

Crystal Reports

Getting Started with the Crystal Report Designer Component in Lotus Domino

Overview

This document illustrates steps required to integrate the Crystal Reports® Report Designer Component into a Lotus® Domino™ application. A number of common report-handling tasks examples are given and explained. This document is for use with Crystal Reports 9 and later in Domino R5.05 or later.

Contents

INTRODUCTION	2
ADDING THE RDC RUNTIME LIBRARY TO A DOMINO APPLICATION	2
WORKING WITH THE REPORT OBJECT	3
WORKING WITH THE CRYSTAL REPORT VIEWER CONTROL.....	4
LOGGING ON TO A DATABASE.....	5
<i>ConnectionProperty Object</i>	<i>5</i>
SETTING PARAMETER VALUES.....	5
<i>Passing a String Parameter</i>	<i>6</i>
<i>Passing a Number Parameter</i>	<i>6</i>
<i>Passing a Date/Time Parameter</i>	<i>6</i>
SETTING A FORMULA.....	6
SPECIFYING A SELECTION FORMULA.....	7
OPENING A SUBREPORT	7
APPENDIX A – CODE LISTINGS	8
<i>Log On and Preview</i>	<i>8</i>
<i>Parameter Fields</i>	<i>9</i>
<i>Formulas.....</i>	<i>11</i>
<i>Selection Formulas</i>	<i>12</i>
<i>Subreports.....</i>	<i>13</i>
CONTACTING CRYSTAL DECISIONS FOR TECHNICAL SUPPORT	15

Introduction

Crystal Reports is the most popular web and developer reporting tool in the world. It provides flexibility, ease-of-use, and comprehensive support for popular standards that enable users to build powerful reports from virtually any data source, including Lotus Notes and Domino, and deliver high-fidelity output to the Web in a variety of formats or integrated with applications.

From simple to complex reports, Crystal Reports makes it easy to present a compelling view of business data. Users can add charts, drill-down, alerting, logos, parameter prompts, interactive documents, or web content to any report. An extensive formula language gives developers full control over report formatting, complex business logic, and data selection.

As well, Crystal Reports enables developers to integrate reports directly into applications (Lotus Domino, VB, C++, etc). Crystal Reports' methods enable applications to take advantage of the same powerful report writing contained in the designer. Also, licensed users of Crystal Reports receive Royalty Free Client Runtime that allows you to distribute your application to others as needed.

NOTE

For more information regarding current runtime file requirements, see the Runtime File Requirements Help.

The Crystal Report Designer Component Automation Server (craxdr.dll - often referred to as the RDC or Report Designer Component) is designed to take advantage of several features of the Microsoft Visual Basic™ IDE. However, the RDC can be easily integrated in Lotus Domino R5.05 or later.

The RDC is broken into two main components: the Automation Server Component (craxdr9.dll) and the Crystal Report Viewer control (crviewer9.dll).

The “Working with the Crystal Report Viewer Control” section of this paper illustrates how to use these two components to preview a report. The sections beginning with “Logging on to a Database” extend this base and show how to work with the report at runtime.

NOTE

Refer to the Technical Reference Guides on the Crystal Reports CD for a thorough description of the following structures and functions.

Adding the RDC Runtime Library to a Domino Application

The craxdr.dll and crviewer.dll are available to your Domino application as long as they are present and registered on the development machine. Generally, development should occur on the machine where Crystal Reports Developer's (or Advanced) Edition is installed.

1. In Domino Designer, add a Form to the application.
2. In the programmer's pane, go to the (Declarations) handler of the Form and enter:

```
Dim App As Variant
```

```
Dim Rep As Variant
```

3. Select Create | Object and select the “Control” radio button in the “Create Object” dialog box that comes up. From the list of controls, select “Crystal Report Viewer Control.” This adds the control to the Form.
4. Click on the control and select Edit | Properties. In the “Applet” properties dialog, select the “Size object to fit window” radio button and then close the properties dialog.
5. Go to the PostOpen event of the form and enter the code:

```
' The Crystal Report Viewer Control
```

```
Dim m_Viewer As Variant
```

```
Set m_Viewer = Source.GetObject("Crystal Report Viewer  
Control 9")
```

```
' The application object
```

```
Set App = CreateObject("CrystalRuntime.Application.9")
```

These steps will add and instantiate the crviewer.dll and craxdrt.dll, which creates the groundwork for using the RDC to work reports.

NOTE	The crviewer9.dll cannot be used in versions of Domino prior to R5.05. However, the craxdrt.dll can be used in version R4.5 and higher. Used by itself, the craxdrt.dll can print and export reports, but cannot preview the report (since the crviewer.dll is used for that purpose).
-------------	--

Working with the Report Object

The Application object is the root object of the RDC object model. By creating this object, any of the other objects in the object hierarchy are accessible.

Two objects are mandatory when working with the RDC: the Application object and the Report object. Other objects may or may not be necessary when working with a particular .rpt file. (For example, if you need to connect a report to its ODBC datasource, you also need to use the Database object, the DatabaseTables collection and DatabaseTable object.)

However, at the very minimum, you need to instantiate at least an Application object and a Report object. For example:

```
Dim Ws As New NotesUIWorkspace
```

```
' The report
```

```
Dim theReport As Variant
```

```
theReport = Ws.OpenFileDialog(False, "Select a Report",  
"*.*.rpt | *.rpt", "c:\lotus\notes\data")
```

```
Set Rep = App.OpenReport(theReport(0))
```

The **OpenReport** method creates the Report object **Rep**. The **OpenReport** method of the Application object takes the path to the .rpt file and returns a **Report** object. In this example, the OpenFileDialog function of the NotesUIWorkspace class is used to get the file name. Note, since this function returns an array of file names, pass **theReport(0)**.

Working with the Crystal Report Viewer Control

The Crystal Report Viewer Control (CRViewer) is an ActiveX control that can be placed on a Form to preview a report. If your application only prints or exports reports, then there is no need to add the CRViewer in to your application.

The steps involved in adding the Viewer are outlined in steps 3 and 4, above. It is only possible to use the CRViewer on a Form.

The CRViewer has its own properties and methods that can be read and set using LotusScript code. As an example, the following code snippet demonstrates how to enable the refresh and export buttons in the Viewer. This allows end users to refresh the data in the report and export the report to another format (such as html).

```
Dim m_Viewer As Variant

Set m_Viewer = Source.GetObject("Crystal Report Viewer
Control 9")

Set App = CreateObject("CrystalRuntime.Application.9")
Set Rep = App.OpenReport("c:\lotus\notes\rep.rpt")
Set m_Viewer.ReportSource = Rep

' So that we can refresh the report data
m_Viewer.EnableRefreshButton = 1

' So that we can export the report after previewing
m_Viewer.EnableExportButton = 1

m_Viewer.ViewReport
```

The **ReportSource** property specifies the report that the CRViewer will display. The **ViewReport** method causes the Viewer (and report) to be displayed.

Logging on to a Database

To connect a report to a database use the following:

- ConnectionProperties Object property bag

ConnectionProperty Object

The other method that can be used to connect a report to a database is the **ConnectionProperties** collection of the ConnectionProperties object. The ConnectionProperty Object is a property bag that stores connection information for the report. The values for these properties can be set to connect to the current data source for the report or to change the data source for the report.

To use the **ConnectionProperties object**, first instantiate a DatabaseTable object. Then set the ConnectionProperties for the Data Source Name, User ID and Password. ConnectionProperty Object properties are different depending on the datasource being used. Please refer to the techref.pdf included on the Crystal Reports cd for a list of properties appropriate for the datasource that you are using.

CAUTION

It is possible to use a NotesName to retrieve the current user's UserName. However, be sure to use the Abbreviated form. See the example, below.

```
Set App = CreateObject("CrystalRuntime.Application.9")
Set Rep = App.OpenReport("c:\lotus\notes\report.rpt")
Dim UserName As New NotesName(Session.UserName)
Dim Password As String

Password = Ws.Prompt(PROMPT_PASSWORD, "Password", "Enter
your password" )

rep.database.tables(1).connectionproperties("Data Source
Name") = "ODBC DSN Data Source Name"

rep.database.tables(1).connectionproperties("User ID") =
UserName.Abbreviated

rep.database.tables(1).connectionproperties("Password")=
Password
```

NOTE

The Data Source Name and UserName arguments for ConnectionProperties can be found by clicking **Set Datasource Location** from the **Database** menu in the Crystal Reports Designer. The dialog screen that appears contains the members just mentioned.

Setting Parameter Values

When reports have parameters, it is often desirable to handle parameter value assignment either in a custom dialog or programmatically. To pass a parameter value to a report, use the **AddCurrentValue** method of the ParameterFieldDefinition object.

The following examples demonstrate how to pass a string parameter, a number parameter and a date/time parameter.

Passing a String Parameter

```
Set App = CreateObject("CrystalRuntime.Application.9")
Set Rep = App.OpenReport("c:\lotus\notes\report.rpt")
'Handle parameters programmatically
' the first parameter is a string representing the region
Rep.ParameterFields(1).AddCurrentValue "BC"
```

Passing a Number Parameter

```
' the second parameter is a number
Rep.ParameterFields(2).AddCurrentValue 2
```

Passing a Date/Time Parameter

```
'Handle parameters programmatically
' the third parameter is a DateTime - we'll use today's
date
Rep.ParameterFields(3).AddCurrentValue Date

' Or optionally use Datenumber()
Rep.ParameterFields(3).AddCurrentValue
(Datenumber(2003,09,13))
```

Setting a Formula

Formulas in Crystal Reports are used for a variety of tasks, such as aggregating values, conditional formatting, and even conditional text display. Additionally, the formula fields of a report are accessible using the `FormulaFieldDefinition` object. The formula text is accessed using the `Text` property of the `FormulaFieldDefinition` object. For example:

```
Set App = CreateObject("CrystalRuntime.Application.9")
Set Rep = App.OpenReport("c:\lotus\notes\report.rpt")
' Set a value for the "caption" formula
' In this case we'll use straight text
' Note: when specifying text - enclose the text in
' single quotes
Rep.FormulaFields(1).Text = " 'Crystal Decisions' "
```

Specifying a Selection Formula

Selection Formulas specify criteria that apply a filter on the number of records in the report. To access the selection formula for a report, use the **RecordSelectionFormula** property of the Report object. The Selection Formula is specified using Crystal Reports formula syntax. The following example demonstrates how to select on the value of one of the fields in the report:

```
Set App = CreateObject("CrystalRuntime.Application.9")
Set Rep = App.OpenReport(theReport(0))
' Specify a RecordSelectionFormula (i.e that the Region
should be 'BC'
' Note the single quotes around the string value - if you
' neglect these, you'll get an error 515 (Error in Formula)
Rep.Recordselectionformula = "{Sales Info.Region} = 'BC' "
```

NOTE

The syntax used for the RecordSelectionFormula value (ie. The field name), can be found by testing the formula in the Crystal Reports Designer and running the report. In the report designer the formula can be found by selecting **Selection Formulas > Record** under the **Report** menu. Setting the selection formula at runtime will overwrite any selection formula set in the report during design time.

Opening a Subreport

Sometimes it is necessary to open a subreport into its own Report object; for example, when log on criteria must be specified for each table in a subreport using **connectionproperties**.

The **OpenSubreport** method of the report object is used to create a Report object with the attributes of the subreport. Then, use subreport's newly retrieved report object to set log on criteria, for example.

```
Set theSubreport = Rep.OpenSubreport("sub1")
' now we can set the log on criteria for the subreport
' note in this case we're only setting the criteria for
' the first table, if the criteria are the same for the
' other tables/view/forms/etc then the criteria will be
' propagated
theSubreport.database.tables(1).connectionproperties("Data
Source Name") = "Data Source Name"
theSubreport.database.tables(1).connectionproperties("User
ID") = UserName.Abbreviated
theSubreport.database.tables(1).connectionproperties("Passw
ord")= Password
```

Appendix A – Code Listings

The code listings above were derived from five working examples. Below are the full listings

NOTE	All of these examples are the PostOpen event handlers from a form in a Domino NSF. In addition to this code, the Form has a Crystal Report Viewer Control on it and configured in the manner described in “Adding the RDC runtime library to a Domino Application,” above. As well, the variables App and Rep are declared in the (Declarations) handler of the same Form.
-------------	--

Log On and Preview

```

Sub Postopen(Source As Notesuidocument)

    ' For file and password prompt dialogs
    Dim Ws As New NotesUIWorkspace
    Dim Session As New NotesSession

    ' The Crystal Report Viewer Control
    Dim m_Viewer As Variant
    Set m_Viewer = Source.GetObject("Crystal Report
Viewer Control 9")

    ' The application object
    Set App =
CreateObject("CrystalRuntime.Application.9")

    ' The report
    Dim theReport As Variant
    theReport = Ws.OpenFileDialog(False, "Select the
'Customer Profile' report", "*.rpt | *.rpt",
"c:\lotus\notes\data")
    Set Rep = App.OpenReport(theReport(0))

    ' The UserName
    Dim UserName As New NotesName(Session.UserName)

    ' The password
    Dim Password As String
    Password = Ws.Prompt(PROMPT_PASSWORD, "Password",
"Enter your password" )

```

```
' RDCEXamples is the name of my ODBC dsn
rep.database.tables(1).connectionproperties("DSN") =
"RDCEXamples"

rep.database.tables(1).connectionproperties("User
ID") = UserName.Abbreviated

rep.database.tables(1).connectionproperties("Password
")= Password

' Pass the report to the viewer control
Set m_Viewer.ReportSource = Rep

' So that we can drill down on the map
m_Viewer.Enabledrilldown = 1

' View the report
m_Viewer.ViewReport

End Sub
```

Parameter Fields

```
Sub Postopen(Source As Notesuidocument)

' For file and password prompt dialogs
Dim Ws As New NotesUIWorkspace
Dim Session As New NotesSession

' The Crystal Report Viewer Control
Dim m_Viewer As Variant
Set m_Viewer = Source.GetObject("Crystal Report
Viewer Control 9")

' The application object
Set App =
CreateObject("CrystalRuntime.Application.9")

' The report
Dim theReport As Variant
theReport = Ws.OpenFileDialog(False, "Select the
'cp_parameter' report", "*.rpt | *.rpt",
"c:\lotus\notes\data")
Set Rep = App.OpenReport(theReport(0))
```

```
' The UserName
Dim UserName As New NotesName(Session.UserName)

' The password
Dim Password As String
Password = Ws.Prompt(PROMPT_PASSWORD, "Password",
"Enter your password" )

' Log On
' RDCEXamples is the name of my ODBC dsn
rep.database.tables(1).connectionproperties("Data
Source Name") = "RDCEXamples"
rep.database.tables(1).connectionproperties("User
ID") = UserName.Abbreviated
rep.database.tables(1).connectionproperties("Password
")= Password

'Handle parameters programmatically
' the first parameter is a string representing the
region
Rep.ParameterFields(1).AddCurrentValue "USA"

' the second parameter is a DateTime - we'll use
today's date
Rep.ParameterFields(2).AddCurrentValue Date

' the third parameter is a number
Rep.ParameterFields(3).AddCurrentValue 2

' Pass the report to the viewer control
Set m_Viewer.ReportSource = Rep

' So that we can drill down on the map
m_Viewer.Enabledrilldown = 1

' View the report
m_Viewer.ViewReport
End Sub
```

Formulas

```
Sub Postopen(Source As NotesUIDocument)

    ' For file and password prompt dialogs
    Dim Ws As New NotesUIWorkspace
    Dim Session As New NotesSession

    ' The Crystal Report Viewer Control
    Dim m_Viewer As Variant
    Set m_Viewer = Source.GetObject("Crystal Report
Viewer Control 9")

    ' The application object
    Set App =
CreateObject("CrystalRuntime.Application.9")

    ' The report
    Dim theReport As Variant
    theReport = Ws.OpenFileDialog(False, "Select the
'cp_formula' report", "*.rpt | *.rpt",
"c:\lotus\notes\data")
    Set Rep = App.OpenReport(theReport(0))

    ' The UserName
    Dim UserName As New NotesName(Session.UserName)

    ' The password
    Dim Password As String
    Password = Ws.Prompt(PROMPT_PASSWORD, "Password",
"Enter your password" )

    ' Log On
    ' RDCEXamples is the name of my ODBC dsn
    rep.database.tables(1).connectionproperties("Data
Source Name") = "RDCEXamples"
    rep.database.tables(1).connectionproperties("User
ID") = UserName.Abbreviated
    rep.database.tables(1).connectionproperties("Password
")= Password
```

```
' Set a value for the "caption" formula
' In this case we'll use straight text
' Note: when specifying text - enclose the text in
single quotes
Rep.FormulaFields(2).Text = "'What a cool report!!'"

' Pass the report to the viewer control
Set m_Viewer.ReportSource = Rep

' So that we can drill down on the map
m_Viewer.Enabledrilldown = 1

' View the report
m_Viewer.ViewReport
End Sub
```

Selection Formulas

```
Sub Postopen(Source As Notesuidocument)

' For file and password prompt dialogs
Dim Ws As New NotesUIWorkspace
Dim Session As New NotesSession

' The Crystal Report Viewer Control
Dim m_Viewer As Variant
Set m_Viewer = Source.GetObject("Crystal Report
Viewer Control 9")

' The application object
Set App =
CreateObject("CrystalRuntime.Application.9")

' The report
Dim theReport As Variant
theReport = Ws.OpenFileDialog(False, "Select the
'Customer Profile' report", "*.rpt | *.rpt",
"c:\lotus\notes\data")
Set Rep = App.OpenReport(theReport(0))

' The UserName
```

```
Dim UserName As New NotesName(Session.UserName)

' The password
Dim Password As String
Password = Ws.Prompt(PROMPT_PASSWORD, "Password",
"Enter your password" )

' Log On
' RDCEXamples is the name of my ODBC dsn
rep.database.tables(1).connectionproperties("Data
Source Name") = "RDCEXamples"
rep.database.tables(1).connectionproperties("User
ID") = UserName.Abbreviated
rep.database.tables(1).connectionproperties("Password
")= Password

' Specify a RecordSelectionFormula (i.e that the
Country should be 'USA'
' Note the single quotes around the string value - if
you neglect these, you'll
' get an error 515 (Error in Formula)
Rep.Recordselectionformula = "{Customer.Country} =
'USA'"

' Pass the report to the viewer control
Set m_Viewer.ReportSource = Rep

' So that we can drill down on the map
m_Viewer.Enabledrilldown = 1

' View the report
m_Viewer.ViewReport
End Sub
```

Subreports

```
Sub Postopen(Source As NotesUIDocument)

' For file and password prompt dialogs
Dim Ws As New NotesUIWorkspace
```

```
Dim Session As New NotesSession

' The Crystal Report Viewer Control
Dim m_Viewer As Variant
Set m_Viewer = Source.GetObject("Crystal Report
Viewer Control 9")

' The application object
Set App =
CreateObject("CrystalRuntime.Application.9")

' The report
Dim theReport As Variant
theReport = Ws.OpenFileDialog(False, "Select the
'cp_subreport' report", "*.rpt | *.rpt",
"c:\lotus\notes\data")
Set Rep = App.OpenReport(theReport(0))

' The UserName
Dim UserName As New NotesName(Session.UserName)

' The password
Dim Password As String
Password = Ws.Prompt(PROMPT_PASSWORD, "Password",
"Enter your password" )

' Log On
' RDCEXamples is the name of my ODBC dsn
rep.database.tables(1).connectionproperties("Data
Source Name") = "RDCEXamples"
rep.database.tables(1).connectionproperties("User
ID") = UserName.Abbreviated
rep.database.tables(1).connectionproperties("Password
")= Password

Set theSubreport = Rep.OpenSubreport("sub1")
' RDCEXamples is the name of my ODBC dsn
theSubreport.database.tables(1).connectionproperties(
"Data Source Name") = "RDCEXamples"
theSubreport.database.tables(1).connectionproperties(
"User ID") = UserName.Abbreviated
theSubreport.database.tables(1).connectionproperties(
"Password")= Password
```

```
' Pass the report to the viewer control
Set m_Viewer.ReportSource = Rep

' So that we can drill down on the map
m_Viewer.Enabledrilldown = 1

' View the report
m_Viewer.ViewReport

End Sub
```

Contacting Crystal Decisions for Technical Support

Along with this document, and the *Crystal Reports User's Guide*, we recommend that you visit our Technical Support web site for further resources and sample files. For further assistance, visit us at the web sites below.

Technical Support web site:

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

Answers By Email Support:

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

Phone Support:

Tel: (604) 669-8379