Crystal Reports

Crystal Reports Automation Server - Events

Overview

This module outlines how to print reports directly to the screen using the new functionality of the Crystal Reports Automation Server. At completion of this module you will be able to respond to events in a Microsoft Visual Basic application based on actions performed on the previewed report.

The Objects (or properties) that will be specifically mentioned and worked with in this document are as follows:

Application

Report

View

EventInfo

PrintWindowOptions

NOTE	Although the SCRAS can be used in Visual Basic 4, events are not supported in that version of Visual Basic. This tutorial applies to Visual Basic 5 and 6.

Contents

Creating a one button application that will launch a report

Assuming that you have already added the Crystal Reports Automation Server to your Visual Basic project and created a report to be used within this Visual Basic application. If you are not sure how to do this, please start with the Automation-native.doc which is Part I of the Crystal Reports Automation Server tutorial.

We are now going to declare the essential Objects that we are going to use in this Visual Basic project.

Within the General Declarations portion of Visual Basic, type in the code listed below.

```
Dim CrystalApplication As CRPEAuto.Application

Dim CrystalReport As CRPEAuto.Report
```

Explanation of the code above:

In the General Declarations, we will declare the Application and Report objects.

NOTE	The reason why these objects are declared here is because they will not go out of scope before the preview window has a chance to display the report.

Now that we have declared the Application and Report objects, the next step is to add a command button to the form.

Steps to follow:

Step 1: Inserting a Command Button

1. Insert a Command Button onto the form.
2. Click F4 to go to the Properties box.
3. Change the following properties for the command button:

   Caption - Print Report'This is the title of the object that you will see on the form

   (Name) - cmdPrintReport 'This is the name of the object
4. Double click onto the command button. Place in the following code:

```
Private Sub Form_Load()

'The Application and Report objects are defined in the Form Load for

'the project because it allows for the objects

'to be loaded for as long as the form is loaded (scope).
```

```
'The following creates the Application object (which is the Crystal Reports
'Automation Server) and assigns it to CrystalApplication
'Creating this object must be done for each application that uses
'the Crystal Reports Automation Server.
'If this is to be related to using the ActiveX or OCX control, this
'following method would be thought of as simply placing the control
'onto the Visual Basic form
Set CrystalApplication = CreateObject("Crystal.CRPE.Application")

'From the Application object CrystalApplication, we invoke the OpenReport
'method.  This opens up the report and assigns it to CrystalReport1
'If this is to be related to using the ActiveX or OCX control, this
'following method would be considered as the .ReportFileName property
'you need to input the path and name of your own report in this method:
Set CrystalReport = CrystalApplication.OpenReport("D:\Crw50 Reports\shell.rpt")
End Sub

Private Sub Command1_Click()
'From the Report object CrystalReport1, we invoke the Preview method.
'This sends the report out to the screen.
'If this is to be related to using the ActiveX or OCX control, this
'following method would be considered as the .PrintReport property
CrystalReport.Preview
End Sub
```

Changing the Print Preview window options at runtime in order to show a group tree.

Now that we have set up a simple application that launches a report, we are now going to modify the existing application such that it will now change the Print Preview screen and show a group tree for the report.

Within the General Declarations portion of Visual Basic, add in the code (in bold) listed below.

```
General Declarations

'In the General Declarations, we will declare the
Application

'and Report objects.The reason why these objects are
declared here is because they will not go 'out of scope
before the preview window has a chance to display the
report.

Dim CrystalApplication As CRPEAuto.Application

Dim CrystalReport As CRPEAuto.Report


'In the General Declarations, we will declare the
PrintWindowOptions

'object.

'This object will be used for changing the 'look and feel'
of how the Print

'Preview window looks up on the screen

Dim PrintWindow As CRPEAuto.PrintWindowOptions
```

Now that we have declared the PrintWindowOptions object, the next step is to add the following code to the existing command button that's on the form.

Steps to follow:

Step 1: Updating the existing Command Button

1. Edit the existing Command Button that's on the form.
2. Click F4 to go to the Properties box.
3. Check the following properties for the command button:

   Caption - Print Report'This is the title of the object that you will see on the form

   (Name) - cmdPrintReport 'This is the name of the object
4. Double click onto the command button. Place in the following code (in bold):

   ```
   Private Sub Form_Load()

   'The Application and Report objects are defined in
   the Form Load for

   'the project because it allows for the objects

   'to be loaded for as long as the form is loaded
   (scope).

   'The following creates the Application object (which
   is the Crystal Reports
   ```

```
'Automation Server) and assigns it to
CrystalApplication
'Creating this object must be done for each
application that uses
'the Crystal Reports Automation Server.
'If this is to be related to using the ActiveX or OCX
control, this
'following method would be thought of as simply
placing the control
'onto the Visual Basic form
Set CrystalApplication =
CreateObject("Crystal.CRPE.Application")


'From the Application object CrystalApplication, we
invoke the OpenReport
'method.  This opens up the report and assigns it to
CrystalReport1
'If this is to be related to using the ActiveX or OCX
control, this
'following method would be considered as the
.ReportFileName property
Set CrystalReport =
CrystalApplication.OpenReport("D:\Crw50
Reports\shell.rpt")
End Sub


Private Sub Command1_Click()
'From the Report object CrystalReport, we set the
print window options
'at runtime to make the Print Preview of the report
to look the way we want it.
'In order to accomplish this, we will assign the
PrintWindowOptions property
'to PrintWindow.
Set PrintWindow = CrystalReport.PrintWindowOptions


'Using the HasGroupTree property, we have set the
grouptree option and the
'grouptree button to appear
PrintWindow.HasGroupTree = True


'From the Report object CrystalReport, we invoke the
Preview method.
'This sends the report out to the screen.
'If this is to be related to using the ActiveX or OCX
control, this
```

```
'following method would be considered as the
.PrintReport property

CrystalReport.Preview
```

Responding to SCRAS Events from the report in your application.

We have set up the application such that it launches a report to the window and allows us to change the way it looks through code. We are now going to modify the existing application such that it will be able to respond to an event if an event has occurred on the report preview (such as clicking on a button).

Within the General Declarations portion of Visual Basic, add in the code (in bold) listed below.

```
General Declarations

'In the General Declarations, we will declare the
Application

'and Report objects

'NOTE: The reason why these objects are declared here is
because they will not go 'out of scope before the preview
window has a chance to display the report.


Dim CrystalApplication As CRPEAuto.Application

Dim CrystalReport As CRPEAuto.Report


'In the General Declarations, we will declare the
PrintWindowOptions

'object.

'This object will be used for changing the 'look and feel'
of how the Print

'Preview window looks up on the screen

Dim PrintWindow As CRPEAuto.PrintWindowOptions


'In the General Declarations, we will declare the View and
EventInfo

'objects.

'These objects will be used for responding to events that
occur on the Print

'Preview window.

Dim CRReportView As CRPEAuto.View

Dim CREventInfo As CRPEAuto.EventInfo


'In the General Declarations, we will declare the Report
and Window
```

```
'objects which will be used in conjunction with Visual
Basic's WithEvents keyword.

'These objects will be used to respond to events that
occurred on the Print

'Preview window back to the Visual Basic application.

Public WithEvents CRReportEvents As CRPEAuto.Report

Public WithEvents WindowEvents As CRPEAuto.Window
```

Now that we have declared the View, EventInfo and Window objects, the next step is to add the following code and make ammendments to code in the existing command button that's on the form.

Steps to follow:

Step 1: Updating the existing Command Button

1. Edit the existing Command Button that's on the form.
2. Click F4 to go to the Properties box.
3. Check the following properties for the command button:

   Caption - Print Report'This is the title of the object that you will see on the form

   (Name) - cmdPrintReport 'This is the name of the object
4. Double click onto the command button. Place in the following code (in bold):

```
Private Sub Form_Load()

'The Application and Report objects are defined in the Form
Load for

'the project because it allows for the initialization of
the objects (scope)

'to be loaded for the entire app.


'The following creates the Application object (which is the
Crystal Reports

'Automation Server) and assigns it to CrystalApplication

'Creating this object must be done for each application
that uses

'the Crystal Reports Reports Automation Server.

'If this is to be related to using the ActiveX or OCX
control, this

'following method would be thought of as simply placing the
control

'onto the Visual Basic form

Set CrystalApplication =
CreateObject("Crystal.CRPE.Application")
```

```
'From the Application object CrystalApplication, we invoke
the OpenReport

'method.  This opens up the report and assigns it to
CrystalReport1

'If this is to be related to using the ActiveX or OCX
control, this

'following method would be considered as the
.ReportFileName property

Set CrystalReport = CrystalApplication.OpenReport("D:\Crw50
Reports\shell.rpt")


End Sub


Private Sub Command1_Click()

'From the Report object CrystalReport, we set the print
window options

'at runtime to make the Print Preview of the report to look
the way we want it.

'In order to accomplish this, we will assign the
PrintWindowOptions property

'to PrintWindow.

Set PrintWindow = CrystalReport.PrintWindowOptions


'Using the HasGroupTree property, we have set the grouptree
option and the

'grouptree button to appear

PrintWindow.HasGroupTree = True


'This portion of code executes the report for printing and
allows the Visual Basic app to respond to an event if an
'event has occurred on the preview screen.

'(EG.  A mouse click on a button, etc...)

Set CREventInfo = CrystalReport.EventInfo


'From the Report object CrystalReport, we invoke the
Preview method.

'This sends the report out to the screen.

'If this is to be related to using the ActiveX or OCX
control, this

'following method would be considered as the .PrintReport
property

'We set the Preview method to the View object CRReportView.

Set CRReportView = CrystalReport.Preview

Form1.Hide

'From the View object CRReportView, we invoke the Parent
method.
```

```
'This assigns any events that occur on the Print Preview
window to

'WindowEvents.

Set WindowEvents = CRReportView.Parent


End Sub


Private Sub WindowEvents_ClosePrintWindow(useDefault As
Boolean)

'If the user clicks on the exit button of the preview
button, bring the original form

'to the foreground

Form1.Show

End Sub


Private Sub WindowEvents_ExportButtonClicked(useDefault As
Boolean)

'If the user clicks onto the export button that's on the
Print Preview screen,

'print out a message to the screen that the button was
pressed.

MsgBox "You have clicked onto the Export Button"


End Sub


Private Sub WindowEvents_ShowGroup_

(ByVal GroupNameList As Variant, useDefault As Boolean,
ByVal ReportName As Variant)


'If the user clicks onto a group in the group tree, print
out a message to the screen

'that a group from the group tree was selected.

MsgBox "You have clicked onto a group in the group tree"

'Print out a message to the screen of which group was
selected from the group tree.

MsgBox GroupNameList(0)


End Sub

Private Sub WindowEvents_ZoomLevelChanging(ByVal ZoomLevel
As Integer)


'If the user changes the zoom level of the Print Preview
screen,
'print out a message to the screen that the zoom was
changed.

MsgBox "You have clicked onto the zoom level control"

End Sub
```

References

The following reference materials were used during the creation of this Tech Support document, or are recommended for further reading on subjects covered by the Topics of this Module Section.

Crystal Reports Technical Reference Manual (Techref.pdf) p.92 Crystal Reports Developer's Help (Developr.hlp)

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