

# Crystal Reports Automation Server

## Remote Data Object Subreport Sample

---

### Overview

This module outlines how to create a report that contains one subreport and use Remote Data Objects (RDO) as a data source. This section covers how to return the RDO result set at runtime to both the main and subreport, display the data you are going to report off in a list box, create the report at runtime and preview the report.

### Contents

<b>MICROSOFT'S REMOTE DATA OBJECT.....</b>	<b>2</b>
<i>Completing the Application: .....</i>	<i>7</i>
<b>REFERENCES .....</b>	<b>8</b>
<b>CONTACTING CRYSTAL DECISIONS FOR TECHNICAL SUPPORT .....</b>	<b>8</b>

## Microsoft's Remote Data Object

RDO is the Com interface to the ODBC API. It is primarily used to access Remote ODBC Relational Data sources. First two things to do are reference the Microsoft RDO 2.0 Object Library and the Crystal Report Engine 6.0 Object Library. If it is not listed, go under Project, References, browse and find the MSRDO20.DLL. Add a module to the Project, in the General declarations add this code:

```
Declare Function vbEncodeIPtr Lib "p2smon.dll" (X As Object) As String
```

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

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

In the General Declarations of the form add this code:

```
Dim CrApp As CRPEAuto.Application
Dim CrRep As CRPEAuto.Report
Dim CrSub As CRPEAuto.Report
Dim RDOCn As New RDO.rdoConnection
Dim RDOCustomer As RDO.rdoResultset
Dim RDOOrders As RDO.rdoResultset
```

Step 1: Setting the database and creating the resultset.

```
Private Sub Form_Load()
    'this section binds the application object to the CrApp
    'variable. Also the RDORs object is created and bound.
    Set CrApp = CreateObject("crystal.crpe.application")

    'Open the connection to the database through ODBC
    conn$ = "UID=;PWD=;Database=Xtreme.mdb;DSN=Xtreme sample
    data;"

    With rdoCn
        .Connect = conn$
        .LoginTimeout = 5
        .CursorDriver = rdUseOdbc
        .EstablishConnection rdDriverNoPrompt, True
    End With
```

```
'Select all of the fields from the Customer table and  
'set this to the rdoCustomer (resultset object)  
SQL$ = "select * from customer"  
Set rdoCustomer = rdoCn.OpenResultset(SQL$, rdOpenDynamic)
```

```
'Select all of the fields from the Orders table and  
'set this to the rdoOrders (resultset object)  
SQL$ = "select * from orders"  
Set rdoOrders = rdoCn.OpenResultset(SQL$, rdOpenDynamic)
```

```
End Sub
```

Step 2: Put five command buttons, a listbox and a label on the form.

Label command button 2: Populate Resultset

```
Private Sub Command2_Click()  
  
'this fills a list box with the third column, which for  
'Xtreme Sample Data happens to be Customer Name from  
Xtreme.mdb  
RDOCustomer.MoveFirst  
  
Label1.Caption = RDOCustomer(2).Name  
  
Do Until RDOCustomer.EOF  
    List1.AddItem RDOCustomer(2)  
    RDOCustomer.MoveNext  
Loop  
  
MsgBox "Resultset Populated"  
  
Command3.Enabled = True  
Command1.Enabled = True  
Command4.Enabled = True  
  
End Sub
```

Label command button 3 "Create Field Definition"

```
Private Sub Command3_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. Here a ttx is created for the Customer Table and the
Orders Table. The Customer Table will be used in the Main
Report and the Orders Table will be used in the Subreport.

If CreateFieldDefFile(RDOCustomer, App.Path &
"\RDOCustomer.ttx", True) <> 0 Then

    MsgBox "Customer Table Field definition successfully
created"

Else

    MsgBox "Failed to create Customer Table field
definition file"

End If

If CreateFieldDefFile(RDOOrders, App.Path &
"\RDOOrders.ttx", True) <> 0 Then

    MsgBox "Orders Table Field definition successfully
created"

Else

    MsgBox "Failed to create Orders Table field
definition file"

End If

End Sub
```

Label command button 1 " Create Report"

```
Private Sub Command1_Click()

'call to the function for creating a report on the RDO
Resultset. This creates the Customer Tables field
definition file. Use this to create the main report. Then
Insert a Subreport based off the Orders.ttx created with
the CreateFieldDef call. Name the Subreport "Orders" for
the purposes of this demo application.

Refer to Creating the Report in ("How to Create a Crystal
Report using Data Definition Files" written by Mandeep
Jassal) for the steps to add ttx file to a report.

If CreateReportOnRuntimeDS(RDOCustomer, App.Path &
\testRDO.rpt", App.Path & "\test.ttx", True, True) <> 0
Then

    MsgBox "Customer Table Field definition created,
continue creating report. Save report before
previewing"
```

```
Else
    MsgBox "Failed to create Field Definition"
End If

'if the call returns 0 it failed to create the field
definition file (.ttx)
Command4.Enabled = True

End Sub

Label command button 4" Preview Report"

Private Sub Command4_Click()
Dim CrDB As crpeAuto.Database
Dim CrTables As crpeauto.DatabaseTables
Dim CrTable As crpeauto.DatabaseTable
Dim crSecs As crpeauto.Sections
Dim crSec As crpeauto.Section
Dim crRepObjs As crpeauto.ReportObjects
Dim crSubRepObj As crpeauto.SubreportObject
Dim cnt As Integer

On Error GoTo ErrHand:

'Now that the report has been created, we can set the
report object to it (open the report)

Set CrRep = CrApp.OpenReport(App.Path & "\test.rpt")

'Here are 2 methods to Set the datasource to the RDO
resultset. Each uses the SetPrivateData method. The
difference is how they get to the method. Comment out the
one you do not wish to test.

'Begin Method 1:

'This method steps through the object model by setting each
object until the SetPrivateData method is 'reached. We need
to set the database, database tables and database table to
get to the method that we 'need: 'SetPrivateData

Set CrDB = CrRep.Database
Set CrTables = CrDB.Tables
Set CrTable = CrTables.Item(1)
```

```
'SetPrivateData sets the main reports data to the
RDOCustomer resultset in memory

Call CrTable.SetPrivateData(3, RDORs)

'Go through each section in the main report until the
subreport is found. Then open the subreport and
SetPrivateData to the RDOOrders Resultset. Here we set the
DatabaseTable, Tables, and Table objects to the subreport

Set crSecs = crRpt.Sections
For Each crSec In crSecs
    Set crRepObjs = crSec.ReportObjects
    For cnt = 1 To crRepObjs.Count
        If crRepObjs.Item(cnt).Kind =
            crSubreportObject Then
            Set crSubRepObj = crRepObjs.Item(cnt)
            Set crSub =
                crRpt.OpenSubreport(crSubRepObj.Name)
            MsgBox "Found Subreport: " &
                crSubRepObj.Name
            Set crDB = crSub.Database
            Set crDbTables = crDB.Tables
            Set crDbTable = crDbTables.Item(1)
            'SetPrivateData sets the subreports
            data to the RDOOrders resultset in
            memory
            crDbTable.SetPrivateData 3, RDOOrders
            Exit For
        End If
    Next cnt
Next crSec

'End Method 1:

'Begin Method 2:
'Quick and Dirty method for setting private data. This
method steps through the Object model in one 'line instead
of creating and setting separate objects to reach the
desired method. Make sure the Subreport was named Orders
when created in the report.

Call crRep.Database.Tables.Item(1).SetPrivateData(3, RDORs)
Set crSub = crRep.OpenSubreport("Orders")
Call crSub.Database.Tables.Item(1).SetPrivateData(3, RDORs)

'End Method 2:
```

```
'Preview the Report.  
CrRep.Preview  
  
Exit Sub  
ErrHand:  
  
If Err.Number = 20507 Or Err.Number = 20525 Then  
    MsgBox "Error opening report, please create report  
    then preview"  
    Call Command1_Click  
Else  
    MsgBox Err.Description & Err.Number  
End If  
End Sub
```

Label command button 5: "EXIT"

```
Private Sub Command5_Click()  
Unload Me  
End Sub
```

## Completing the Application:

Now that the basic application has been created, we must follow a series of steps to successfully generate reports.

We begin first by running the project.

1. Push the button labeled "Populate Resultset". By doing so, we are simply populating the listbox and enabling the remaining buttons.
2. Next, we have the option of, either, pressing the "Create Field Def" or "Create Report" button. In this instance, however, we will press the "Create Report" button.

By pushing the "Create Field Def":

We are creating the field definition files (TTX files) only. Here we create RDOCustomer.ttx for the Main report and RDOOrders.ttx for the Subreport.

By pushing the "Create Report":

We are not only creating the TTX file RDOCustomer.ttx but, also, bringing up the Crystal Reports designer so that the report can be created at that time. Insert a subreport named "Orders" using the RDOOrders.ttx file. See "How to Create a Crystal Report using Data Definition Files" for steps to add the ttx file to the new report.

3. Proceed to create a report and Save it as test.rpt.
4. Finally, we push preview. It generates the report that we just created based off the RDO resultset.
5. Congratulations!! You just created an application that reports off a Remote Data Object.

## 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.

For more information on using the Active Data Driver and RDO see the Seagate Crystal Reports 6.0 Technical Reference, Ch. 6, p. 92 (techref.pdf) "Working with Active Data" and Seagate Crystal Reports Developer's Manual.

For a sample application of using the Automation Server and RDO, go to <http://support.crystaldecisions.com/downloads> and search for 'RDOSamp'.

## 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>