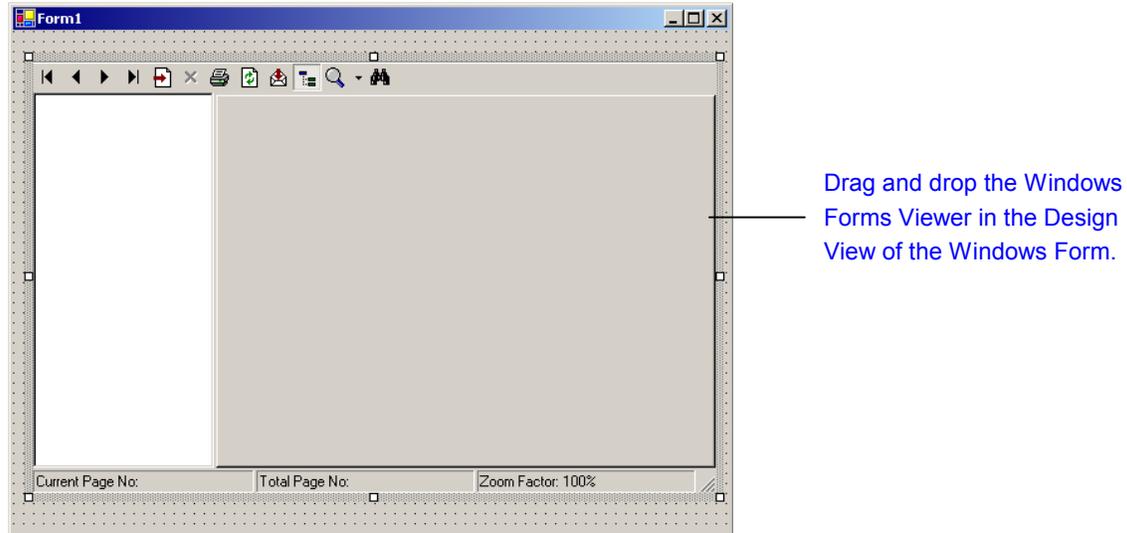
TIP In Visual Studio .NET, you can anchor the Toolbox by clicking the "pin" icon at the top-right corner of the Toolbox.

As well, items in a Toolbox tab can be displayed alphabetically by right-clicking on the tab, and selecting **Sort Items Alphabetically** from the shortcut menu.



The Windows Forms Viewer is labeled "CrystalReportViewer" in the Windows Forms tab of the Toolbox.

2. If you are not already in the Design View of Form1.cs, go to the Design View by highlighting Form1.cs in the Solution Explorer, and clicking the **View Designer** icon on the Solution Explorer toolbar.
3. You can now add the Windows Forms Viewer to Form1.cs by dragging CrystalReportViewer from the **Windows Forms** tab of the Toolbox and dropping it in the Windows Form.



After inserting the Windows Forms Viewer into Form1.cs, you can then move and resize it in the Design View of the Windows Form.

The Windows Forms Viewer

The Windows Forms Viewer is currently available as a control with the following components:

- Toolbar containing these buttons:
 - go to first page
 - page backward
 - page forward
 - go to last page
 - go to specified page
 - close current window (for group or subreport only)
 - print
 - refresh report
 - export
 - toggle group tree
 - magnify / reduce report
 - search
- Group Tree
- Main Report Window

Specifying a Report to be hosted on the Windows Forms Viewer

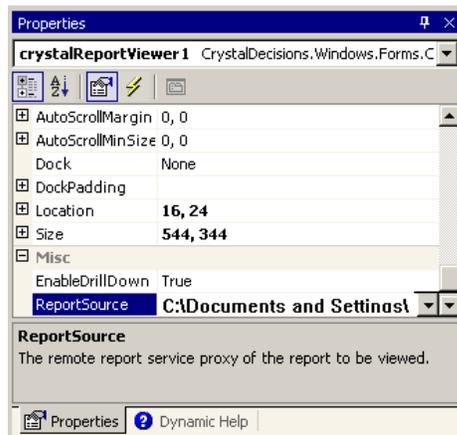
You can specify a report to be hosted on the Windows Forms Viewer through the ReportSource property of the CrystalReportViewer object. You can do so via the Properties Window in Visual Studio .NET (when the Design View of the Windows Form is active), or directly through code (in the Code View of the Windows Form).

In the following example, you will host the report OrderByCustomer.rpt in the Windows Forms Viewer. OrderByCustomer.rpt is the same report created in the project Walkthrough in the section "Designing a Crystal Report" above.

For a more detailed description of the different methods that you can use to bind a report to a Windows Forms Viewer, please refer to Appendix B.

Specifying Report Through the Properties Window

1. If you are not already in the Design View of Form1.cs, go to the Design View by highlighting Form1.cs in the Solution Explorer, and clicking the **View Designer** icon on the Solution Explorer toolbar.
2. Select the Windows Forms Viewer control.
3. In the Visual Studio .NET Properties Window, scroll down to the **ReportSource** property. Click the down-arrow, browse and select the report with its full path name, "C:\Documents and Settings\Administrator\My Documents\Visual Studio Projects\Walkthrough\OrderByCustomer.rpt". (Both projects Walkthrough and WindowsFormWalkthrough have been created in the same location C:\Documents and Settings\Administrator\My Documents\Visual Studio Projects.)



Browse and select the full path name of the report for the ReportSource property.

After you build and run the Windows application, the Windows Forms Viewer will display OrderByCustomer.rpt.

Specifying Report Through the Code View

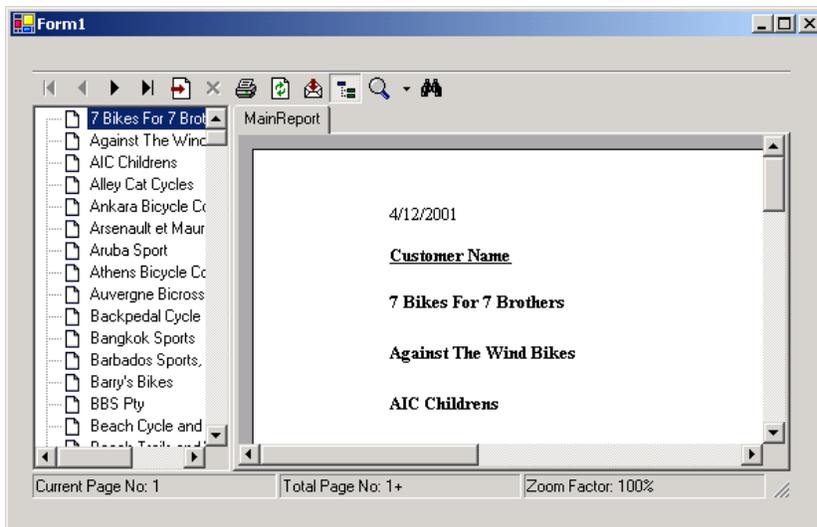
As an alternative to specifying the ReportSource property in the Properties Window, you can also specify the full path name of the report directly in the Code View of Form1.cs, as highlighted in bold below:

```
public Form1()
{
    //
    // Required for Windows Form Designer support
    //
    InitializeComponent();

    //
    // TODO: Add any constructor code after InitializeComponent call
    //
    crystalReportViewer1.ReportSource = "C:\\Documents and" +
    "Settings\\Administrator\\My Documents\\Visual Studio" +
    "Projects\\Walkthrough\\OrderByCustomer.rpt";
}
```

Building and Running the Windows Application

To build the application, select **Build** from the **Build** menu. To run the application, select **Start Without Debugging** from the **Debug** menu.



Appendix A: Class Description for the Windows Forms Viewer Control

Namespace	<code>CrystalDecisions.Windows.Forms</code>
Class name	<code>CrystalReportViewer</code>
Methods	<pre> public virtual void CloseView (object viewURI) public override void Dispose() public void DrillDownOnGroup (TotallerNodeID groupInfo) public virtual void ExportReport () public virtual int GetCurrentPageNumber () public virtual void PrintReport () public virtual void RefreshReport () public virtual bool SearchForText (string text) public virtual void ShowFirstPage () public virtual void ShowGroupTree () public virtual void ShowLastPage () public virtual void ShowNextPage () public virtual void ShowNthPage ([in] int PageNumber) public virtual void ShowPreviousPage () public virtual void Zoom (int ZoomLevel) </pre>
Properties	<pre> Int32 ActiveViewIndex Boolean DisplayBackgroundEdge Boolean DisplayGroupTree Boolean DisplayToolBar Boolean EnableDrillDown TableLogOnInfos LogOnInfo ParameterFields ParameterFieldInfo Object ReportObjectSource Object ReportSource RightToLeft RightToLeft String SelectionFormula Boolean ShowCloseButton Boolean ShowExportButton Boolean ShowGotoPageButton Boolean ShowGroupTreeButton </pre>

	<pre> Boolean ShowPrintButton Boolean ShowRefreshButton Boolean ShowTextSearchButton Boolean ShowZoomButton Int32 ViewCount StatusBar ViewerStatusBar ToolBar ViewerToolBar </pre>
Events	<pre> public event EventHandler Drill; public event EventHandler DrillDownSubreport; public event EventHandler HandleException; public event EventHandler Navigate; public event EventHandler Refresh; public event EventHandler Search; public event EventHandler ViewZoom; </pre>

Appendix B: Report Binding Methods for the Windows Forms Viewer

In general, there are four methods you can use to bind a report to a Windows Forms Viewer:

- By the report name
- By a report object
- By untyped report component
- By strongly-typed report component

The following section lists several scenarios and describes the methods you can use to bind a report to a Windows Forms Viewer in each scenario.

Scenario A – Local Report

Suppose the following conditions exist and a Crystal report is available on a local machine:

- You have a Crystal report, My Report.rpt, in the folder, c:\Reports, on a local machine.
- You have created a Visual C# Windows project, MyProject, in the default folder for the user logged on as Beaver, c:\Documents and Settings\Beaver\My Documents\Visual Studio Projects.
- You have inserted a Windows Forms Viewer in the Windows Form, Form1.cs, in MyProject.

You can bind My Report.rpt to the Windows Forms Viewer using the following methods:

By report name:

You can specify the ReportSource property through either the Properties window or the Code view for Form1.cs:

- Activate the Design view of Form1.cs and select crystalReportViewer1. In the Visual Studio Properties window, browse and select the report "c:\Reports\My Report.rpt" for the **ReportSource** property.
- Or, in the Code view of Form1.cs, specify:

```
crystalReportViewer1.ReportSource = "c:\\Reports\\My Report.rpt";
```

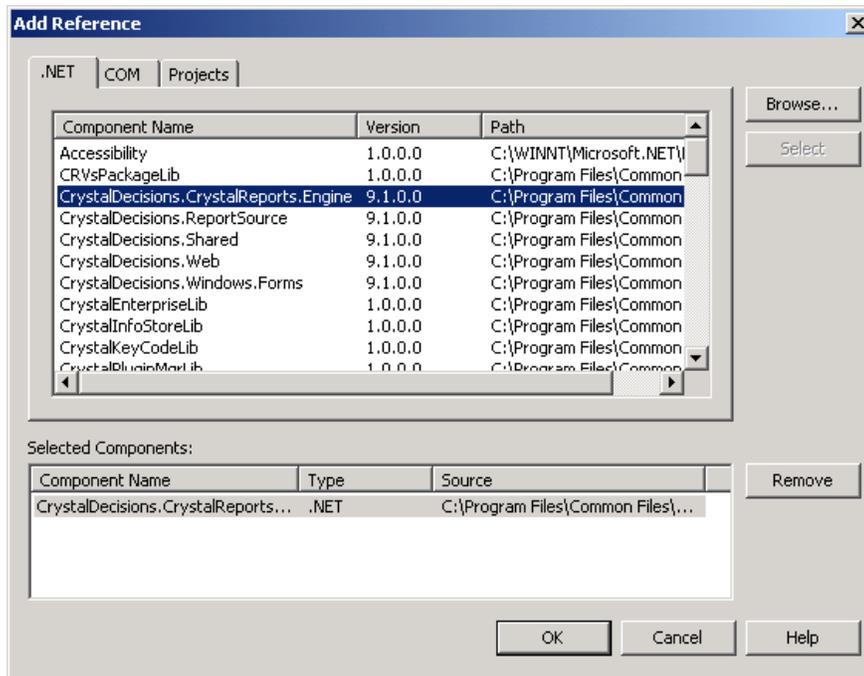
By report object:

You may create a report object, assign a report to that object, and bind the object to a Windows Forms Viewer. Through the report object, you can access the entire object model exposed by Crystal Reports for Visual Studio .NET.

To bind a report object for My Report.rpt to the Windows Forms Viewer, do the following:

1. Add CrystalDecisions.CrystalReports.Engine as a reference to the project. To do so:
 - In the Solution Explorer, right-click on References under MyProject, and select Add Reference from the shortcut menu. This invokes the Add Reference dialog box to allow you to add references.

- In the .NET tab of the Add Reference dialog box, add the component CrystalDecisions.CrystalReports.Engine, by highlighting the component, and clicking the Select button. Then click OK.



2. In the Solution Explorer, highlight Form1.cs and click the **View Code** icon on the Solution Explorer Toolbar. This displays the codes for Form1.cs.
3. Add the following namespace to Form1.cs:

```
using CrystalDecisions.CrystalReports.Engine;
```
4. Create a report object as a public ReportDocument variable in Form1():

```
public ReportDocument oRpt=null;
```
5. In the method Form1(), use the Report Document object to load My Report.rpt, and bind the object to the Windows Forms Viewer:

```
Public Form1()
{
    //
    // Required for Windows Form Designer support
    //
    InitializeComponent();

    //
    // TODO: Add any constructor code after InitializeComponent call
```

```
//  
oRpt = new ReportDocument();  
oRpt.Load("c:\\Reports\\My Report.rpt");  
crystalReportViewer1.ReportSource = oRpt;  
}
```

By untyped report component:

Suppose you have added an untyped report component, reportDocument1, to the Windows Form.

1. Load the untyped report component with c:\Reports\My Report.rpt, by adding the following line in the form's source file after the call to `InitializeComponent()`:

```
reportDocument1.Load ("c:\\My Report.rpt");
```

2. Bind the untyped report component to the Windows Forms Viewer by specifying the ReportSource property through the Code view for Form1.cs:

```
crystalReportViewer1.ReportSource = reportDocument1;
```

NOTE For more information on report components, please refer to the online help in Visual Studio .NET.

Scenario B – Local Report Added to Project

Suppose the conditions in Scenario A exist, plus the following:

- You have added the report My Report.rpt to MyProject.

You can bind My Report.rpt to the Windows Forms Viewer using the following methods:

By report object:

You can bind My Report.rpt to the Windows Forms Viewer by assigning the corresponding report object to the ReportSource property through the Code view for Form1.cs: in the function `Form1()`, instantiate an object of the class `My_Report` (which is created in My Report.cs when you add the report to My Project), and bind the object to the Windows Forms Viewer:

```
crystalReportViewer1.ReportSource = new My_Report();
```

By strongly-typed report component

Suppose you have added My Report.rpt as a strongly-typed report component to MyProject.

You can bind the report as a component to the Windows Forms Viewer by specifying the ReportSource property through either the Properties window or the Code View for Form1.cs:

- Select the Windows Forms Viewer in the Design view of Form1.cs. In the Properties window, for the **ReportSource** property, select **my_Report1 [MyProject.My_Report]** from the drop down list.
- Or, in the Code view of Form1.cs, specify:

```
crystalReportViewer1.ReportSource = my_Report1;
```

NOTE For more information on report components, please refer to the online help in Visual Studio .NET.