

Crystal Reports

Report Designer Component – Tutorial and Getting Started Guide

Overview

This document describes how to create a Visual Basic application using the Crystal Report Designer Component 8 (RDC). This document covers the basic properties and methods that would be used to create an application that prints, previews, or exports a report. This document is for use with Crystal Reports 8 and higher.

Contents

INTRODUCTION	2
THE OBJECT MODEL	2
CREATING AN APPLICATION THAT USES DESIGNER (DSR) FILES.....	3
<i>Adding the RDC to a new application.....</i>	<i>3</i>
<i>Code added by the RDC.....</i>	<i>4</i>
<i>Setting the Logon info for the Report Object.....</i>	<i>5</i>
<i>Previewing the Report.....</i>	<i>6</i>
<i>Printing the Report Directly to the Printer</i>	<i>6</i>
<i>Exporting the Report directly to a Disk File.</i>	<i>7</i>
USING REPORT (RPT) FILES WITH THE RDC IN VISUAL BASIC	7
<i>Creating a RPT file</i>	<i>8</i>
<i>Creating the VB application to load the RPT file.....</i>	<i>9</i>
<i>Setting the Logon info for the Report Object.....</i>	<i>10</i>
<i>Previewing the Report.....</i>	<i>11</i>
<i>Printing the Report Directly to the Printer</i>	<i>11</i>
<i>Exporting the Report directly to a Disk File.</i>	<i>11</i>
FINDING MORE INFORMATION.....	12
CONTACTING CRYSTAL DECISIONS FOR TECHNICAL SUPPORT	13

Introduction

The RDC is based on Microsoft's COM technology, and is supported in Microsoft Visual Basic (VB) 5 or higher. For VB developers, creation of reports and developing with them are simplified in one environment. VB developers can now create reports at design time and use the RDC object models to take advantage of Object Oriented programming techniques.

Report development is achieved by accessing properties and methods through an object hierarchy similar to those found in other popular Automation Servers like DAO and Excel.

The Report Designer Component (RDC) is composed of 4 primary DLLs:

Craxdr.dll – (*Crystal Reports ActiveX Designer Runtime Library*). This runtime library formats and populates the report with data from the database. This DLL needs to be included in the distribution of your compiled application.

Craxddt.dll – (*Crystal Reports ActiveX Designer Design Time Library*). This design time library processes the design time formatting. This DLL is only used when designing a report in the VB environment and should not be distributed with your compiled application.

Craxdui.dll – (*Crystal Reports ActiveX Designer User Interface Library*). This designer user interface library displays a user interface in VB that can be used to create a report in the VB IDE. This DLL is only used in the application at design time and should not be distributed with your compiled application.

Crviewer.dll – (*Crystal Reports Viewer Control Library*). This ActiveX Viewer control previews the report objects and can export or print the report from the viewer window. The crviewer.dll also has an object model hierarchy.

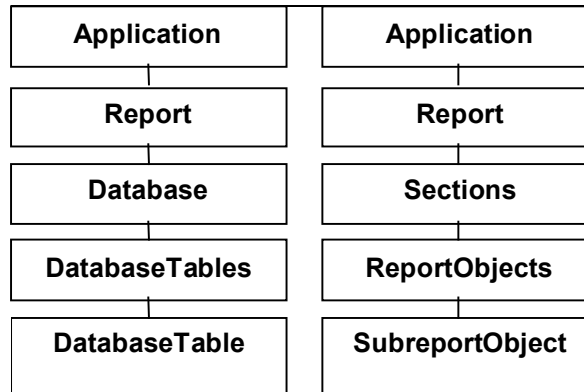
The Object Model

The following is a brief description of the Object Model that the RDC uses. The RDC Automation Server is the file craxdr.dll and at the top of the RDC Automation Server is the Application object.

All RDC objects are children of the Application object. This means that the Application object must first be created and instantiated before you can start working with other objects in the Object Model hierarchy.

The RDC Application object is similar to the Application object from the Crystal Reports (CR) Automation Server (cpeaut32.dll), as well as Excel's Automation Server and the DAO Engine object.

If you are designing reports inside of Visual Basic and are loading these reports as DSR files in your application then you will not need to create this Application object. It will automatically be created when you dimension your Report Object.



The diagram above shows a break down of the parent child relationship between the objects in which this tutorial will be based. The full Object Model hierarchy can be found in the Developer's Help file (**Developr.hlp**) by searching the index for "**The Report Designer Object Model overview**".

The naming of the objects is significant because the use of a plural denotes a Collection. A Collection is an object that holds objects of a similar kind. For example, the **DatabaseTables** collection is used to reference all the **DatabaseTable** objects available in a report.

Creating an application that uses Designer (DSR) files

A DSR file is a designer file that can be used to design reports within the Visual Basic IDE. When using DSR files it is important to remember that once your application is compiled into an EXE file so are the DSR files. The reports will remain the same and cannot be changed or updated outside of the VB IDE. This is useful if you do not wish for your customers to access these reports outside of your application.

Adding the RDC to a new application

The following steps describe how to add the RDC to a VB application and how to create a report in the VB IDE.

1. Open a new project in Visual Basic (VB). Click the **File** menu and then click **New Project**. Select **Standard EXE** and click **OK**.
2. Click the **Project** menu, and click **More ActiveX Designers**. Select **Crystal Reports 8**.
3. The **Seagate Crystal Reports Gallery** dialog box appears. Select **Using the Report Expert** and then select **Standard** from the **Expert** list box. Click **OK**.
4. The **Standard Report Expert** dialog box appears. Select the **Data** tab and click the **Other** button.

NOTE

In the Data Tab, there are three option you can select for your data source:

	-Data Environment -Project (for example, Active Data) -Other (for example, ODBC)
--	--

From the **Data Explorer**, select **ODBC** then select **Xtreme Sample Database** (The default ODBC Data source that Crystal Reports installs). Select the **Customer** Table and click the **Add** button. Click **Close**.

- The **Customer** table is now in the **Tables** list box in the **Standard Report Expert**. Click on the **Next** button to add fields from the Customer table to the report.

Add the **Customer ID** and the **Customer Name** field to the report and then click the **Finish** button.

- The **Crystal Report Expert** dialog appears with the following two options:

- “Do you want the expert to add a form containing the Crystal Report Viewer control.”

This option will create a form in the project with a viewer control on it and write some basic code that will preview this report.

- “Would you like the expert to modify the properties to set this form as the startup object?”

This option will set this newly created form as the startup form in your project.

Select **Yes** for both these options and click **OK**. This causes the RDC to add the following references and component to your application:

References (Project | References menu):

- Crystal Report 8 ActiveX Designer Run Time Library**
- Crystal Report 8 ActiveX Designer Design Time Library**
- Crystal Report Viewer Control**

Components (Project | Components menu):

- Crystal Report Viewer Control**

Code added by the RDC

The RDC inserts code in the **General Declaration**, **Form_Load** and **Form_Resize** events of the Form that was added to your project.

General Declarations

```
Dim Report As New CrystalReport1
```

```
'Declare the report object as a new CrystalReport1
```

'By declaring the Report Object in the General Declarations section of the Form with the CRViewer on it, you can assure that the Report Object will stay in scope as long as this form is open.

```
Private Sub Form_Load()
    Screen.MousePointer = vbHourglass
    CRViewer1.ReportSource = Report
    CRViewer1.ViewReport
    Screen.MousePointer = vbDefault
End Sub
```

'Passes the report object to the Crystal Reports Viewer Control. The ReportSource property tells the viewer what report object to load when the ViewReport method is called.

'The ViewReport method tells the viewer to load the report, start processing the data and format the report.

```
Private Sub Form_Resize()
    CRViewer1.Top = 0
    CRViewer1.Left = 0
    CRViewer1.Height = ScaleHeight
    CRViewer1.Width = ScaleWidth
End Sub
```

'Aligns the Crystal Report Viewer Control to the same size as the form it is on.

NOTE

Any additional code for the Report object, like exporting or logging onto a database, should be placed before **CRViewer1.ReportSource = Report**.

Setting the Logon info for the Report Object

Use the **SetLogonInfo** method (of the **DatabaseTables** object) to create a connection to the database that your report object uses. The SetLogonInfo method can be used in two different cases:

- To create a connection to the database that the report was created from
- To change the database that the report uses to retrieve its data

In this project, to create a connection to the **Xtreme Sample Database** ODBC connection, add the following code to your **Form_Load** event before you set the **ReportSource** property of the CRViewer.

```
Private Sub Form_Load ()
    Screen.MousePointer = vbHourglass
```

```

For I = 1 to Report.Database.Tables.Count
  Report.Database.Tables(I).SetLogonInfo _
    "Xtreme Sample Database", "", "", ""
Next I
CRViewer1.ReportSource = Report
CRViewer1.ViewReport
Screen.MousePointer = vbDefault
End Sub

```

NOTE	<p>The syntax for the SetLogonInfo method is:</p> <p>Object.SetLogonInfo Server Name, Database Name, User ID, Password</p> <p>Server Name: Specifies the DSN that you would like your report to connect to if you are using ODBC. If you are using a native connection to your database, then specify the server name. If you pass in an empty string then the default server name (from the report) is used.</p> <p>Database Name: Specifies the database name that you would like to connect this report to. If you pass in an empty string then the default database name (from the report) is used.</p> <p>User ID: Specifies the user id that you would like to use to make the connection to the database. If you pass in an empty string then the default User ID (from the report) is used.</p> <p>Password: Specifies the password needed to connect to your database with the user id that was specified in the user id parameter. Passwords are never saved with the report so if your database requires a password you will have to set this parameter in order to create a connection to your database.</p>
-------------	--

Previewing the Report

After setting the logon info using the **SetLogonInfo** method, and sending the Report object to our CRViewer control, you are now ready to run this project.

To run the project go to the **Run** menu and click on **Start** or press **F5**. This should display the Report in the Crystal Report Viewer Control.

Printing the Report Directly to the Printer

To print a report directly to the printer without previewing, use the **PrintOut** method of the **Report** object.

To send print the report directly to the printer:

1. Comment out the following two lines of code that are in the **Form_Load** event:

```

`CRViewer1.ReportSource = Report
`CRViewer1.ViewReport

```

2. Replace those two lines of code with the following line of code:

Report.PrintOut False

NOTE	<p>The syntax for the PrintOut method is:</p> <pre>object.PrintOut PromptUser, NumberOfCopies, Collated, StartPageNumber, StopPageNumber</pre> <p>PromptUser: Specifies whether or not you would like your application to prompt for the printer options.</p> <p>NumberOfCopies (Optional): Specifies how many copies you would like to print to the printer.</p> <p>Collated (Optional): Specifies if you would like the report to print collated or not.</p> <p>StartPageNumber (Optional): Specifies the page you would like to start printing from.</p> <p>StopPageNumber (Optional): Specifies the page you would like to stop printing from.</p>
-------------	--

Exporting the Report directly to a Disk File.

To export a report directly to a disk file without prompting for customer input, use the **ExportOptions** object off the **Report** object.

To export a report directly to a disk file:

1. Comment out the following two lines of code that are in the Form_Load event:

```
`CRViewer1.ReportSource = Report
`CRViewer1.ViewReport
```

2. Replace those two lines of code with the following lines of code:

```
With Report.ExportOptions
    .DestinationType = CrEDTDiskFile
    .FormatType = crEFTRichText
    .DiskFileName = App.Path & "\Report.rtf"
End With
Report.Export False
```

NOTE	<p>For further information on the properties and methods of the ExportOptions object, refer to the Crystal Developer help file (Devlopr.hlp) or the Report Designer Component help file (Crrdc.hlp).</p>
-------------	--

Using Report (RPT) files with the RDC in Visual Basic

When using RPT files in your application with the Report Designer Component (RDC), there are advantages and disadvantages

Some advantages of using RPT files are:

- You can distribute the reports with your application and allow the customers to update, maintain, and modify the reports themselves.
- Your executable will not be as large as it would be if the reports were compiled into it.
- You can modify and distribute the same application with different reports to provide flexible applications for multiple work environments.
- Separate designers are required for DSR files, but are not required for RPT files. Not using separate designers will reduce the resources your application uses.

Some disadvantages of using RPT files are:

- Customers can modify these reports even if you do not want them to. (you cannot write protect a report)
- Distributing RPT files will increase the size of your application installation package.

Creating a RPT file

When a report is created outside of Visual Basic in the stand-alone Crystal Report Designer it is saved as a RPT file. These RPT files can be modified by anyone who has the stand-alone Crystal Report Designer installed.

NOTE	The stand-alone Crystal Reports designer does not allow you to choose a Data Environment as the database or use unbound fields.
-------------	---

To create a RPT file in the stand alone Crystal Reports Designer

1. Open the **Crystal Reports Designer** from your Start Menu.
2. From the startup screen, select **Standard** from the **Choose an Expert** box. Click **OK**.
3. The **Standard Report Expert** dialog box appears. Under the **Data** tab click on the **Database** button.

The **Data Explorer** dialog box appears. Under **ODBC**, select **Xtreme Sample Database**.

4. Under **Xtreme Sample Database** select the **Customer** table and click **Add**. Click the **Close** button to return to the **Standard Report Expert**.
5. Click the **Next** button to move to the **Fields** tab.
6. From the **Fields** tab, select the **Customer ID** and **Customer Name** field then click **Add**. Click the **Finish** button to preview this report.
7. A report with the **Customer ID** and **Customer Name** field will now appear. To save this report, under the **File** menu select **Save**. Save this report as C:\RDCTest.rpt.

Now that the report is saved as a disk file you can setup a new VB project that will load this report into the Crystal Reports Viewer Control.

Creating the VB application to load the RPT file

To create a new VB project that loads a report into the Crystal Reports Viewer Control:

1. Open a new project in Visual Basic (VB). Click the **File** menu and then click **New Project**. Select **Standard EXE** and click **OK**.
2. Click the **Project** menu and then click on **Components**. Put a check beside the **Crystal Reports Viewer Control** and then click **OK**.

Click the **Project** menu and then click **References**. Put a check mark beside **Crystal Report 8 ActiveX Designer Run Time Library** and then click **OK**.

3. To add the **Crystal Reports Viewer Control** to Form1, under the **View** menu and click **Toolbox**. Double click the **CRViewer** Control from the **Toolbox** to place it on **Form1**.
4. Open the code window for **Form1** and under the **General Declarations** section, type the following lines of code:

```
'General Declarations
Dim Appl as New CRAXDRT.Application
Dim Report as New CRAXDRT.Report
'Declare the application and report object
```

5. Open the code window for the **Form_Resize** event of **Form1** and type the following lines of code:

```
Sub Form_Resize()
    With CRViewer1
        .Top = 0
        .Left = 0
        .Width = Me.ScaleWidth
        .Height = Me.ScaleHeight
    End With
    'This above code resizes the Viewer Control to the size of
    'the form
End Sub
```

6. To populate the report object and send it to the Viewer, open the code window for the **Form_Load** event of **Form1** and type the following lines of code:

```

Sub Form_Load()
    Set Report = Appl.OpenReport("C:\RDCTest.Rpt")
    CRViewer1.ReportSource = Report
    CRViewer1.ViewReport
End Sub

`Pass that Report object to the CrViewer control using the
`ReportSource property and View the report using the
`ViewReport method

```

Setting the Logon info for the Report Object

Use the **SetLogonInfo** method (of the **DatabaseTables** object) to create a connection to the database that your report object uses. The **SetLogonInfo** method can be used in two different cases:

- To create a connection to the database that the report was created from
- To change the database that the report uses to retrieve its data

In this project, to create a connection to the **Xtreme Sample Database** ODBC connection, add the following code to your **Form_Load** event before you set the **ReportSource** property of the CRViewer.

```

Private Sub Form_Load ()
    Set Report = Appl.OpenReport
    Screen.MousePointer = vbHourglass
    For I = 1 to Report.Database.Tables.Count
        Report.Database.Tables(I).SetLogonInfo _
            "Xtreme Sample Database", "", "", ""
    Next I
    CRViewer1.ReportSource = Report
    CRViewer1.ViewReport
    Screen.MousePointer = vbDefault
End Sub

```

NOTE	<p>The syntax for the SetLogonInfo method is:</p> <p>Object.SetLogonInfo Server Name, Database Name, User ID, Password</p> <p>Server Name: Specifies the DSN that you would like your report to connect to if you are using ODBC. If you are using a native connection to your database, then specify the server name. If you pass in an empty string then the default server name (from the report) is used.</p> <p>Database Name: Specifies the database name that you would like to connect this report to. If you pass in an empty string then the default database name (from the report) is used.</p> <p>User ID: Specifies the user id that you would like to use to make the connection to the database. If you pass in an empty string then the default User ID (from the report) is</p>
-------------	---

	<p>used.</p> <p>Password: Specifies the password needed to connect to your database with the user id that was specified in the user id parameter. Passwords are never saved with the report so if your database requires a password you will have to set this parameter in order to create a connection to your database.</p>
--	---

Previewing the Report

After setting the logon info using the **SetLogonInfo** method, and sending the Report object to our CRViewer control, you are now ready to run this project.

To run the project go to the **Run** menu and click on **Start** or press **F5**. This should display the Report in the Crystal Report Viewer Control.

Printing the Report Directly to the Printer

To print a report directly to the printer without previewing, use the **PrintOut** method of the **Report** object.

To send print the report directly to the printer:

1. Comment out the following two lines of code that are in the Form_Load event:

```
'CRViewer1.ReportSource = Report
'CRViewer1.ViewReport
```

2. Replace those two lines of code with the following line of code:

```
Report.PrintOut False
```

NOTE	<p>The syntax for the PrintOut method is:</p> <pre>object.PrintOut PromptUser, NumberOfCopies, Collated, StartPageNumber, StopPageNumber</pre> <p>PromptUser: Specifies whether or not you would like your application to prompt for the printer options.</p> <p>NumberOfCopies (Optional): Specifies how many copies you would like to print to the printer.</p> <p>Collated (Optional): Specifies if you would like the report to print collated or not.</p> <p>StartPageNumber (Optional): Specifies the page you would like to start printing from.</p> <p>StopPageNumber (Optional): Specifies the page you would like to stop printing from.</p>
-------------	--

Exporting the Report directly to a Disk File.

To export a report directly to a disk file without prompting for customer input, use the ExportOptions object off the Report object.

To export a report directly to a disk file:

1. Comment out the following two lines of code that are in the Form_Load event:

```
`CRViewer1.ReportSource = Report  
'CRViewer1.ViewReport
```

2. Replace those two lines of code with the following lines of code:

```
With Report.ExportOptions  
    .DestinationType = CrEDTDiskFile  
    .FormatType = crEFTRichText  
    .DiskFileName = App.Path & "\Report.rtf"  
End With  
Report.Export False
```

NOTE

For further information on the properties and methods of the ExportOptions object, refer to the Crystal Developer help file (Devopr.hlp) or the Report Designer Component help file (Crrdc.hlp).

Finding More Information

The following documents can be downloaded from our website at

<http://support.crystaldecisions.com/docs>

- RDC8_FAQ.pdf - This document contains a listing of frequently asked questions posed by users of the Crystal Report Designer Component (RDC) version 8.
- SCR_TTXADO.pdf - This document provides information about using Data Definition (TTX) files and Active Data with Crystal Reports. This document outlines how to create a report that has a Microsoft ActiveX Data Objects (ADO) recordset passed to it at runtime.
- Apps_VB_RDC.pdf - This document provides a listing of all available Report Designer Component (RDC) sample applications in Microsoft Visual Basic available on the Crystal Decisions Support web site. Included is a description of each application and a link to its download location.
- Apps_C++_RDC.pdf - This document provides a listing of all available Report Designer Component (RDC) sample applications in Microsoft Visual C++ available on the Crystal Decisions Support web site. Included is a description of each application and a link to its download location.
- RDC8_Browser.exe - A Help file for developers using the Report Designer Component (RDC). This utility allows you to navigate through the object hierarchy of the RDC using an "Explorer tree" type interface. Included are descriptions and/or syntax of objects, collections, methods, and properties.

The following knowledge base article can be downloaded from our website at

<http://support.crystaldecisions.com/kbase>

- Kbase C2007290 - Runtime File Requirements for the Report Designer Component (RDC) 8

Contacting Crystal Decisions for Technical Support

We recommend that you refer to the product documentation and that you visit our Technical Support web site for more resources.

Self-serve Support:

<http://support.crystaldecisions.com/>

Email Support:

<http://support.crystaldecisions.com/support/answers.asp>

Telephone Support:

<http://www.crystaldecisions.com/contact/support.asp>