

# Crystal Reports

## CDO, DAO and the Crystal ActiveX (OCX ) control

---

### Overview

The document demonstrates how to use Crystal Data Objects (CDO) and Data Access Objects (DAO) with the Crystal ActiveX (OCX) control.

### Contents

<b>INTRODUCTION .....</b>	<b>2</b>
<b>CRYSTAL DATA OBJECT (CDO).....</b>	<b>2</b>
<i>Creating a one button application that will launch a report.....</i>	<i>2</i>
Step 1: Setting our Objects and populating VB array and CDO Rowset.....	3
Step 2: Inserting a Command Button .....	3
Step 3: Creating the report.....	4
Step 4: Run the application.....	4
<b>MICROSOFT'S DATA ACCESS OBJECT (DAO) .....</b>	<b>4</b>
Step 1: Setting the database and creating the recordset.....	5
Step 2: Put five command buttons on the form.....	5
<b>REFERENCES .....</b>	<b>7</b>
<b>CONTACTING CRYSTAL DECISIONS FOR TECHNICAL SUPPORT .....</b>	<b>7</b>

## Introduction

This module outlines how to report off of memory arrays using the Crystal Data Object. Use of the Crystal Data Object follows the same principles as other Active Data sources (ADO, RDO etc). Please refer to documentation on Active Data for further information in this area.

This module outlines how to report off of memory arrays using the Data Access Object. This section covers how to create the data at runtime, display the data you are going to report off of in a list box, create the report at runtime and preview the report to window. Please refer to documentation on Active Data for further information in this area.

**NOTE**

If you are using version 6 of Seagate Crystal Reports, you will need to be using build 6.0.x.151 (Maintenance Release 1).

## Crystal Data Object (CDO)

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

**From the Crystal OCX:**

- ReportFileName
- Datafiles

**From the Crystal Data Object:**

- AddField
- AddRowSet

## Creating a one button application that will launch a report

We need to declare the Objects and Variables that we will be using in this VB Project. This is done within the **General Declarations** portion of VB so that the Objects and Variables will not go out of scope.

General Declarations:

```
Dim cdoRowSet as Object
```

```
'Declaring the Rows as 1, 2 defines 2 Records or rows and 3 columns or fields.
```

```
'VB counts the 0 values.
```

```
Dim Rows(1,2) as Variant
```

Define the Crystal Object on the Form\_Load() event.

### Step 1: Setting our Objects and populating VB array and CDO Rowset.

```
Private Sub Form_Load()

    'After creating the application object, we need to point to
    'our CDO report. We will be

    'creating this report a little bit later, for now just
    'point to a valid path

    CrystalReport1.ReportFileName = "C:\temp\cdoSample.rpt"

    'Next we set the Database and Table objects. We will need
    'when we set the

    'database to the Active Data.

    'Now we create our cdoRowSet object.

    Set cdoRowSet =
    CreateObject("CrystalDataObject.CrystalComObject")

    'The following creates the format for the cdoRowSet. We are
    'creating the columns that will be used by the cdoRowSet.

    cdoRowSet.AddField "Customer Name", vbString
    cdoRowSet.AddField "City", vbString
    cdoRowSet.AddField "Last Year's Sales", vbCurrency

    'Now we populate our VB array with data.

    Rows(0,0) = cStr("John Smith")
    Rows(0,1) = cStr("Vancouver")
    Rows(0,2) = cCur(12345.67)
    Rows(1,0) = cStr("Jane Doe")
    Rows(1,1) = cStr("Toronto")
    Rows(1,2) = cCur(89123.45)

    'Now that the array is populated, we add it to the
    'cdoRowSet

    cdoRowSet.AddRows Rows
```

### Step 2: Inserting a Command Button

1. Insert a **Command Button** onto the form.
2. Press F4 to go to the Properties box.
3. Change the following properties for the command button:  
*Caption* - Print Report  
*Name* - cmdPrintReport
4. Double-click onto the command button. Place in the following code:

```
Private Sub cmdPrintReport_Click()

    'With the cdoRowSet is populated, we direct the report
    'to use this as its datasource

    CrystalReport1.SetTablePrivateData 0,3,CDOrowset
```

*'Once the report knows where to get the data to run the report, we can preview.*

```
CrystalReport1.Action =1
```

### Step 3: Creating the report

1. Open the Crystal Reports Designer.
2. Create a new report by going File|New, then Custom.
3. Select SQL/ODBC as our datasource, then select the Active Data for Server Type.
4. At the Select Field Definition, click on New. This brings up the Database Definition Tool.
5. Enter Customer Name as our first field, choose String as Data Type, then enter a length of 50. For sample data enter a sample string, then click Add.
6. Enter City as our second field, choose String as Data Type, then enter a length of 30. Enter a sample string, then click Add.
7. Enter Last Year's Sales as our last field, choose Currency as the Data Type. Enter a sample currency, then click Add.
8. Closing the Database Definition Tool brings up the dialog asking us to save, click yes. Save this file as: cdoSample.ttx in the c:\temp\ directory.
9. Click OK to select cdoSample as our Database Definition File, then click OK at the Choose SQL Table dialog.
10. Add the three database fields to Detail Section of the report.
11. Save the report, File|Save As, as cdoSample.rpt in the c:\temp directory.

### Step 4: Run the application.

Now that the report is created, we can run the application. Preview the report by clicking the command button.

## Microsoft's Data Access Object (DAO)

First thing to do is reference the Microsoft Dao 3.5 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 daoDb As DAO.Database
Dim daoSet As DAO.Recordset
```

### Step 1: Setting the database and creating the recordset.

```
Private Sub Form_Load()
    'set the location of your craze.mdb here
    Set db =
    DBEngine.Workspaces(0).OpenDatabase("c:\crw516\craze.mdb")

    'code for creating the recordset. Uses the ODBC datasource
    name
    'Craze Sample Data. Make sure that this DSN is properly set
    up.
    Set daoSet = db.OpenRecordset("Orders", dbOpenTable)
End Sub
```

### Step 2: Put five command buttons on the form.

Label this command button: 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
    daoSet.MoveFirst
    Label1.Caption = daoSet(2).Name
    Do Until daoSet.EOF
    List1.AddItem daoSet(2)
    daoSet.MoveNext
    Loop
    MsgBox "RecordSet Populated"
    Command3.Enabled = True
    Command1.Enabled = True
    Command4.Enabled = True
End Sub
```

Label this command button "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  
If CreateFieldDefFile(daoSet, App.Path & "\ado.ttx", True)  
<> 0 Then  
MsgBox "Field definition successfully created"  
Else  
MsgBox "Failed to create field definition file"  
End If  
End Sub
```

Label this command button " Create Report"

```
Private Sub Command1_Click()  
'call to the function for creating a report on the DAO  
RecordSet  
  
If CreateReportOnRuntimeDS(daoSet, App.Path & "\test.rpt",  
App.Path & "\test.ttx", True, True) <> 0 Then  
MsgBox "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 this command button " Preview Report"

```
Private Sub Command4_Click()  
On Error GoTo ErrHand:  
  
'Now that the report has been created, we can set the  
report  
  
'object to it (open the report)  
CrystalReport1.ReportFileName = App.Path & "\test.rpt"
```

*'SetPrivateData sets the report data to the recordset in memory*

```
CrystalReport1.SetTablePrivateData 0, 3, daoSet
CrystalReport1.Action = 1
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 this button "EXIT"

```
Private Sub Command5_Click()
unload me
End Sub
```

## References

For more information on the Active Data Driver see the Seagate Crystal Reports 6.0 Technical Reference, Ch. 6, p. 100 (techref.pdf) and Seagate Crystal Reports Developer's Manual.

## 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>