

Crystal Reports 8

Using Data Definition (TTX) files to pass an ADO recordset to a Crystal Report.

Overview

This document provides information about using Data Definition (TTX) files and Active Data with Crystal Reports. This document outlines how to create a report that has a Microsoft ActiveX Data Objects (ADO) recordset passed to it at runtime.

This document is for use with Crystal Reports 8 and higher.

Contents

INTRODUCTION	2
CREATING THE DATA DEFINITION (TTX) FILE	2
<i>Data Definition Tool.....</i>	<i>3</i>
<i>Using Active Data driver functions.....</i>	<i>5</i>
CreateFieldDefFile()	5
CreateReportOnRuntimeDS()	6
CREATING THE REPORT.....	6
CREATING A DATA DEFINITION (TTX) FILE IN VB OFF AN ADO RECORDSET..	7
PASSING AN ADO RECORDSET TO THE REPORT VIA THE CRYSTAL AUTOMATION SERVER	8
PASSING AN ADO RECORDSET TO THE REPORT VIA THE RDC	9
PASSING AN ADO RECORDSET TO A SUBREPORT VIA THE CRYSTAL AUTOMATION SERVER	11
PASSING AN ADO RECORDSET TO A SUBREPORT VIA THE RDC	13
FINDING MORE INFORMATION	15
CONTACTING CRYSTAL DECISIONS FOR TECHNICAL SUPPORT	15

Introduction

Traditionally, reports created in Crystal Reports have been created from physical databases. The only limitation a report had with connecting to a database was whether the database had a native driver (direct connection) or an ODBC driver.

With the 32-bit Active Data driver, P2smon.dll (Pdsmon.dll for 16-bit), Crystal Reports has opened up to other types of data sources. Now, a report can be created so that it can report from Active Data that exists in an application.

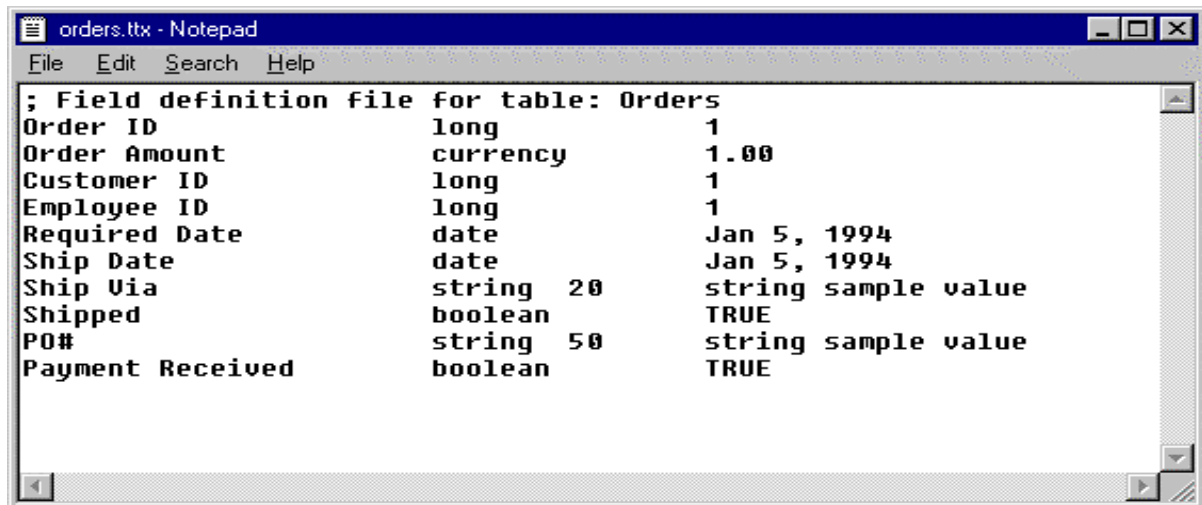
This document uses Visual Basic (VB) 6 and Crystal Reports 8 for its steps and samples. The following sections demonstrate how to create a report that reads Active Data using either the Crystal Report Engine Automation server or the Crystal Report Designer Component (RDC).

Steps 1 and 2 demonstrate how to create a Data Definition (TTX) file as well as using that TTX file to create a report. Steps 3 to 7 demonstrate passing a Microsoft ActiveX Data Objects (ADO) recordset to a Crystal Report.

1. Creating the Data Definition (TTX) file.
2. Creating the Report.
3. Creating a TTX file in VB off an ADO recordset.
4. Passing an ADO recordset to a report via the Automation Server.
5. Passing an ADO recordset to a report via the RDC.
6. Passing an ADO recordset to a subreport via the Automation Server.
7. Passing an ADO recordset to a subreport via the RDC.

Creating the Data Definition (TTX) file

A Data Definition file is a tab separated text file, with a TTX file extension. Basically, it is an outline of a table structure. Each line in a TTX file contains a field name, data type, length (if the data type is a string), and one piece of sample data. The contents of the sample Data Definition file, Orders.ttx, are shown below.



```
; Field definition file for table: Orders
Order ID          long          1
Order Amount      currency      1.00
Customer ID       long          1
Employee ID       long          1
Required Date     date          Jan 5, 1994
Ship Date         date          Jan 5, 1994
Ship Via          string 20      string sample value
Shipped           boolean       TRUE
PO#              string 50      string sample value
Payment Received  boolean       TRUE
```

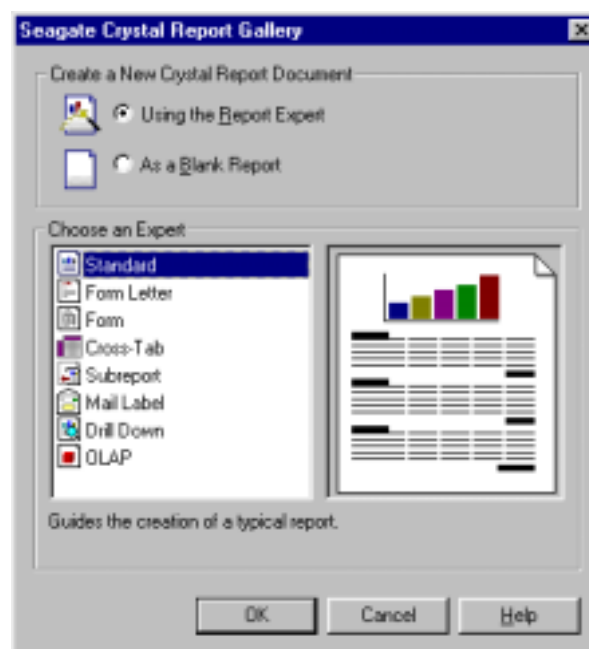
There are two ways to create a Data Definition (TTX) file:

- Data Definition Tool
- Using Active Data Driver Functions

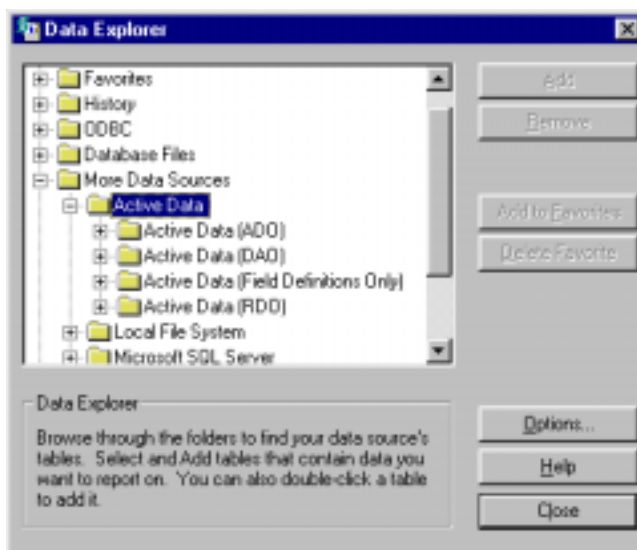
Data Definition Tool

The Data Definition Tool is a component of the Crystal Reports designer. The following steps create a TTX file using the Data Definition Tool.

1. Open the Crystal Reports designer. Click the **File** menu then click **New**.
2. The **Seagate Crystal Report Gallery** appears. Select **Using the Report Expert** and highlight the **Standard** expert then click **OK**.



3. The **Standard Report Expert** dialog box appears. Under the **Data** tab, click **Database**. The **Database Explorer** appears. Click **More Data Sources** then click **Active Data**.
4. Double click **Active Data (Field Definitions Only)**. The **Select Data Source** dialog box appears. Click **Data Definitions**.



5. The **Database Definition Tool** dialog box appears. Enter a value for the **Field Name**, **Field Type**, and **Sample Data**. Click **Add** to place the values into the grid.

Database Definition Tool - Untitled.ttx

File

Field Name : CustID

Field Type : Number

Sample Data : 1.00

Add Clear Delete

Field Name	Field Type	Length	Sample Data
CustName	String	35	Customer Name

6. Repeat step 5 for each field you wish to include.

7. Click the **File** menu then click **Save As...** to save the TTX file.

The TTX file has now been created using the Data Definition Tool.
The next step is to create a report based on this TTX file.

Using Active Data driver functions

The Active Data driver P2smon.dll (Pdsmon.dll for 16-bit) has two functions that create a TTX file based on a recordset that exists in the VB project. This is the preferred method to create a Data Definition file, as the TTX file will always match the recordset (or vice versa). The two functions available in the DLL are:

- CreateFieldDefFile()

Creates a TTX file at runtime based on a recordset.

- CreateReportOnRuntimeDS()

Creates a TTX file at runtime based on a recordset. A report (RPT) file is also created off the new TTX file, and there is an option to open the RPT file in the Crystal Reports designer.

These functions are purely for development purposes (they are not required at runtime).

NOTE

CreateReportOnRuntimeDS() does not place any fields on the report. A blank report is created instead.

The function declarations for CreateFieldDefFile() and CreateReportOnRuntimeDS() are as follows:

CreateFieldDefFile()

```
Declare Function CreateFieldDefFile Lib "p2smon.dll" (lpUnk
As Object, ByVal fileName As String, ByVal
bOverWriteExistingFile As Long) As Long
```

Parameter	Description
LpUnk	The active data source used to create the field definition file. In C or C++, this is a pointer to an IUnknown derived COM interface relating to a DAO or ADO Recordset. In Visual Basic, this is a Recordset or Rowset object.
Filename	The path and file name of the field definition file to be created.
bOverWriteExistingFile	If a field definition file already exists with the specified path and file name, this flag indicates whether or not to overwrite that file.

CreateReportOnRuntimeDS()

```
Declare Function CreateReportOnRuntimeDS Lib "p2smon.dll" (
    lpUnk As Object, ByVal reportFile As String, ByVal
    fieldDefFile As String, ByVal bOverWriteFile As Long, ByVal
    bLaunchDesigner As Long) As Long
```

Parameter	Description
LpUnk	The active data source used to create the field definition file. In C or C++, this is a pointer to an IUnknown derived COM interface relating to a DAO or ADO Recordset. In Visual Basic, this is a Recordset or Rowset object.
ReportFile	The path and file name of the report file to be created.
FieldDefFile	The path and file name of the field definition file to be created.
BoverWriteFile	If a field definition file already exists with the specified path and file name, this flag indicates whether or not to overwrite that file.
BlaunchDesigner	If True (1), Crystal Reports is launched with the newly created report file opened. Crystal Reports must be installed on the system.

NOTE

Since a TTX file is a tab-separated text file, it can be manually created or edited using Microsoft Notepad or any other text editor. This method is not recommended for creating TTX files due to possible typing errors.

Refer to the section **Creating a data definition (TTX) file in VB off an ADO recordset** for an example of using CreateFieldDefFile() and CreateReportOnRuntimeDS() in VB.

Creating the Report

The following steps describe how to create a report from a Data Definition (TTX) file.

1. Open the Crystal Reports designer. Click the **File** menu then click **New**.
2. The **Seagate Crystal Report Gallery** appears. Select **Using the Report Expert** and highlight the **Standard** expert then click **OK**.
3. The **Standard Report Expert** dialog box appears. Under the **Data** tab, click **Database**. The **Database Explorer** appears. Click **More Data Sources** then click **Active Data**.
4. Double click **Active Data (Field Definitions Only)**. The **Select Data Source** dialog box appears. Click the **Browse** button then browse to your TTX file and click **OK**.

6. This will return you to the **Data Explorer**. Under the **Active Data (Field Definitions Only)** tree is the path and filename of your TTX file. Highlight your TTX file and click **ADD** then click **Close**.
7. You are brought back to the **Standard Report Expert**. Go to the **Fields** tab and add some fields to display, then click **Finish**.

The report now displays in the Crystal Reports Designer with sample data. Save this report.

CAUTION

By default the report will be created with Saved Data.

Go to the File menu, and make sure **Save Data with Report** is unchecked, and then resave your report. This will prevent sample data from being passed to your report when you run your application.

Creating a Data Definition (TTX) file in VB off an ADO recordset

The following steps describe how to create a Data Definition (TTX) file at runtime based on an ADO recordset. The first task is to create the ADO recordset, and the second is to create the TTX file based on this ADO recordset.

To create the ADO Recordset:

1. Open a new project in Visual Basic (VB). Click the **File** menu and then click **New Project**. Select **Standard EXE** and click **OK**.
2. Click the **Project** menu, and then click **References**. Select the **Microsoft ActiveX Data Object 2.x Library** check box.

This adds a reference to the ADO object library to your project.

3. Open the code window for **Form1** and under the **General Declarations** section, type the following line of code:

```
Dim AdoRs as New ADODB.RecordSet
```

This declares an ADO recordset object in your project.

4. Under the **Form_Load** event of **Form1**, type the following lines of code:

```
Adors.Open "Select * from Customer","Xtreme Sample  
Database", adOpenKeyset, adLockBatchOptimistic
```

This creates and populates the ADO recordset object using the **Xtreme Sample Database** ODBC Data Source Name.

NOTE

The **Xtreme Sample Database** ODBC Data Source Name (DSN) is automatically created when you install Crystal Reports 8.

5. Click the **Project** menu and click **Add Module**. Click **Open** and type the following lines of code:

```
Declare Function CreateFieldDefFile Lib  
"p2smon.dll"(lpUnk As Object, ByVal fileName As String,  
ByVal bOverWriteExistingFile As Long) As Long
```

This creates an Application Program Interface (API) declaration for the CreateFieldDefFile() function contained within the P2smon.dll (Crystal Active Data driver)

6. On a new line in the **Form_Load** event for **Form1**, type the following line of code (Ensure that the folder C:\Temp exists. If it does not, modify the **FileName** parameter to specify a valid path.):

```
HResult = CreateFieldDefFile(AdoRs, "C:\temp\AdoRs.ttx", True)
```

This creates the TTX file based off the ADO Recordset object.

Confirm that the CreateFieldDefFile function returns a value of 1 (successful) and that the TTX file is created in the specified folder.

Passing an ADO recordset to the report via the Crystal Automation Server

Once the report is created using the TTX file, you need to pass your ADO recordset to the report. This section shows you how to pass an ADO recordset to a report using the Crystal Automation Server in VB 6.

The TTX file used to create the report in this project is created from the Customer table in our sample database, Xtreme.mdb. The project connects to the database using ADO and an ODBC connection to the Xtreme.mdb.

To pass a ADO recordset to a report via the Crystal Automation Server:

1. Open a new project in Visual Basic (VB). Click the **File** menu and then click **New Project**. Select **Standard EXE** and click **OK**.
2. Click the **Project** menu, and then click **References**. Select the **Microsoft ActiveX Data Object 2.x Library** and **Crystal Report Engine 8 Object Library** check boxes.

This adds an ADO object library and a Crystal Automation Server reference to your project.

3. Open the code window for **Form1** and under the **General Declarations** section, type the following lines of code:

```
Dim CrAppl As New CRPEAuto.Application
Dim CrRep As CRPEAuto.Report
Dim CrDB As CRPEAuto.Database
Dim CrTables As CRPEAuto.DatabaseTables
Dim CrTable As CRPEAuto.DatabaseTable
Dim AdoRs As New ADODB.Recordset
```

NOTE

When using the Crystal Automation Server it is important to DIM each object to avoid errors in your code.

	errors in your code.
--	----------------------

4. Under the **Form_Load** event of **Form1**, type the following lines of code:

```
`open the ADO recordset
AdoRs.Open "Select * from Customer", "Xtreme Sample
Database", adOpenDynamic, adLockBatchOptimistic

`open the report
Set CrRep = CrAppl.OpenReport("C:\temp\adors.rpt")

`set the database object to the reports database
Set CrDB = CrRep.Database

`set the databaseTables object
Set CrTables = CrDB.Tables

`set the databaseTable object to the first table in the
`report
Set CrTable = CrTables.Item(1)

`sets our ADO recordset as the data for the first table
CrTable.SetPrivateData 3, AdoRs

`Preview the report with the ADO recordset as the data
CrRep.Preview
```

5. Run this project to preview the report using the ADO recordset as data.

Passing an ADO recordset to the report via the RDC

Once the report is created using the TTX file, you need to pass your ADO recordset to the report. This section shows you how to pass an ADO recordset to a report using the Crystal Report Designer Component (RDC) in VB 6.

The TTX file used to create the report in this project is created from the Customer table in our sample database, Xtreme.mdb. The project connects to the database using ADO and an ODBC connection to the Xtreme.mdb.

To pass a ADO recordset to a report via the RDC:

1. Open a new project in Visual Basic (VB). Click the **File** menu and then click **New Project**. Select **Standard EXE** and click **OK**.
2. Click the **Project** menu, and then click **References**. Select the **Microsoft ActiveX Data Object 2.x Library** and the **Crystal Reports 8 ActiveX Designer Run Time Library** check boxes.

This adds an ADO object library and the Crystal Report Designer Component reference to your project.

3. Click the **Project** menu and then click **Components**. Select the **Crystal Report Viewer Control**.

4. Open the code window for **Form1** and under the **General Declarations** section, type the following line of code:

```
Dim CrAppl As CRAXDRT.Application
Dim CrRep As CRAXDRT.Report
Dim AdoRs As ADODB.Recordset
```

5. Under the **Form_Load** event of **Form1**, type the following lines of code:

```
'Open the ADO recordset
AdoRs.Open "Select * from Customer", "Xtreme Sample
Database", adOpenDynamic, adLockBatchOptimistic

'Open the report
Set CrRep = CrAppl.OpenReport("C:\temp\adors.rpt")

'This sets our ADO recordset to the first table in the
'report
CrRep.Database.Tables(1).SetDataSource AdoRs, 3
```

6. Insert the Crystal Reports Viewer Control on **Form1**.
7. To view the report using the Viewer, under the **Form_Load** event of **Form1**, type the following lines of code:

```
'Set the report source of the Crviewer control to our
'report object.
CrViewer1.ReportSource = CrRep

'Tell viewer control to process and preview the report
CrViewer1.ViewReport
```

8. Under the **Resize** event of **Form1**, type the following lines of code:

```
'Resize the Crystal Viewer if Form1 is resized
CrViewer1.Top = 0
CrViewer1.Left = 0
CrViewer1.Width = Form1.Width - 200
Crviewer1.Height = Form1.Height - 400
```

9. Run this project and the report will preview with the ADO recordset as the data.

Passing an ADO recordset to a subreport via the Crystal Automation Server

This section shows you how to pass an ADO recordset to a main report and subreport using the Crystal Automation Server in VB 6.

Once the main report and subreport are created using the TTX file, you need to pass your Active Data recordset to the main report and subreport.

Since you have a subreport, you need to use the **SetPrivateData** method for the subreport. You can do this by using the **OpenSubreport** method of the **Report** object.

To pass an ADO recordset to a subreport and main report via the Crystal Automation Server:

1. Open a new project in Visual Basic (VB). Click the **File** menu and then click **New Project**. Select **Standard EXE** and click **OK**.
2. Click the **Project** menu, and then click **References**. Select the **Microsoft ActiveX Data Object 2.x Library** and the **Crystal Report Engine 8 Object Library**.

This adds a reference to the ADO object library and the Crystal Automation Server to your project.

3. Open the code window for **Form1** and under the **General Declarations** section, type the following line of code:

```
Dim CrAppl as New CRPEAUTO.Application
Dim CrRep as CRPEAUTO.Report
Dim CrDatabase as CRPEAUTO.Database
Dim CrDatabaseTables as CRPEAUTO.DatabaseTables
Dim CrDatabaseTable as CRPEAUTO.DatabaseTable
Dim CrSections as CRPEAUTO.Sections
Dim CrSection as CRPEAUTO.Section
Dim CrReportObjs as CRPEAUTO.ReportObjects
Dim CrSubreportObj as CRPEAUTO.SubreportObject
Dim CrSubreport as CRPEAUTO.Report

` Recordset object for main report
Dim AdoRs as New ADODB.Recordset
` Recordset object for subreport
Dim AdoRs1 as New ADODB.Recordset
```

NOTE

When using the Crystal Automation Server it is important to DIM each object to avoid errors in your code.

4. Under the **Form_Load** event of **Form1**, type the following lines of code:

```
Dim x As Integer
Dim y As Integer

'open the ADO recordset for the main report
AdoRs.Open "Select * from Customer", "Xtreme Sample
Database", adOpenDynamic, adLockBatchOptimistic

'open the ADO recordset for the subreport
AdoRs1.Open "Select * from Customer", "Xtreme Sample
Database", adOpenDynamic, adLockBatchOptimistic

'Opens the report
Set CrRep = CrAppl.OpenReport("C:\temp\temp.rpt")

'set the database, database tables and database table
for 'the main report by using the SetDataSource method
Set CrDatabase = CrRep.Database
Set CrDatabaseTables = CrDatabase.Tables
Set CrDatabaseTable = CrDatabaseTables.Item(1)
CrDatabaseTable.SetPrivateData 3, ADORs

'Loop through each section and report object. When a
'subreport object is found, set the subreport object to
'a report object.

Set CrSections = CrRep.Sections
For x = 1 To CrSections.Count
    Set CrSection = CrSections.Item(x)
    Set CrReportObjs = CrSection.ReportObjects
    For y = 1 To CrReportObjs.Count
        If CrReportObjs.Item(y).Kind = crSubreportObject
Then
            Set CrSubreportObj = CrReportObjs.Item(y)
            Set CrSubReport =
CrRep.OpenSubreport(CrReportObjs(y).Name)
'set the database, database tables and database table
'for the subreport by using the SetDataSource method

            Set CrDatabase = crSubReport.Database
            Set CrDatabaseTables = CrDatabase.Tables
            Set CrDatabaseTable = CrDatabaseTables.Item(1)
            CrDatabaseTable.SetPrivateData 3, ADORs1
        End If
    Next
Next

CrRep.Preview
```

End Sub

5. Run this project to preview the main and subreport with the ADO recordset as the data.

Passing an ADO recordset to a subreport via the RDC

This section shows you how to pass an ADO recordset to a main report and subreport using RDC in VB 6.

Once the main report and subreport are created using the TTX file, you need to pass your ADO Recordset to the main report and subreport.

Since you have a subreport, you need to use the **SetDataSource** method for the subreport. You can do this by using the **OpenSubreport** method of the **Subreport** object.

To pass an ADO recordset to a subreport and main report via the RDC:

1. Open a new project in Visual Basic (VB). Click the **File** menu and then click **New Project**. Select **Standard EXE** and click **OK**.
2. Click the **Project** menu, and then click **References**. Select the **Microsoft ActiveX Data Object 2.x Library** and the **Crystal Reports 8 ActiveX Designer Run Time Library**.

This adds a reference to the ADO object library and the Crystal Report Designer Component reference to your project.

3. Click the **Project** menu and then click **Components**. Select the **Crystal Report Viewer Control**.
4. Insert the Crystal Reports Viewer Control on **Form1**.
5. Open the code window for **Form1** and under the **General Declarations** section, type the following line of code:

```
Dim CrAppl as New CRAXDRT.Application
Dim CrRep as CRAXDRT.Report
Dim CrDatabase as CRAXDRT.Database
Dim CrDatabaseTables as CRAXDRT.DatabaseTables
Dim CrDatabaseTable as CRAXDRT.DatabaseTable
Dim CrSections as CRAXDRT.Sections
Dim CrSection as CRAXDRT.Section
Dim CrReportObjs as CRAXDRT.ReportObjects
Dim CrSubreportObj as CRAXDRT.SubreportObject
Dim CrSubreport as CRAXDRT.Report

'ADO Recordset object for main report
Dim AdoRs as New ADODB.Recordset
```

```
'ADO Recordset object for subreport
```

```
Dim AdoRs1 as New ADODB.Recordset
```

NOTE

When using the Crystal Automation Server it is important to DIM each object to avoid errors in your code.

6. Under the **Form_Load** event of **Form1**, type the following lines of code:

```
Dim x As Integer
Dim y As Integer

'open the ADO recordset for the main report
AdoRs.Open "Select * from Customer", "Xtreme Sample
Database", adOpenDynamic, adLockBatchOptimistic

'open the ADO recordset for the subreport
AdoRs1.Open "Select * from Customer", "Xtreme Sample
Database", adOpenDynamic, adLockBatchOptimistic

'Open the report.
Set CrRep = CrAppl.OpenReport("C:\temp\temp.rpt")

'Set the database, database tables and database table for
'the main report by using the SetDataSource method.
Set CrDatabase = CrRep.Database
Set CrDatabaseTables = CrDatabase.Tables
Set CrDatabaseTable = CrDatabaseTables.Item(1)
CrDatabaseTable.SetDataSource AdoRs, 3

'Loop through each section and report object. When a
'subreport object is found, set the subreport object to a
'report object.

Set CrSections = CrRep.Sections
For x = 1 To CrSections.Count
    Set CrSection = CrSections.Item(x)
    Set CrReportObjs = CrSection.ReportObjects
    For y = 1 To CrReportObjs.Count
        If CrReportObjs.Item(y).Kind = crSubreportObject Then
            Set CrSubreportObj = CrReportObjs.Item(y)
            Set CrSubReport = CrSubreportObj.OpenSubreport

'Set the database, database tables and database table for
'the subreport by using the SetDataSource method

            Set CrDatabase = crSubReport.Database
            Set CrDatabaseTables = CrDatabase.Tables
            Set CrDatabaseTable = CrDatabaseTables.Item(1)
```

```
        CrDatabaseTable.SetDataSource AdoRs1, 3
    End If
Next
Next
CRViewer1.ReportSource = CrRep
CRViewer1.ViewReport
End Sub
```

7. Run this project to preview the main and subreport with the ADO recordset as the data.

Finding More Information

The following documents can be found on our website at
<http://support.crystaldecisions.com/docs>.

- **Scr_ttxado.pdf** – A tutorial on using Data Definition (TTX) files and Active Data with Crystal Reports 6 and 7.
- **Adobasic.pdf** – Tutorial on passing an ADO recordset to a report using the Crystal Automation Server.
- **Adosubrep.pdf** - Tutorial on passing an ADO recordset to a main and subreport using the Crystal Automation Server.
- **Ado_rdc.pdf** - Tutorial on passing an ADO recordset to a report using the Crystal Report Designer Component (RDC).
- **Ado_rdcsubrep.pdf** - Tutorial on passing an ADO recordset to a main and subreport using the Crystal Report Designer Component (RDC).
- **Rdc8_browser.exe** – A Utility to navigate through the object hierarchy of the RDC using an "Explorer tree" type interface
- **Scr8_techrefguide.exe** – A Technical Reference Guide that provides an overview of development tools available with Seagate Crystal Reports 8.
- **Scr8_devgde.exe** - Seagate Crystal Reports 8 features for developers who want to integrate reporting into web and Microsoft Windows applications

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>

