

Crystal Reports Automation Server

Data Access Objects Subreport Sample Application

Overview

This module outlines how to create a report that contains one subreport using the Crystal Report Automation Server and both using Data Access Objects (DAO) as data source. This section covers how to bind a DAO recordset at runtime to both the main report and the subreport, display the recordset data in a list box, create the report at runtime and preview the report

Contents

MICROSOFT'S DATA ACCESS OBJECTS	1
<i>Completing the Application:</i>	7
REFERENCES	8
CONTACTING CRYSTAL DECISIONS FOR TECHNICAL SUPPORT	8

Microsoft's Data Access Objects

DAO is the COM interface to the Jet database engine. It is primarily used to access ISAM (Indexed Sequential Access Method) databases and supports most common ISAM database techniques. This would be used for the Native Jet (MDB), or ISAM databases such as Btrieve, dBase, FoxPro, or Paradox. DAO 3.5 can be used as an interface with RDO to access ODBC datasources; though it is limited in its abilities and functionality with ODBC.

The first two things to do are reference the Microsoft DAO 3.51 Object Library and the Crystal Report Engine 6.0 Object Library. If it is not listed, go under Project, References, browse and find the DAO350.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 DAOCustomer As DAO.Recordset  
Dim DAOOrders As DAO.Recordset  
Dim DAODB As DAO.Database  
Dim wrkODBC As Workspace  
Dim wrkJet As Workspace
```

Step 1: Setting the database and creating the recordset.

```
Private Sub Form_Load()  
  
'this section binds the application object to the  
CrystalApplication  
  
'variable. Also the DAOCustomer object is created and  
bound. There are 2 methods shown. One uses  
  
ODBC the other Jet (Native). You can try both, just comment  
out the one you are not using.  
  
Set CrApp = CreateObject("crystal.crpe.application")  
  
'Begin Method 1:  
  
'The following can be used to open the Database Craze.mdb  
via ODBC  
  
'Create Microsoft ODBC Workspace. If you have Crystal  
Reports 7, use Xtreme.mdb  
Set wrkODBC = CreateWorkspace("", "admin", "", dbUseODBC)  
  
'set the location of your craze.mdb here  
Set DAODB = wrkODBC.OpenDatabase("Craze sample data")  
  
'code for creating the recordset. Uses the ODBC data  
source name  
  
'Craze Sample Data to create a recordset off the Customer  
Table. Make sure that this DSN is properly  
set up. If you have Crystal Reports 7, use "Xtreme Sample  
Data"
```

```

Set DAOCustomer = DAODB.OpenRecordset("Customer",
dbOpenDynamic)

'code for creating the recordset. Uses the ODBC datasource
name
'Craze Sample Data to create a recordset off the Orders
Table. Make sure that this DSN is properly
set up.
Set DAOOrders = DAODB.OpenRecordset("Orders",
dbOpenDynamic)

'End Method 1:

'Begin Method 2:
'The following can be used to open the Database Craze.mdb
Natively
'Create Microsoft Jet Workspace
Set wrkJet = CreateWorkspace("", "admin", "", dbUseJet)

'set the location of the database here
Set DAODB = wrkJet.OpenDatabase("C:\crw6032\craze.mdb")

'code for creating recordset off Customer table.
Set DAOCustomer = DAODB.OpenRecordset("Customer",
dbOpenTable)

'code for creating recordset off Orders table.
Set DAOOrders = DAODB.OpenRecordset("Orders", dbOpenTable)
'End Method 2:
End Sub

```

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

Rename command button 2 to "Populate Recordset"

```

Private Sub Command2_Click()
'this fills a list box with the third column, which for
'Craze Sample Data happens to be Customer Name from
Craze.mdb
DAOCustomer.MoveFirst
Label1.Caption = DAOCustomer(2).Name
Do Until DAOCustomer.EOF
    List1.AddItem DAOCustomer(2)

```

```
        DAOCustomer.MoveNext
```

```
    Loop
```

```
    MsgBox "RecordSet Populated"
```

```
    Command3.Enabled = True
```

```
    Command1.Enabled = True
```

```
    Command4.Enabled = True
```

```
End Sub
```

Rename command button 3 to "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(DAOCustomer, App.Path &  
"\DAOCustomer.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(DAOOrders, App.Path &  
\DAOOrders.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
```

Rename command button 1 to "Create Report"

```
Private Sub Command1_Click()
```

```
'call to the function for creating a report on the DAO  
RecordSet. 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(DAOCustomer, App.Path &
\testDAO.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
```

Rename command button 4 to "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 data source to the report.
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
daoCustomer recordset in memory

Call CrTable.SetPrivateData(3, DAORs)

'Go through each section in the main report until the
subreport is found. Then open the subreport and

SetPrivateData to the daoOrders Recordset. 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 daoOrders recordset in
 memory

 crDbTable.SetPrivateData 3, daoOrders

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

'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

Rename command button 5 to "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 Recordset”. 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 daoCustomer.ttx for the

Main report and daoOrders.ttx for the Subreport.

By pushing the “Create Report”:

We are not only creating the TTX file daoCustomer.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 daoOrders.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 DAO recordset.
5. Congratulations!! You just created an application that reports off a Data Access 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 DAO see the Crystal Reports 6.0 Technical Reference, Ch. 6, p. 92 (techref.pdf) “Working with Active Data” and Crystal Reports Developer’s Manual.

See also “How to Create a Crystal Report using Data Definition Files”.

The sample application DAOSamp.zip can be found on our website at <http://support.crystaldecisions.com/updates>

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>