

Crystal Reports in PowerBuilder

Report Designer Component – Tutorial and Getting Started Guide

Overview

This document describes how to create a PowerBuilder application using the Crystal Report Designer Component 9 (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 9.0 and higher.

Contents

INTRODUCTION	2
THE OBJECT MODEL	2
USING REPORT FILES WITH THE RDC	2
<i>Creating a Report file</i>	<i>3</i>
<i>Creating the PowerBuilder application</i>	<i>4</i>
Add the Crystal Viewer Control	4
Logon to the Database	5
Setting Parameters	5
Preview the Report.....	6
Exporting the Report.....	6
Printing the Report.....	7
Closing the Window	8
FINDING MORE INFORMATION	8
Database Connectivity	8
Printing	8
Parameter Fields	9
Exporting	9

Introduction

The RDC is supported in PowerBuilder as an OLEObject. Report development is achieved by accessing properties and methods through an object hierarchy.

The Report Design Component (RDC) for use in PowerBuilder is composed of 2 primary DLLs:

Craxdrt9.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.

Crviewer9.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 crviewer9.dll also has an object model hierarchy.

NOTE	In Crystal Reports 10.0 the <i>Crystal Reports ActiveX Designer Runtime Library</i> will be CRAXDRT.dll and the <i>Crystal Reports Viewer Control Library</i> will be CRVIEWER.dll
-------------	--

The Object Model

The following is a brief discussion of the Object Model that the RDC uses. The RDC Automation Server is the file craxdrt9.dll and at the top of the Automation Server is the Application object.

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

The Object Model hierarchy can be found in the Crystal Reports Developer Help file (CrystalDevHelp.chm) by searching “Index” for “Object Model, Report Designer” or by using “Search” for “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.

Using Report Files with the RDC

When using Report (RPT) files in your application with the Report Designer Component (RDC) be aware of the following points:

- You can distribute the reports with your application and allow the customers to update, maintain and modify the reports themselves using the Crystal Reports designer.
- Customers with a Crystal Reports designer installed, can modify the reports even if you do not want them to. (You cannot write protect a report).

- You can modify and distribute the same application with different reports to provide flexible applications for multiple work environments.
- Distributing report files will increase the size of your application package.

Creating a Report file

When a report is created in the stand-alone designer it is saved as an RPT file. We are going to modify the sample Employee Profile report for use with this application.

Remove any saved data so it will refresh from the database.

Add a parameter field so we can narrow down the number of employees displayed in the report.

1. Open the Crystal Reports Designer from your Start Menu.
2. Close the Startup screen and choose File Open. You will find the Employee Profile report at:

"C:\Program Files\Crystal Decisions\Crystal Reports
9\Samples\En\Reports\General Business\Employee Profile.rpt"

3. This report is created against the default ODBC Data source that Crystal Reports installs, "Xtreme Sample Database 9". There should be no password.
4. Under File in the menu, uncheck "Save Data with Report".
5. To see the Field Explorer, choose View Field Explorer. Right click on Parameter Fields and choose New. Name the parameter Employee ID. Prompting text can be "Enter the Employee ID". The value type is number. Choose OK.
6. Under the Report Menu, choose Selection Formulas then Record. Expand the Report Fields. Double click on Xtreme Sample Database 9 and expand the Employee table. Double click on Employee ID. That will bring the {Employee.Employee ID} field down into the Formula box. Type "=" . Then expand the report fields the same way and double click on ?Employee ID. It will move down to complete the selection formula.

```
{Employee.Employee ID} = {?Employee ID}
```

7. Save and close the selection formula.
8. Run the report to be sure it is successful. Save the report to: "C:\ Employee Profile.rpt" (or a path of your choice.)

Now that the report is saved as a disk file, you can set up a PowerBuilder project that will load this report into the Crystal Reports Viewer Control.

Creating the PowerBuilder application

To create a new PowerBuilder application to load the report into the Crystal Viewer Control:

1. Open a new workspace in PowerBuilder. Click the File menu and then choose Workspace. Name your Workspace and choose Save.
2. Open an application. Click the File menu and then choose New Target. Name the application and it will name the library as well.
3. Create the variables you will need for Crystal Reports. They will need to be seen from a number of functions so they should be global or instance variables.

```
global variables
OLEObject g_ole_crx_application
OLEObject g_ole_crx_report
OLEObject g_ole_crx_connection_info
OLEObject g_ole_crx_export_options

string gs_gs_rpt_filename
integer gi_return
end variables
```

4. Again click File and this time choose New PB Object. From the dialog box choose a Window object. In the Properties box, title the Window object crPreview. Save the Window object as w_crpreview.

Add the Crystal Viewer Control

5. To add the Crystal Reports Viewer Control to the Window form, click the Insert Control 9 and click OK. Click on your Window to place this control where you want menu. Choose Control and then click OLE.... In the Insert Object dialog box go to the Insert Control Tab. Scroll down and choose Crystal Report Viewer it displayed.
6. In the Property box rename the viewer from ole_1 to ole_crviewer.
7. Open the code window for the Window object, w_crpreview. Create an OLE object and connect it to the Crystal Application object. Gi_return an error if it fails.

The ProgID for Crystal Reports 10 is
CrystalRuntime.Application.10.

```
gi_return =
g_ole_crx_application.ConnectToNewObject('CrystalRuntime.Ap
plication.9')
if gi_return < 0 then
    MessageBox("Error", "Did not connect to Crystal
Application Object")
return
```

```
end if
```

8. Use the Application object to load the report. OpenReport takes the full path to the report and "1" to open it as a temporary copy rather exclusively.

```
gs_rpt_filename = "C:\ Employee Profile.rpt"
```

```
g_ole_crx_report =  
g_ole_crx_application.OpenReport(gs_rpt_filename, 1)
```

Logon to the Database

Database connectivity has changed in Crystal Reports 9.0. The older DatabaseTable method SetLogonInfo has been replaced by the ConnectionProperties collection, which has different properties depending on the database driver being used.

9. In the code create the ConnectionProperties collection. Delete the connection properties used to create the report. This is recommended when using reports created in an earlier version of Crystal Reports.

```
g_ole_crx_connection_info =  
g_ole_crx_application.database.tables[1].ConnectionProperti  
es  
g_ole_crx_connection_info.deleteAll
```

10. Then add the new connection properties. For the report we created we need only the DSN property. The database remains unchanged from the report.

```
g_ole_crx_connection_info.add("DSN", "Xtreme Sample  
Database 9")
```

For other types of connection where there is a database schema and security you would need the DSN, the Database, User ID and Password properties

```
g_ole_crx_connection_info.add("Database", "Xtreme")  
g_ole_crx_connection_info.add("User ID", "UID")  
g_ole_crx_connection_info.add("Password", "PWD")
```

11. Then verify the database to be sure Crystal sees the data as it is now and not as it was when the report was created.

```
g_ole_crx_application.database.Verify
```

Place four command buttons on the Window form. The first, title "Set Parameters", the second "Preview"; the third, "Export"; and the fourth, "Print".

Setting Parameters

Crystal Reports parameters will prompt for the value needed without any code being created. Once the application is complete, preview the report without clicking on the "Set Parameter" button and you will get the default prompt.

Place the code for the Report parameter field into the script behind Command Button 1.

You can pass the parameter by the index of the field in the order it appears in the designer.

```
g_ole_crx_application.ParameterFields[1].AddCurrentValue(3)
```

There is also a method to pass the parameter value by name using the "GetItemByName" property of the parameter field.

```
g_ole_crx_application.ParameterFields.GetItemByName("Employee ID").AddCurrentValue(4)
```

Preview the Report

When previewing the report in the Crystal Reports Viewer, you need to pass the report object to the viewer, set any properties of the viewer you want to use and then Preview the report.

1. In code behind Command Button 2, pass the report to the crviewer.

```
ole_crviewer.object.ReportSource(g_ole_crx_application)
```

2. Set crviewer properties. In this case, set a print button and an export button visible on the preview window.

```
ole_crviewer.object.EnablePrintButton = True  
ole_crviewer.object.EnableExportButton = True
```

3. Preview the report.

```
ole_crviewer.object.ViewReport
```

Exporting the Report

Exporting a report can be done from the Export button on the report preview or it can be done in code using the ExportOptions of the Report object without using the viewer at all. For a summary of all the properties available in the ExportOptions collection see Crystal Reports Developer Help.

In the code behind Command Button 3, set the properties of the export format chosen and export the report. To find all the properties available in the ExportOptions collection see Crystal Reports Developer Help. The following example sets exports to a PDF file.

1. Set ExportOptions to Report Export Options.

```
g_ole_crx_export_options =  
g_ole_crx_application.ExportOptions
```

2. Set the path for the exported file

```
g_ole_crx_export_options.DiskFileName = "C:\ cr_report.pdf"
```

3. Set the Destination type to Disk

```
g_ole_crx_export_options.DestinationType = 1  
// crEDTDiskFile
```

4. Set the Format type

```
g_ole_crx_export_options.FormatType = 31  
//crEFTPortableDocFormat
```

5. Set any options specific to PDF. Set all pages

```
g_ole_crx_export_options.PDFExportAllPages = True
```

6. Or set PDFExportAllPages to False and set a page range

```
g_ole_crx_export_options.PDFFirstPageNumber = 2  
g_ole_crx_export_options.PDFLastPageNumber = 3
```

7. Export the report

```
g_ole_crx_application.Export(False)
```

You now have the file “cr_report.pdf” in your “C:\” directory.

Printing the Report

Like Exporting, printing a report can be done from the Print button on the Report preview window or it can be done in code from the Report object. To find the available printer options see Crystal Reports Developer Help.

To simply print the report to printer using the default printer on the client system and the printing properties of the report use the Report PrintOut property. It has the parameters:

```
PrintOut ([promptUser], [numberOfCopy], [collated], [startPageN],  
[stopPageN])
```

```
g_ole_crx_application.PrintOut(False)
```

1. To print to the printer setting specific properties at runtime, use the SelectPrinter method passing the printer driver, printer name and port name.

```
g_ole_crx_application.SelectPrinter("pscript5.dll", "\\TS  
Spooled Printer", "TS Printer Port")
```

2. Set any properties of the printer you want to use such as orientation or suppressing the dialog that lets you know a report is printing.

```
g_ole_crx_application.PaperOrientation = 2  
g_ole_crx_application.DisplayProgressDialog = False
```

3. Once you have set the properties you need, print the report.

```
g_ole_crx_application.PrintOut(False, 1, True, 1, 2)
```

That will print page 1 and 2 of the report in landscape mode without a dialog to say the report is printing.

Caution	Crystal Reports uses the properties available in your printer driver to format the report. That may make a difference to the look of your report from client to client. For information on the influence of printer drivers see: http://support.businessobjects.com/communityCS/TechnicalPapers/scrprinterdependency.pdf.asp
----------------	---

Closing the Window

The OLEObjects should be released when the application is complete. In the Window event close destroy each object.

```
Destroy OLEObject g_ole_crx_connection_info  
Destroy OLEObject g_ole_crx_export_options  
Destroy OLEObject g_ole_crx_application  
Destroy OLEObject g_ole_crx_report
```

Now the objects are released to be recreated for the next report.

Finding More Information

For more information and resources, refer to the product documentation and visit the support area of the web site at: www.businessobjects.com

Database Connectivity

CR_QUERY_ENGINE.pdf - This document provides information about the improvements to the query engine in Crystal Reports (CR) 9 as compared to version 8.5. This includes information about the differences in database architecture, the integrated Database Expert and new data source connections.

Printing

SCRPRINTERDEPENDENCY.pdf - This document discusses how you can design reports that print consistently across different computers. Includes workable solutions, tips, tricks, and design criteria to help you make use of the functionality of the Crystal Reports.

Knowledge Base article c2007185 – Changing Paper Source and Paper Tray

<http://support.businessobjects.com/library/kbase/articles/c2007185.asp>

Parameter Fields

RDCPARAM.pdf – Using parameter fields in the Report Design Component (RDC).

Exporting

CR8_VB_RDC_EXPORT.pdf – Exporting at runtime using Report Design Component (RDC). This is an older paper with good information but some properties are no longer valid in version 9.0. See CrystalDevHelp.chm to confirm the properties you are using.

► www.businessobjects.com

The Business Objects product and technology are protected by US patent numbers 5,555,403; 6,247,008; 6,578,027; 6,490,593; and 6,289,352. The Business Objects logo, the Business Objects tagline, BusinessObjects, BusinessObjects Broadcast Agent, BusinessQuery, Crystal Analysis, Crystal Analysis Holos, Crystal Applications, Crystal Enterprise, Crystal Info, Crystal Reports, Rapid Mart, and WebIntelligence are trademarks or registered trademarks of Business Objects SA in the United States and/or other countries. Various product and service names referenced herein may be trademarks of Business Objects SA. All other company, product, or brand names mentioned herein, may be the trademarks of their respective owners. Specifications subject to change without notice. Not responsible for errors or omissions. Copyright © 2004 Business Objects SA. All rights reserved.