

Seagate Crystal Reports Automation Server

Seagate Crystal Reports Automation Server:

Part II - Reports via ODBC

Overview

This module outlines how to perform specific tasks using the new functionality of the OLE Automation Server (similar to the properties of the Seagate Crystal Reports Custom Control and The Report API) within Microsoft Visual Basic 5.0. At the completion of this module you will be able to create a basic application using the Seagate Crystal Reports Automation Server (OLEAUT32.DLL). The reports that will be launched from this application connect to their database via ODBC connection.

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

- Application
- Report
- Database
- DatabaseTables
- DatabaseTable
- SQLQueryString
- SetLogonInfo

Contents

Creating a simple application that previews a Seagate Crystal report.	1
Passing a SQL Query string from Visual Basic to the report at runtime....	4
References.....	5

Creating a simple application that previews a Seagate Crystal report.

NOTE: This tutorial uses Microsoft Visual Basic 5 and Seagate Crystal Reports 6. This tutorial can be used with Visual Basic 4, 5 and 6 and Seagate Crystal Reports 6 and 7.

Steps to follow:

1. Create a New project in Visual Basic 5, select the File menu and the click on the New option, Standard EXE.
2. Select the Project menu and choose the References option.
3. In the References dialog box, Choose "Crystal Report Engine 6.0 Object Library"
4. If "Crystal Report Engine 6.0 Object Library" is not one of the available references, click on the Browse button and find Cpeaut32.Dll, (32 bit) or Cpeaut16.dll (16 bit).

The above steps are equivalent to adding a Seagate Crystal Reports Custom Control to a form in an application using the Seagate Crystal Reports OCX.

Creating a one button application that will launch a report. Connecting to the reports ODBC datasource. Now that we have set up our application, such that we have added Seagate Crystal Reports' Automation Server to our Visual Basic project, we are now going to declare the essential Objects that we are going to use in the project.

Type in the code listed below within the General Declarations section of the form.

```
Dim CrystalApplication As CRPEAuto.Application
Dim CrystalReport As CRPEAuto.Report
```

'Explanation of the code above:

'In the General Declarations, we will declare the Application and Report objects.

```
Dim DbName As CRPEAuto.Database
```

```
Dim DbTables As CRPEAuto.DatabaseTables
```

```
Dim DbTable As CRPEAuto.DatabaseTable
```

'Explanation of the code above:

'In the General Declarations, we will declare the Database, DatabaseTables and

'DatabaseTable objects. These objects will be used for setting the database location at runtime.

'NOTE: The reason why these objects are declared here is because they will not go out of

'Scope before the preview window has a chance to display the report.

Now that we have declared the necessary objects, we set the value of these CrystalApplication and CrystalReport objects when the form is loaded (this way, their contents are available throughout the entire form).

```

Private Sub Form_Load ()
    'The following code creates the Application object (which is
    the Seagate Crystal Reports
    'Automation Server) and assigns it to CrystalApplication
    'Creating this object must be done for each application that
    uses
    'the Seagate Crystal Reports Automation Server.
    'If this is to be related to using the ActiveX or OCX
    control, this
    'could method would be thought of as simply placing the
    control
    'onto the Visual Basic form
    Set CrystalApplication =
    CreateObject("Crystal.CRPE.Application")

    'From the Application object CrystalApplication, we invoke
    the OpenReport
    'method. This opens up the report and assigns it to
    CrystalReport1
    'If this is to be related to using the ActiveX or OCX
    control, this
    'following method could be considered as the .ReportFileName
    property
    Set CrystalReport =
    CrystalApplication.OpenReport("C:\report1.rpt")

```

Now we can enter the code for the command button.

Steps to follow:

Step 1: Inserting a Command Button

1. Insert a Command Button onto the form.
2. Click F4 to go to the Properties box.
3. Change the following properties for the command button:

Caption - Print Report 'This is the title of the object that you will see on the form

(Name) - CmdPrintReport 'This is the name of the object

4. Double click onto the command button. Place in the following code:

```

Private Sub Command1_Click()
    'The following sets the database location for all
    tables that are used in the report
    'From the Report object CrystalReport, we take the
    current database used

```

```
'with the report and assigns it to dbName using the
Database property

Set dbName = CrystalReport.Database

'From the Database object dbName, we take all the
tables from the database

'used with the report and assign it to DbTables using
the Tables property

Set DbTables = dbName.Tables

'From the DatabaseTables collection DbTables, we
extract one item (the first table)

'using the Item property (in CRW, the tables would be
listed in order in ''Database|Set Location')

Set DbTable = DbTables.Item(1)

'From the DatabaseTable object DbTable, we pass the
ODBC connect string to the

'report using the SetLogOnInfo property which allows
the report to connect to it's

'ODBC datasource.

'When passing values from the SetLogOnInfo property,
it is necessary to pass the

' the datasource name, the database name(optional),
your user id (optional), and your

'password (optional)

Call DbTable.SetLogOnInfo( "My DataSource",
"Northwind.Mdb", "admin", "")

'From the Report object CrystalReport1, we invoke the
Preview method. This sends the report out to 'the
screen. If this is to be related to using the
ActiveX or OCX control, this following method would
'be considered as the .PrintReport property

CrystalReport.Preview

End Sub
```

Passing a SQL Query string from Visual Basic to the report at runtime

Now that we have set up the application such that it launches a report and is able to change the database location at runtime, we are now going to modify the existing application such that it will pass a new SQL Query to the report at runtime.

NOTE: When passing a SQLQuery at runtime, any record selection or group selection that is also being passed at runtime will be nullified.

Steps to follow:

Step 1: Updating the existing Command Button

1. Edit the existing Command Button that's on the form.
2. Double click onto the command button. Add the following code:

```
'From the Report object CrystalReport, we pass a SQL Query  
string at runtime  
  
'using the SQLQueryString property.
```

```
CrystalReport.SQLQueryString = "From `Customers` Customers  
Where Customers.`CompanyName` = 'Blauer See Delikatessen'"
```

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.

- Seagate Crystal Reports Technical Reference Manual (Techref.pdf) p.92
- Seagate Crystal Reports Developer's Help (Developr.hlp)
- Seagate Crystal Reports Automation Server - Reports via ODBC