Crystal Report Datasource remap

Asked 16 years, 2 months ago Modified 9 years, 10 months ago Viewed 5k times

0

I am working with Crystal Reports and MS SQL Server. I need to remap the crystal report to point to a different database on the same SQL Server. Is there an automated way of doing this, or do I have to remap for every single report? I am currently doing this by adding a new data connection, and then updating the stored procedure with the specified paramether to change database(catalog). Also, after remaping, the .asp that displays the report crashes like this:

Active Server Pages, ASP 0115 (0x80004005) A trappable error (E06D7363) occurred in an external object. The script cannot continue running.

The code is:

Set mainReportTableCollection = Session("oRpt").Database.Tables

```
For Each mnTable in mainReportTableCollection
  With mnTable.ConnectionProperties
   .Item("user ID") = "<some_login_name>"
   .Item("Password") = "<some_password>"
   .Item("DSN") = "<some_DSN>"
   .Item("Database") ="<some_Database>"
  End With
Next
```

It runs, however, if i comment out the last two assignations.

Thanks in advance.

Yours trully, Silviu.

sql-server asp-classic crystal-reports

Share
Improve this question
Follow

edited Nov 23, 2011 at 22:16
Joel Coehoorn
415k ●114 ●577 ●813

asked Oct 15, 2008 at 13:00
Silviu Preda

Which version of Crystal Reports are you using? Also, what tools do you have at hand? ASP is not ideal unless you need to do it on the fly in a web app. – Pittsburgh DBA Oct 15, 2008 at 14:07

2 Answers

Sorted by: Highest score (default)

3

You'll find hereafter the procedure I use (I simplified it on the fly, suppressing our own objects and global variables). This procedure allows to redirect a report from an original connection used at development time to the active SQL server. It is written in VB and uses 2 main objects:

1. The original report object opened through an instance of crystal report
2. An ADODB connection being the active connection (called P_currentConnection) to the current SQL server

This function (could be also a sub) is called before viewing/printing the report object in the application. It can be used when distributing reports among replicated databases where users, depending on their location, connect to different servers/databases.

```
Public Function connectReportToDatabase( _
    P_report As CRAXDRT.Report)

Dim table As CRAXDRT.DatabaseTable, _

For Each table In P_report.Database.tables

    If table.DllName <> "crdb_ado.dll" Then
        table.DllName = "crdb_ado.dll"
    End If

    table.ConnectionProperties.DeleteAll

    table.ConnectionProperties.Add "Provider", P_currentConnection.Provider
    table.ConnectionProperties.Add "Data source",
P_currentConnection.Properties("Data source").Value
    table.ConnectionProperties.Add "Database",
P_currentConnection.DefaultDatabase
    table.ConnectionProperties.Add "Integrated security",
P_currentConnection.Properties("Integrated security").Value
    table.ConnectionProperties.Add "Persist Security Info",
P_currentConnection.Properties("Persist Security Info").Value
    table.ConnectionProperties.Add "Initial Catalog",
P_currentConnection.Properties("Initial Catalog").Value

    table.SetTableLocation table.location, "",
P_currentConnection.ConnectionString

    table.TestConnectivity

Next table
```

It can be called with a procedure such as:

```
Dim crystal As CRAXDRT.Application, _
    m_report as CRAXDRT.report
```

```
Set crystal = New CRAXDRT.Application
Set m_rapport = crystal.OpenReport(nameOfTheReport & ".rpt")

connectreportToDatabase(m_report)
```

In case your report includes subreports, You might also have to redirect them to the active connection. In this case, you'll have to browse all objects in your report, check the ones that are of the report type and redirect them to the new connection. I am sure you'll have fun adding the corresponding extra lines to this original procedure.

Share Improve this answer Follow

answered Oct 20, 2008 at 8:10
Philippe Grondier
11.1k ● 3 ● 36 ● 73

0

You can get any of the info from the current report connection info. So if your not changing servers, then set the crystalServer variable to the reports current server.

```
'SET REPORT CONNECTION INFO
For i = 0 To rsource.ReportDocument.DataSourceConnections.Count - 1

rsource.ReportDocument.DataSourceConnections(i).SetConnection(crystalServer,
crystalDB, crystalUser, crystalPassword)
Next

For i = 0 To rsource.ReportDocument.Subreports.Count - 1
    For x = 0 To
rsource.ReportDocument.Subreports(i).DataSourceConnections.Count - 1

rsource.ReportDocument.OpenSubreport(rsource.ReportDocument.Subreports(i).Name).
crystalDB, crystalUser, crystalPassword)
    Next
Next
```

Share

Improve this answer

Follow

edited Feb 4, 2015 at 23:58
KyleMit ♦
29.2k ● 72 ● 501 ● 696

answered Oct 16, 2008 at 15:23
Jas
898 ● 1 ● 8 ● 12