http://www.sqlservercentral.com/articles/Reporting+Services/101183/
Printed 2013/09/30 07:06AM

Bulk Move/Rename of SSRS Reports through Web Services

By Marius Van Schalkwyk, 2013/08/02

Background

I was recently involved in a project that aimed at changing the folder structure on the SSRS Report Manager. This entailed moving over 500 reports to a completely new folder structure with over 40 different folders. Some reports also had to be renamed. Even though the report manager allows you to move several reports at a time from one folder to another, this would still be a very laborious task as the mapping between source and target folders were not exact. For example, not all the reports in a source folder will be moved to the same destination folder. Manually implementing this would be very time consuming and could potentially be prone to errors. Report Manager does not provide an easy way or renaming more than one file at a time.

The fact that this was also not the first time I encountered a requirement like this motivated me to find a quicker solution that would also be reusable. Fortunately, this is something that can be done fairly quickly through the Report Server Web Services.

The aim of this article is to demonstrate how to create a VB.NET console application that will use the Report Server Web Services to move reports from a source to a destination folder or to rename reports. The app will receive a mapping text file as input.

Solution

The following sections will demonstrate how to write a VB.NET console application that will call the Report Server Web Services calls to move and/or rename reports. This will be facilitated by discussion, screenshots and code samples.

Prerequisites

The following are prerequisites in order to implement the solution:

- SQL Server Reporting Services 2012 (in native mode)
- Visual Studio Express 2012 for Windows Desktop (Free download here)

Report Manager Setup

To demonstrate the solution, I created three folders on the report manager (***Source***, ***Target*** and ***Target2***). I then placed three reports in the ***Source*** folder (***ReportA***, ***ReportB*** and ***ReportC***). This is illustrated respectively in figure 1 and figure 2 below.

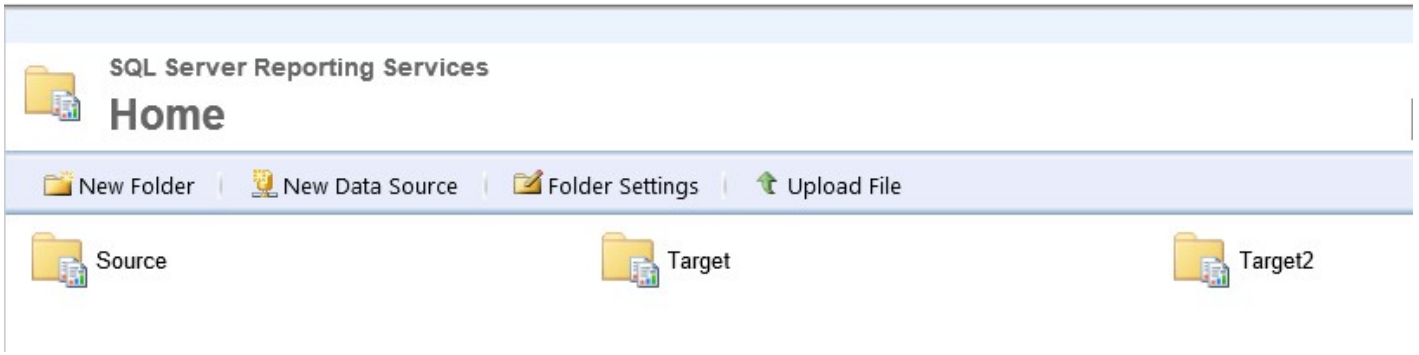

Figure 1 - Report Manager Folders

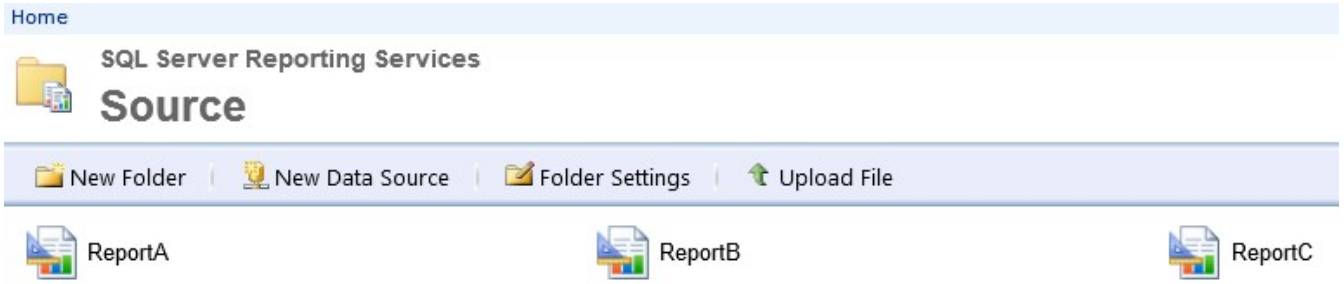

Figure 2 - Reports

Input File

The application will use a pipe delimited text file as input that will map the reports from their current location to a new location and also list required name changes. The first line in the file is a header line. As an example, consider to the file in figure 3 below; ***ReportA*** will be moved from ***/Source*** to ***/Target*** and ***ReportB*** from ***/Source*** to ***/Target2***. ***ReportC*** will not be moved, but will simply be renamed to ***ReportD***.

```
 SourcePath|DestinationPath
/Source/ReportA|/Target/ReportA
/Source/ReportB|/Target2/ReportB
/Source/ReportC|/Source/ReportD
```

Figure 3 - Mapping File

As a side note, we ran query 1 below against the ReportServer database to populate the ***SourcePath*** field in the file and dumped it into Excel (the business tool of choice :-)). The file was then distributed to our business analysts who populated the ***DestinationPath*** field according to their requirements.

```
SELECT
 [Path] AS SourcePath
 FROM
  dbo.[Catalog]
 WHERE
  Type = 2 --Type 2 being reports
```

Query 1 - T-SQL Query against ReportServer Database

Creating the VB.NET Console Application

We need to create the application to do the work in the three steps below.

Create the project

The work up to now has mainly been preparation for what is about to come… the fun stuff. Open Visual Studio Express 2012 for Windows Desktop. Create a new Visual Basic Project by clicking on ***File > New Project***. Select ***Visual Basic*** from the ***Templates*** section then select ***Console Application***. Name the project ***RSBulkMove*** and click the ***OK*** button as illustrated in figure 4 below.

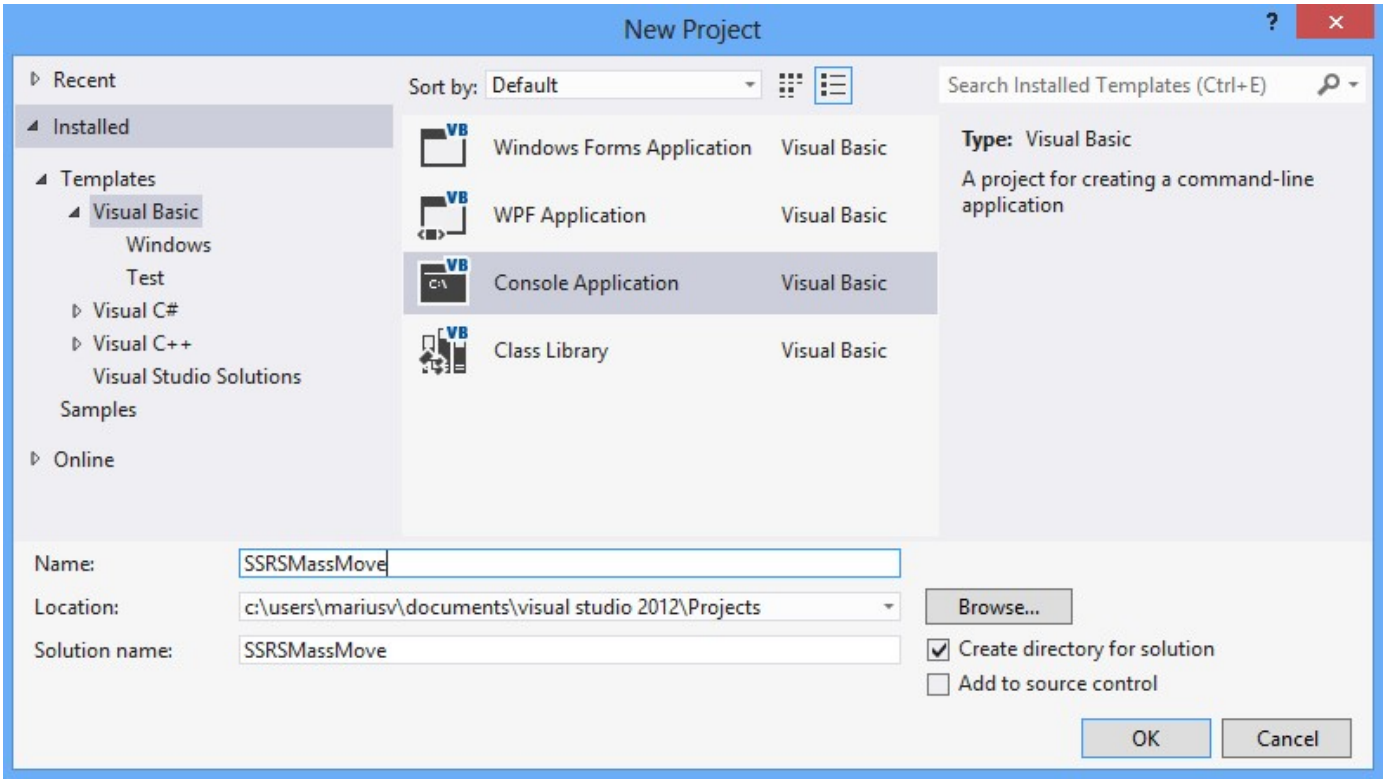

Figure 4 - New Project Dialog Box

Add the Web Service Reference

The .NET Framework provides a class called ReportingService2010 which contains methods and properties that can be used to call the Reporting Services Web Service. This works for Reporting Services configured in both native mode and SharePoint integrated mode. In this instance, the Reporting Services was running in native mode.

The ReportingService2010 class has a method called MoveItem, which moves and/or renames items. We will use this method to move/rename reports, but it can be used to move any Reporting Services items which includes reports, folders, data sources, etc. The MoveItem method accepts two string parameters; the source path and the destination path.

Before you can use the ReportingService2010 class, you need to include a reference to it in your project. You do this by clicking on ***Project > Add Service Reference***. On the ***Add Service*** dialog box, click on the ***Advanced*** button. On the ***Service Reference Settings*** dialog box, click the ***Add Web Reference…*** button. In the ***Add Web Reference*** dialog box type the following URL: `http://<servername>/ReportServer/ReportService2010.asmx?wsdl` and hit ***Enter***. Next, type the name of the namespace. I named my ***RS***, but you can call it anything you want, then click the ***Add Reference*** button. The ***Add Web Reference*** dialog box is shown in figure 5 below.

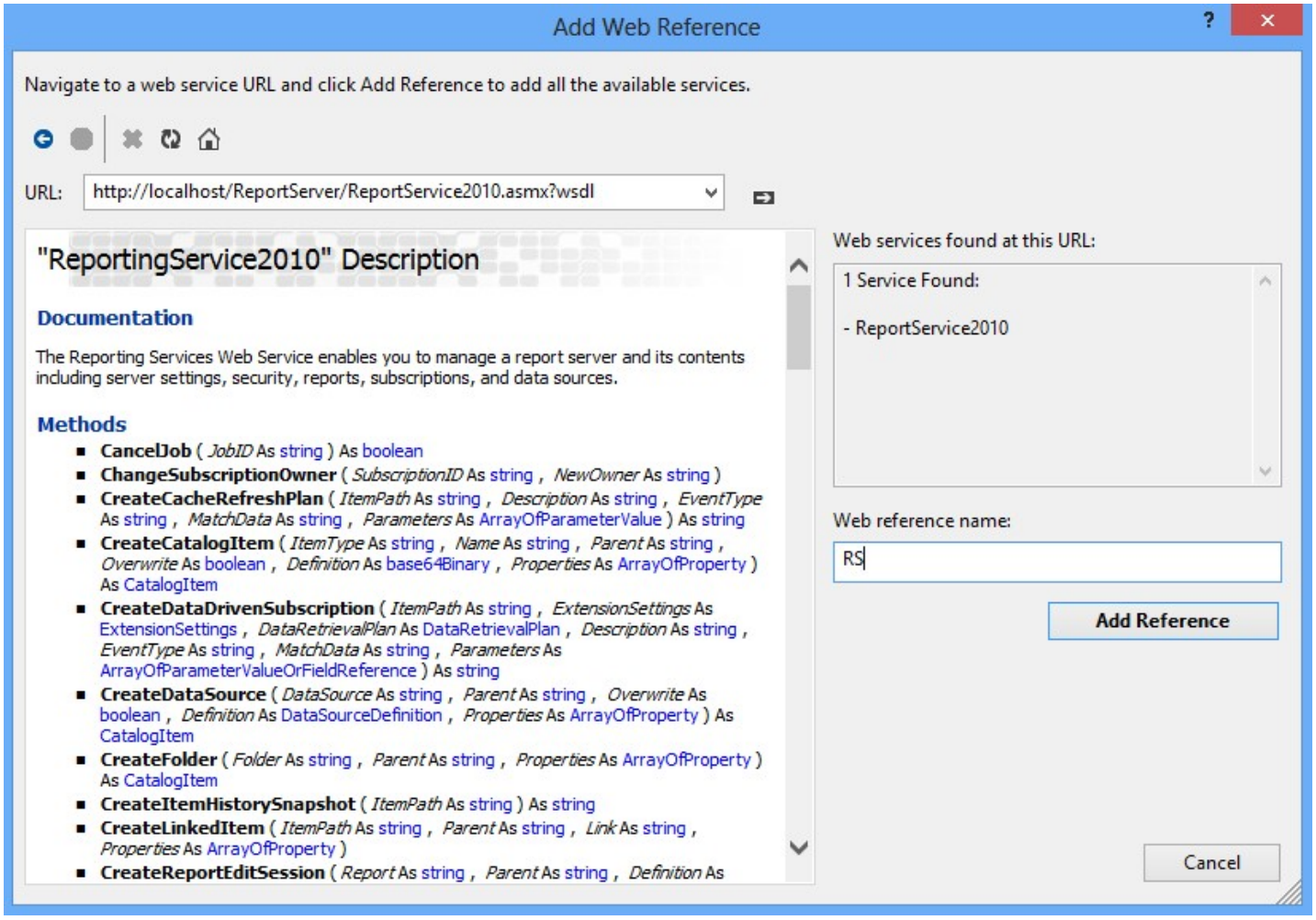

Figure 5 - Add Web Reference Dialog Box

Write the Code

Figure 6 shows the Visual Basic code that is required. The full VB solution is also avaialbe as a zip file in the resource files section at the bottom of this article. Subsequent paragraphs will explain the code.

```
1  Imports System
2  Imports System.IO
3  Imports System.Text
4  Imports System.Web
5  Imports System.Web.Services
6  Imports System.Web.Services.Protocols
7  Imports RSBulkMove.RS
8
9  Module Module1
10
11     Sub Main()
12
13         'Declare variables
14         Dim rs As New ReportingService2010()
15         Dim fileReader As StreamReader
16         Dim line As String
17         Dim fileFields() As String
18         Dim currentPath As String
19         Dim targetPath As String
20
21         rs.Credentials = System.Net.CredentialCache.DefaultCredentials
22         fileReader = My.Computer.FileSystem.OpenTextFileReader("D:\\RSFolderMapping.txt")
23
24         line = fileReader.ReadLine()    'Read and ignore file header
25         line = fileReader.ReadLine()    ' Read next line in file
26         Do
27             'Split each line into two separate variables
28             fileFields = line.Split("|")
29             currentPath = fileFields(0)
30             targetPath = fileFields(1)
31
32             rs.MoveItem(currentPath, targetPath) 'Move/rename report
33
34             line = fileReader.ReadLine() 'Read next line in file
35         Loop Until line Is Nothing 'Continue reading until there are no more lines in the file
36
37     End Sub
38
39 End Module
40
```

Figure 6 - Visual Basic Code

Line 1 – 7 imports the required namespaces. The last import statement (line 7) starts with the name of your project, followed by the name of the namespace you used while adding a reference to the web service (see Figure 5 above).

*(**Note**: It might be necessary to add a reference to System.Web.Services before you can import both System.Web.Services and System.Web.Services.Protocols. You do this by clicking on **Project > Add Reference**. Then check **System.Web.Services** from the list and click the **OK** button.)*

The variables/objects that will be used are declared in line 14 – 19. Line 14 creates an object, rs, from the ReportingService2010 class.

Lines 22 – 30 read the data from the mapping file, and then splits the line using the | delimiter and assigns each field to a variable (currentPath and targetPath).

Line 32 calls the MoveItem method, and passes the currentPath and targetPath variables as parameters. This is what moves/renames the report.

Lines 26 – 35 are repeated for each line in the mapping file (excluding the header).

Conclusion

The SQL Server Reporting Services Report Manager interface does not provide a fast and easy way to move and/or rename a large number of files. Luckily, this can be achieve easily by writing a small console application that calls the Reporting Service Web Service. This article demonstrated how this can be achieved. The Visual Studio solution shown in this article can be downloaded here.

Similarly to how the ReportingService2010 class in this application move/renamed reports in this article, it can also be used to

interrogate or change report subscriptions, change data sources, etc. A full list of properties and methods of the ReporingService2010 class and the syntax for each can be found here on the MSDN site.
