

Report Designer Component (RDC)

Sample application using ADO

Overview

This module outlines how to report off ActiveX Data Objects (ADO) using the Report Designer Component (RDC). This section covers how to return an ADO recordset at runtime, display the data you are going to report off in a list box, create the field definition file (TTX file) and preview the report.

Contents

MICROSOFT'S ACTIVEX DATA OBJECTS.....	2
<i>Completing the Application:</i>	<i>4</i>
CONTACTING CRYSTAL DECISIONS FOR TECHNICAL SUPPORT	5

Microsoft's ActiveX Data Objects

The first thing to do is to reference the Microsoft ADO 1.5/2.0 Object Library. If it is not listed, go under Project, References, browse and find the msado15.dll.

Add a module to the Project, in the General Declarations add this code:

```
Public CrystalDatabase As CRAXDRT.Database
Public CrystalTables As CRAXDRT.DatabaseTables
Public CrystalTable As CRAXDRT.DatabaseTable
Public CrystalTable1 As CRAXDRT.DatabaseTable
Public adoRs As ADO.DB.Recordset

Declare Function CreateFieldDefFile Lib "p2smon.dll" (X As
Object, ByVal fieldDefFilePath$, ByVal
bOverWriteExistingFiles%) As Integer
```

Step 1: Setting the database and creating the recordset.

```
Private Sub Form_Load()
    ' the adoRs object is created and bound
    Set adoRs = CreateObject("adodb.recordset")
    'code for creating the recordset. Uses the ODBC
    datasource name
    'Craze Sample Data. Make sure that this DSN is
    properly set up.
    adoRs.Open "select * from customer",
    "database=craze.mdb;uid=;pwd=;dsn=Craze
    Sample Data", adOpenKeyset, adLockBatchOptimistic
End Sub
```

Step 2: Put four command buttons and a list box on the form.

Rename command button 1 to "Populate Recordset"

```
Private Sub Command1_Click()
    'this fills a list box with the third column, which
    for
    'Craze Sample Data happens to be Customer Name from
    Craze.mdb

    adoRs.MoveFirst
```

```
Do Until adoRs.EOF
    List1.AddItem adoRs(2)
    adoRs.MoveNext
Loop
MsgBox "RecordSet Populated"
End Sub
```

Rename command button 2 to “Create Field Definition”

```
Private Sub Command2_Click()
'call to the function that creates the field definition
file(ttx)
'only. If successful it will return a 1, if it fails it
returns
'a 0

If CreateFieldDefFile(adoRs, App.Path & "\ado.ttx", True)
<> 0 Then
    MsgBox "Field definition successfully created"
Else
    MsgBox "Failed to create field definition file"
End If

End Sub
```

Rename command button 3 to “Preview Report”

```
Private Sub Command3_Click()
'We need to set the database, database tables and database
table
'to get to the method that we need: SetPrivateData
Set CrystalDatabase = Report.Database
Set CrystalTables = CrystalDatabase.Tables
Set CrystalTable = CrystalTables.Item(1)
'SetPrivateData sets the report data to the recordset in
memory
Call CrystalTable.SetPrivateData(3, adoRs)
'Brings up the report in the Viewer Form
form2.Show
Exit Sub
```

Rename command button 4 to “EXIT”

```
Private Sub Command4_Click()  
    End  
End Sub
```

Completing the Application:

Now the basic framework of the application has been created. We need to follow a series of steps to create the field definition file as well as the report.

1. Run the project. Press the “Populate Recordset” button. It should populate the list box with Customer Names.
2. Press the “Create Field Def” button. It indicates that you have just created the field definition file (TTX file). It should reside in the app.path of the project.
3. Stop the project.
4. Now, we add the Crystal Reports 6.0 ActiveX Designer (under Project menu) to the project.
5. It will bring up the Report Gallery expert dialogue box:
 - a. Select the Standard report.
 - b. Under the first tab “Data Source”. Select the “Project...” This brings up a separate dialogue box labeled “Select Datasources”. We want to select the “Data definition” radio button then press Browse. Navigate and find the “ado.ttx” that was created from our “Create Def File” button, it should be in the app.path of the project. Press Open. Press Finish. This should bring us back to the report expert while populating the tables list box.
 - c. Go to the next tab “Fields”. Select some database fields for the Report.
 - d. You may continue to the other tabs for additional formatting or simply press “Finish” at anytime.
6. After completing the report expert, another dialogue box appears prompting the user to “Add a form containing the viewer control”. Answer “Yes” to create the Viewer form and “No” to make this the Startup object. This adds a viewer form (form2) to your project.
7. Next, we need to view the code of form2. In the General Declarations, we see a single line of code:

```
Dim Report as New CrystalReport1
```

Cut this line of code out, paste it into the module and make the following revisions:

```
Public Report as New CrystalReport1
```

8. We can now Run the project and preview the report.

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>