Davy Knuysen
SQL Server, .Net, ...

Home | Contact | Syndication XML | Login   40 Posts | 0 Stories | 9 Comments | 4 Trackbacks

News

Archives

Post Categories

.Net (rss)
SQL (rss)

SQL Server Reporting Services: Create and Call a Custom Assembly

In one of my previous articles I explained how to use embedded code in SQL Server Reporting Services. The possibility to add embedded code to a report, is a very powerfull feature to add some custom functionality to your report.
But sometimes it might be necessary to add some custom functionality that is too complicated for embedded code to handle efficiently, or you need to access the same function from multiple reports, or you would like to write you code in C#...
In one of these cases you should develop a custom assembly to call from your report.

Let's start with a simple example and develop a function PercentageToColor() to return a color based on a numeric value representing a percentage.

```csharp
namespace MyAssembly
{
    public class Colors
    {
        public string PercentageToColor(double percentage)
        {
            string returnValue = string.Empty;

            if (percentage < 20)
            {
                returnValue = "red";
            }
            else if (percentage < 80)
            {
                returnValue = "blue";
            }
            else
            {
                returnValue = "green";
            }

            return returnValue;
        }
    }
}
```
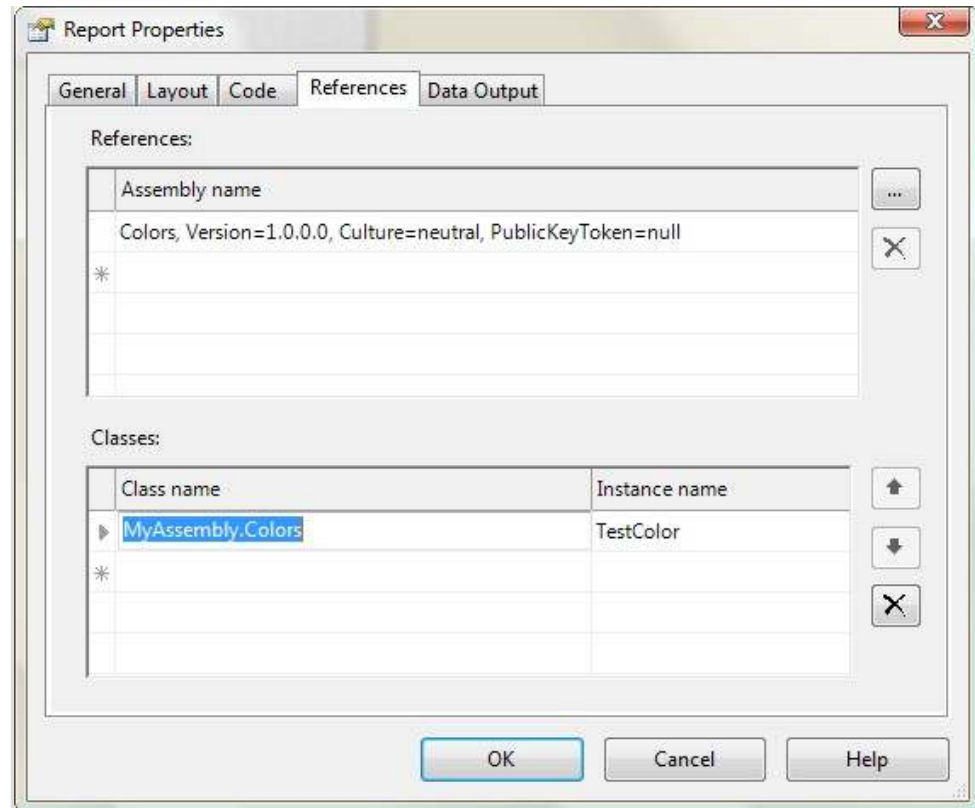
Once you've compiled your class you need to copy the resulting assembly to the directory in which it is accessible from within your report:

- To use it in the report designer, you need to copy it to:
  C:\Program Files\Microsoft Visual Studio 8\Common7\IDE\PrivateAssemblies
- To make it available for reports that have been deployed to the report server, you need to copy it to:
  C:\Program Files\Microsoft SQL Server\MSSQL.3\Reporting Services\ReportServer\bin

Before you can access your custom assembly you have to reference the assembly: Open the report properties (Menu: *Report – Report Properties*) and select the References tab.

Browse to your assembly and define a Class Name and an Instance Name. (The Class Name and the Instance Name or only for non-static methods). Make sure to prefix your class name with the assembly name.

Now you can call the methods in your assembly from your report, using an expression:

- To call a static method: =<AssemblyName>.<ClassName>.<StaticMethodName>
- To call an instance method: =Code.<InstanceName>.<PublicMethodName>

So in our example this would be: =Code.TestColor.PercentageToColor(*Fields! Percentage.Value*)

 That's it! You've just created and called a custom assembly from your report...

 In one of my future articles I will explain how to pass parameters to the class constructor and I will also tell something about SSRS and .Net Permissions...

Davy Knuysen

posted on Monday, March 26, 2007 10:51 PM

Feedback

 # re: SQL Server Reporting Services: Create and Call a Custom Assembly 5/16/2007 11:07 AM john

Top notch - very clear and concise with all of the essentials.

 # re: SQL Server Reporting Services: Create and Call a Custom Assembly 2/13/2008 4:26 PM siva

Thank you

Post Feedback

Title: | re: SQL Server Reporting Services: Create and Call a Cu

Name:

Email: (never displayed)

Url:

Comments:

Submit   ☐ Remember Me?

Powered by: SUBTEXT

Skin design by Mark Wagner, Adapted by David Vidmar

Copyright © Davy Knuysen