



Silverlight Viewer for Reporting Services 2008 Getting Started

Last modified on: December 14, 2011



Table of Content

Introduction	3
Product Prerequisites.....	3
Creating Web application using Microsoft SQL 2008/2008 R2 Reporting Services report server.....	4
Step1	4
Step2	5
Step3	7
Step4	9
Step5	9
Step6	12
Step7	14
Conclusion.....	15

Introduction

The target of the following guide is to demonstrate the basic moments of the use of Silverlight Viewer for Reporting Services 2008. It gives minimum necessary knowledge in order to start working with the component. We will examine the process of generation of web application with the use of Silverlight Viewer for Reporting Services 2008 step by step. We will consider creation and configuration of the service and at last integration of the report viewer component on the application pages. Your application can be configured to use the processing capability of Microsoft SQL Server 2008/2008 R2 Reporting Services report server.

Product Prerequisites

Product works with several software stacks.

In order to get the working sample for **Silverlight 3** the following software should be installed on your computer:

- *MS Visual Studio 2008 or higher;*
- *.NET Framework 3.5 or higher;*
- *ASP.NET 2.0 or higher;*
- *Silverlight 3.0;*
- *Microsoft Silverlight 3 Toolkit;*
- *Microsoft SQL Server 2008 Reporting Services (Developer ,Enterprise, Standard or higher edition) or Microsoft SQL Server 2008 R2 Reporting Services (Developer, Enterprise, Standard or higher edition);*
- *Microsoft SQL Reporting Services 2008 or 2008R2 Sample Reports;*
- *Silverlight Viewer for Reporting Services 1.0 or higher.*

In order to get the working sample for **Silverlight 4** the following software should be installed on your computer:

- *MS Visual Studio 2010 or higher;*
- *.NET Framework 3.5 or higher;*
- *ASP.NET 2.0 or higher;*
- *Silverlight 4.0;*
- *Microsoft Silverlight 4 Toolkit;*
- *Microsoft SQL Server 2008 Reporting Services (Developer ,Enterprise, Standard or higher edition) or Microsoft SQL Server 2008 R2 Reporting Services (Developer, Enterprise, Standard or higher edition);*
- *Microsoft SQL Reporting Services 2008 or 2008R2 Sample Reports;*
- *Silverlight Viewer for Reporting Services 2.0 or higher.*

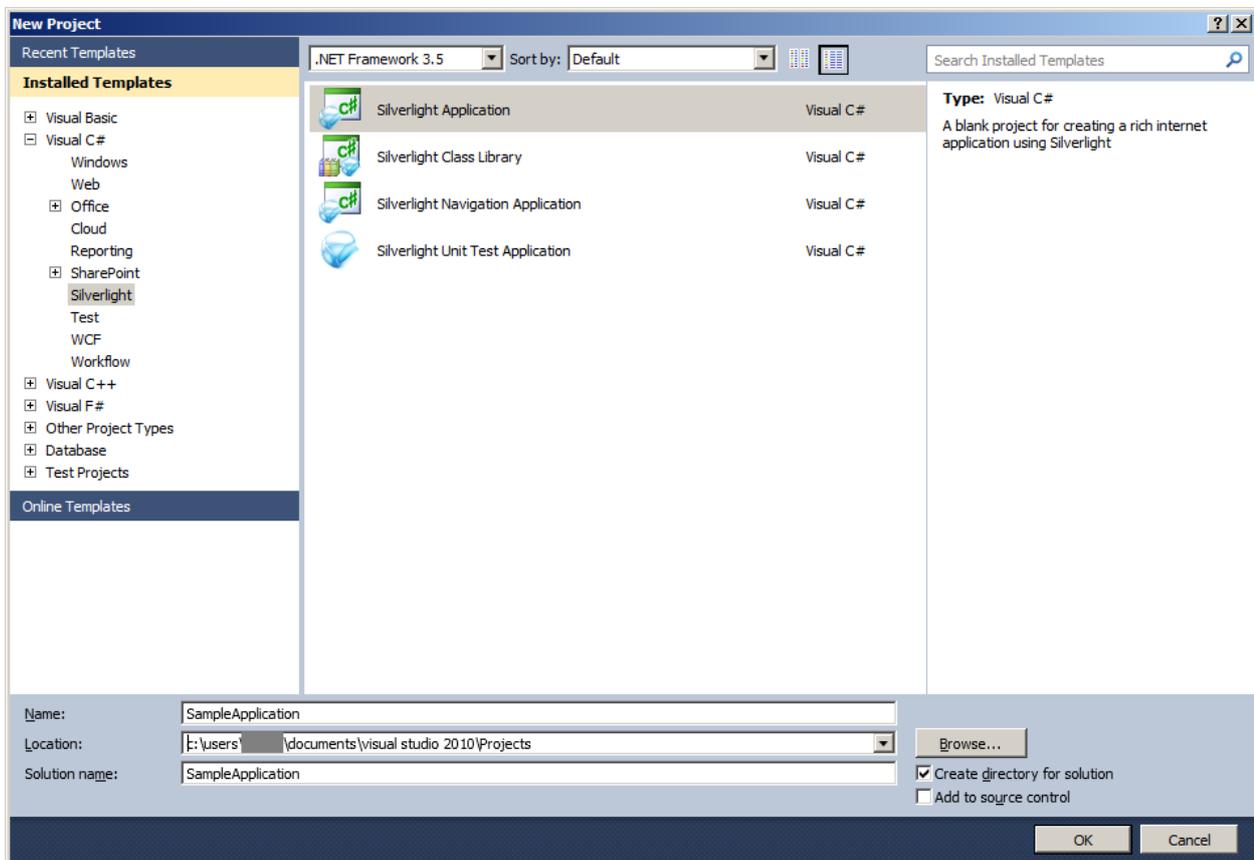
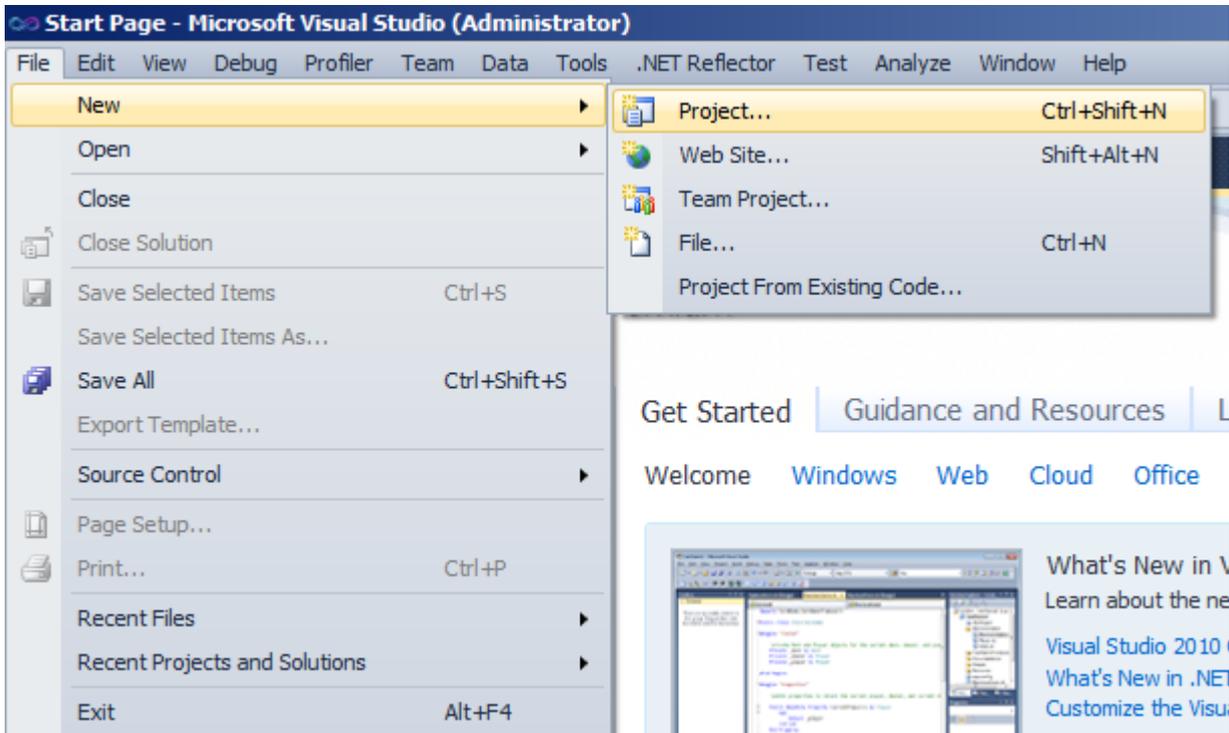
In order to get the working sample for **Silverlight 5** the following software should be installed on your computer:

- *MS Visual Studio 2010 or higher;*
- *.NET Framework 3.5 or higher;*
- *ASP.NET 2.0 or higher;*
- *Silverlight 5.0;*
- *Microsoft Silverlight 5 Toolkit;*
- *Microsoft SQL Server 2008 Reporting Services (Developer ,Enterprise, Standard or higher edition) or Microsoft SQL Server 2008 R2 Reporting Services (Developer, Enterprise, Standard or higher edition);*
- *Microsoft SQL Reporting Services 2008 or 2008R2 Sample Reports;*
- *Silverlight Viewer for Reporting Services 2.0 or higher.*

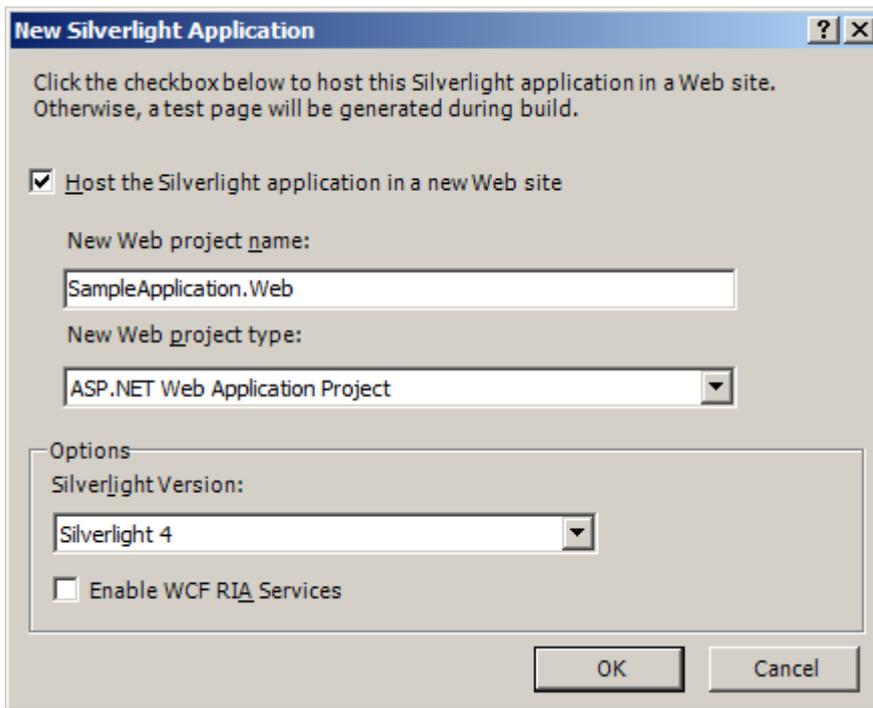
Creating Web application using Microsoft SQL 2008/2008 R2 Reporting Services report server

Step1

Create a new Silverlight Application project.



While creation of the project select the menu item "Add a new ASP.NET Web project to the solution to host Silverlight".



New Silverlight Application [?] [X]

Click the checkbox below to host this Silverlight application in a Web site. Otherwise, a test page will be generated during build.

Host the Silverlight application in a new Web site

New Web project name:

New Web project type:

Options

Silverlight Version:

Enable WCF RIA Services

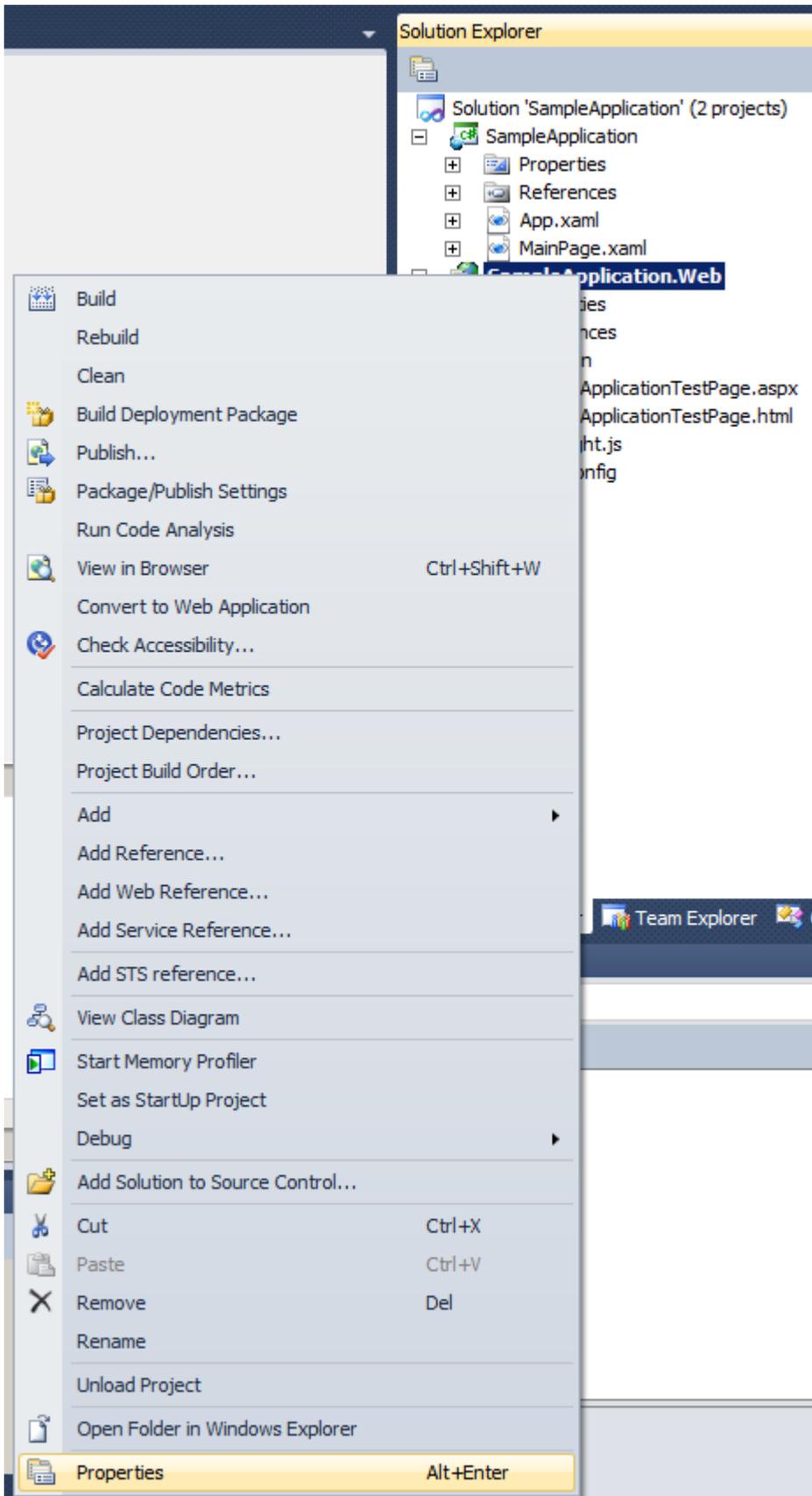
OK Cancel

Step2

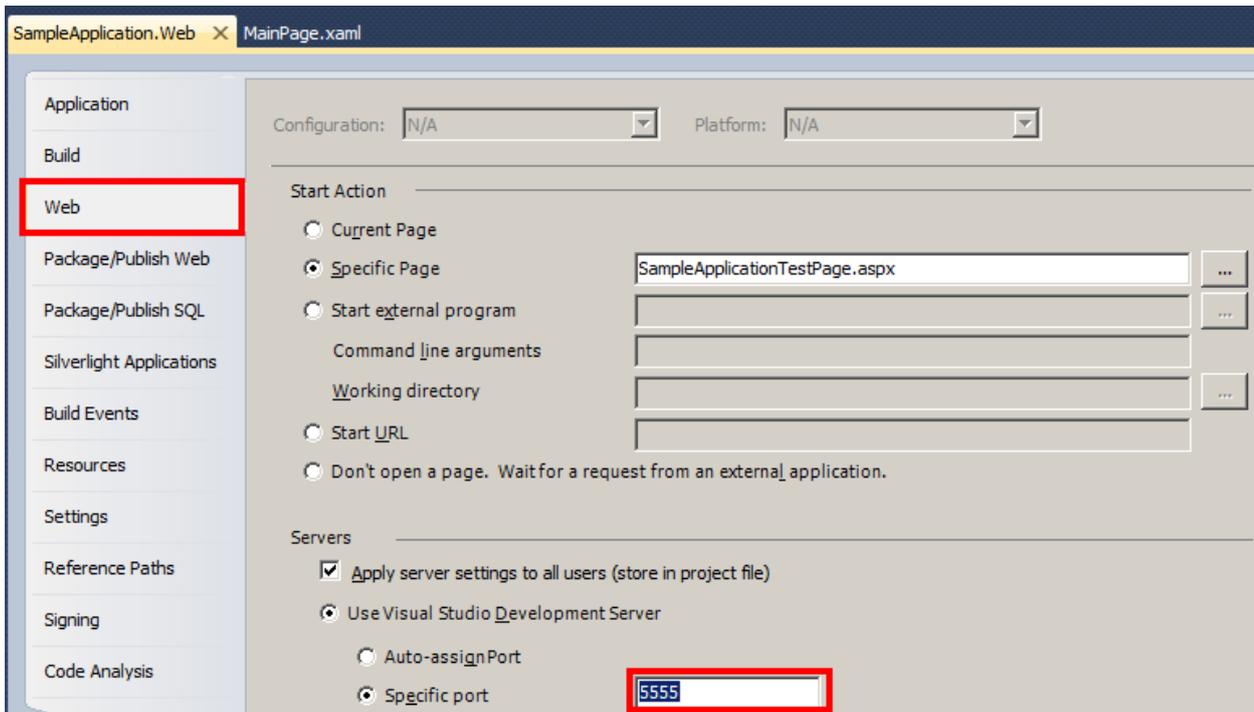
Set the web application to use static port 5555.

Set specific port to 5555 in the Web tab of the SampleApplication.Web properties (in the "Solution Explorer", item "Properties" in contextual menu of the SampleApplication.Web).

Open SampleApplication.Web properties



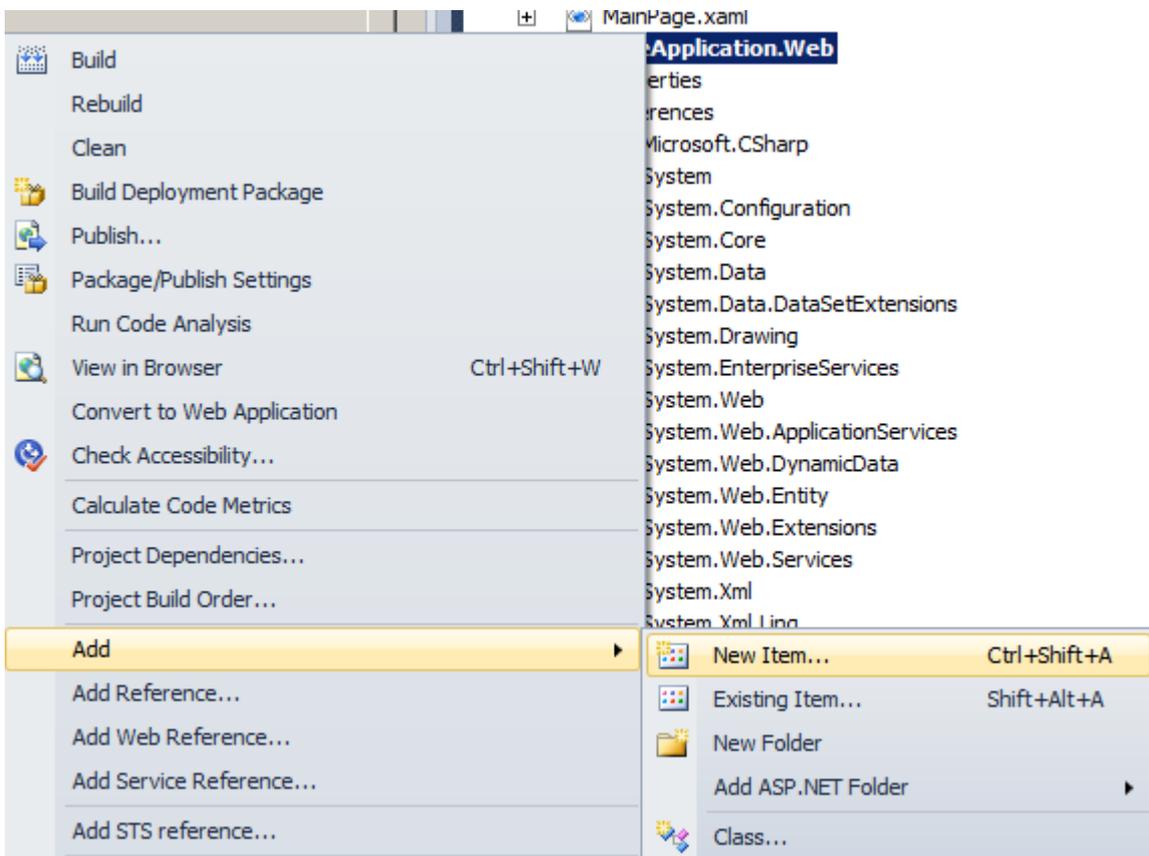
Check option Specific port and set it to 5555 in the Web tab of the SampleApplication.Web properties.



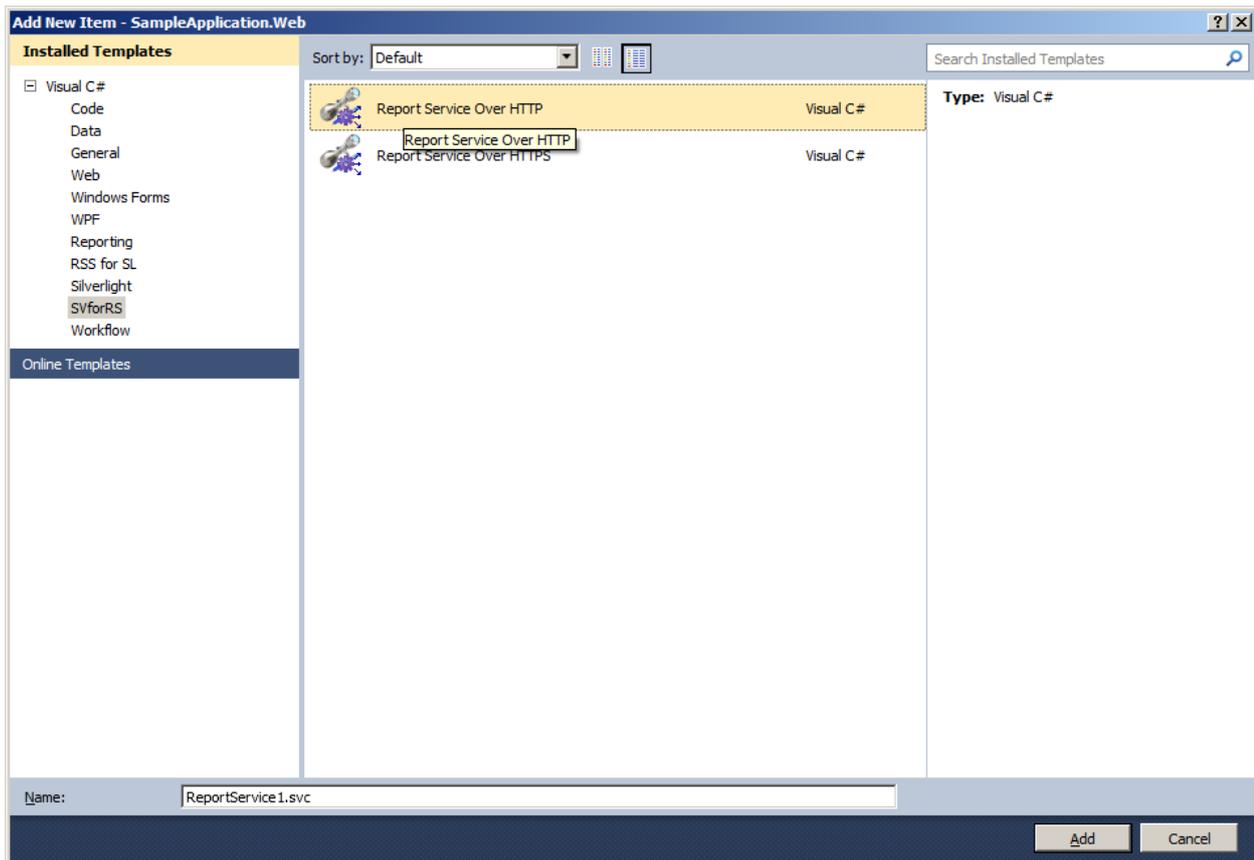
Step 3

To configure server side you can use Templates.

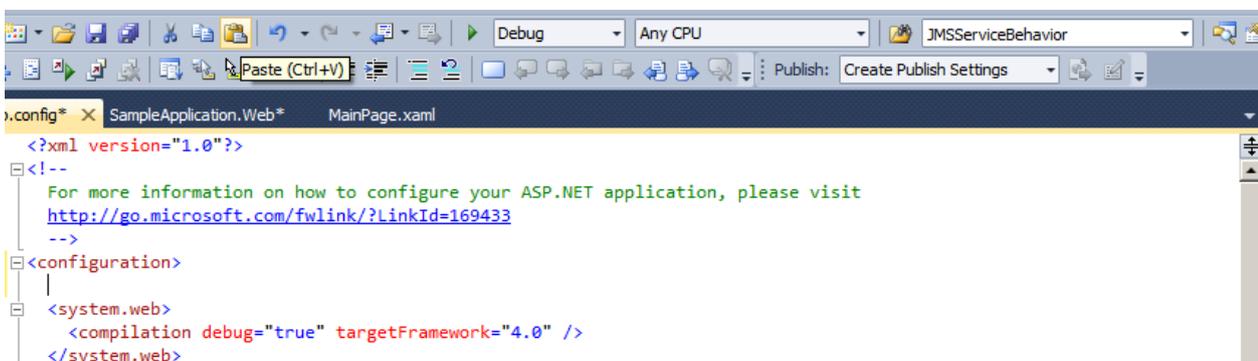
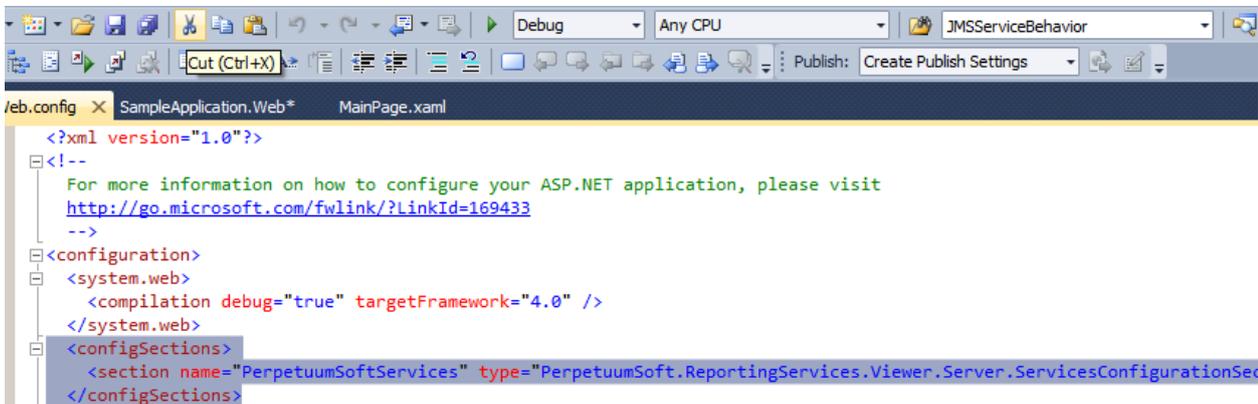
Add new item to the SampleApplication.Web project (right-click SampleApplication.Web in the "Solution Explorer" and choose Add\New Item...):



Then choose Visual C#->SVforRS->Report Service Over HTTP.



After that you should open Web.config (double-click SampleApplication.Web\web.config in the "Solution Explorer") and move "configSections" section containing "PerpetuumSoftServices" definition to the very beginning of the "configuration" section:



After that you will get something like this

```

Web.config* x SampleApplication.Web* MainPage.xaml
<?xml version="1.0"?>
<!--
  For more information on how to configure your ASP.NET application, please visit
  http://go.microsoft.com/fwlink/?LinkId=169433
-->
<configuration>
  <configSections>
    <section name="PerpetuumSoftServices" type="PerpetuumSoft.ReportingServices.Viewer.Server.ServicesConfigurationSec
  </configSections>
  <system.web>
    <compilation debug="true" targetFramework="4.0" />
  </system.web>

```

Step4

Now we need to configure automatically created ReportService1 WCF-service. Make modifications to the web.config file (double-click SampleApplication.Web\web.config in the "Solution Explorer").

In order to configure your WCF service you just need to set up Url attribute of the MsReportingServer section. Also you should provide your Credentials for accessing Reporting Services including Domain, UserName and Password.

```

<PerpetuumSoftServices>
  <!--
    Set LogMode value="On" to log errors.
    If hosting provider does not allow to write into files set LogMode value
    to "Off".
  -->
  <LogMode value="On"/>
  <Service Type="SampleApplication.Server.ReportService,
SampleApplication.Server">
    <MsReportingServer
Url="http://localhost/ReportServer/ReportExecution2005.asmx">
      <Credentials Domain="myDomain.com" UserName="user1" Password="myPassword"/>
    </MsReportingServer>
  </Service>
</PerpetuumSoftServices>

```

LogMode element is designated to setup exception logging. Log.txt file is saved under the {Web Site Root}\Log directory on your server. Please note that some Hosting Providers don't allow your application to access the file system. Try to change the attribute value to "Off" if your SampleApplication cannot be executed properly.

MsReportingServer attributes and elements description:

Url attribute specifies path to your Microsoft SQL Reporting Services 2008/2008 R2 web service.

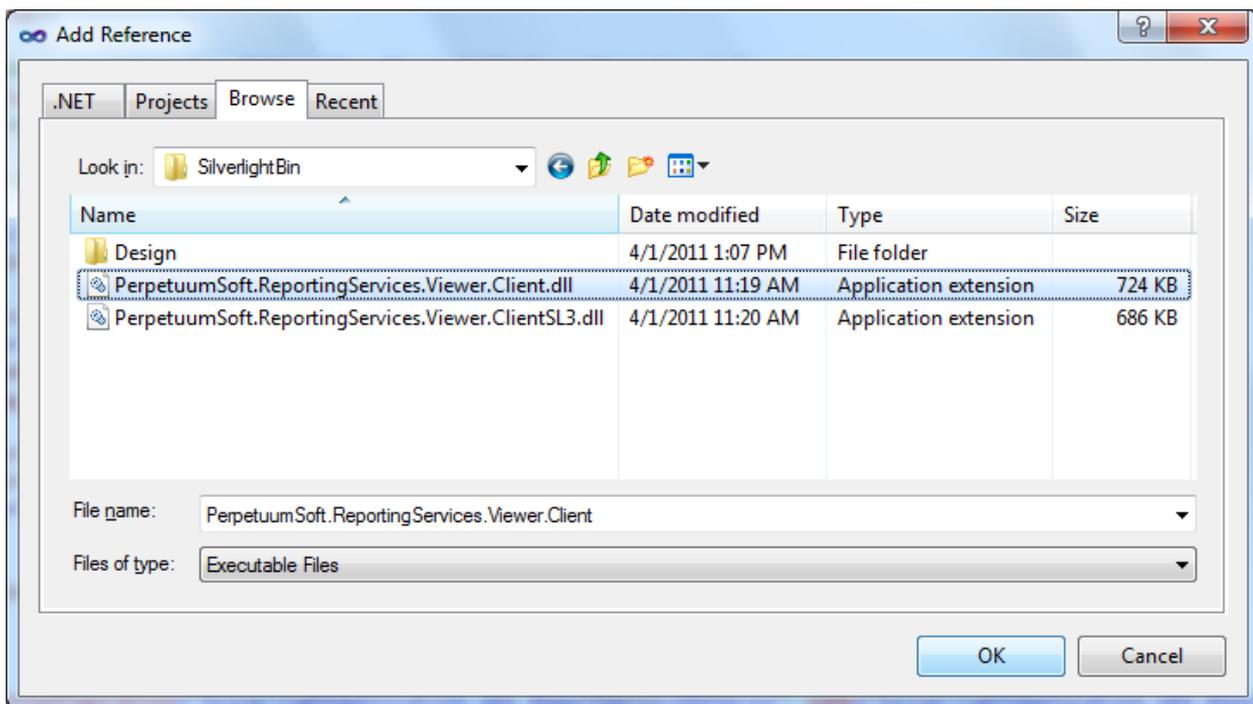
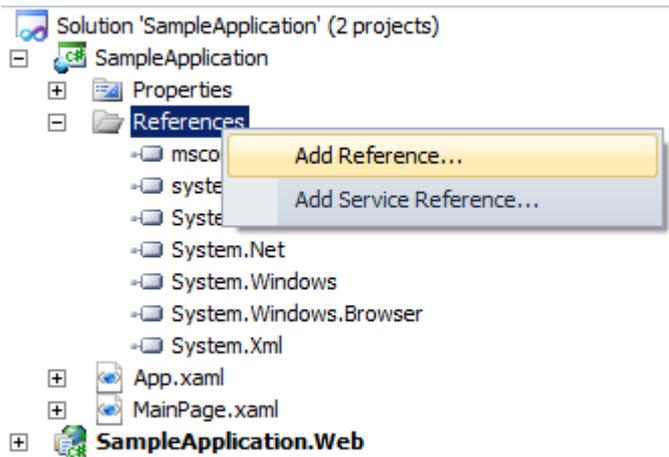
Credentials element defines credentials to access to the Reporting Service web service.

If you omit the Credentials element or leave the attributes blank, the server will use Default Network Credentials of Web Site.

Step5

NOTE: We suppose you have Microsoft SQL Reporting Services installed and configured.

It's necessary to add the report viewer component to the Silverlight application for report displaying. Therefore you should add a reference to the PerpetuumSoft.ReportingServices.Viewer.Client assembly, containing ReportViewer (right-click "Sample Application" in "Solution Explorer" and choose item "Add Reference" in context menu). The assembly is located in the Bin folder of the Silverlight Viewer for Reporting Services installation folder.



Besides this you should add references on the following libraries:

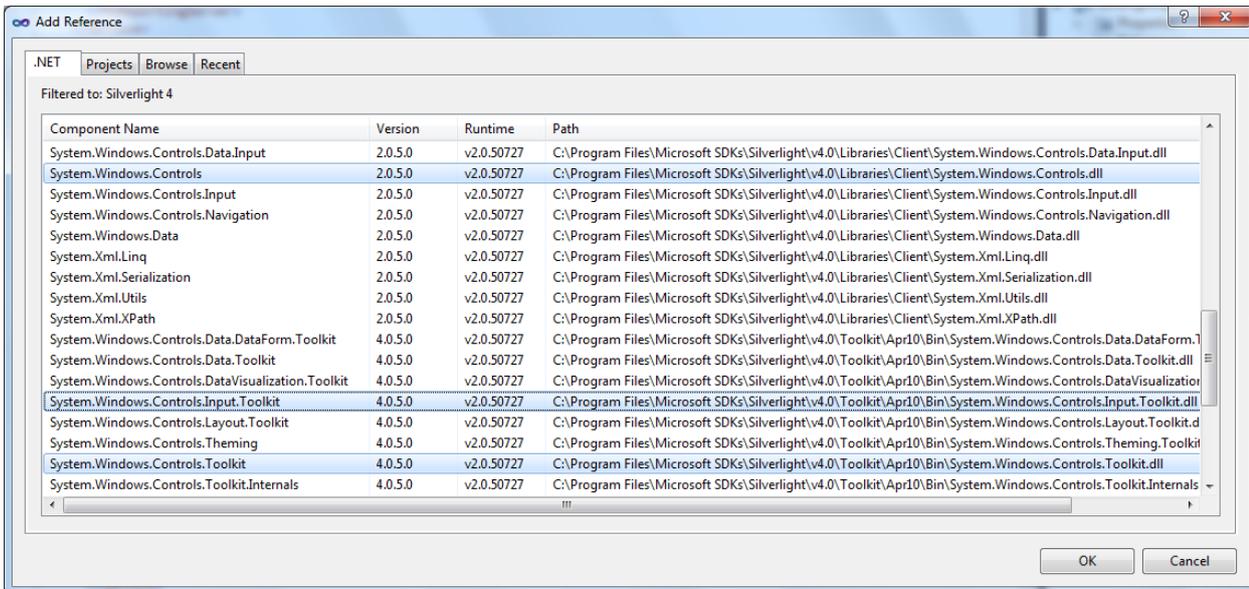
System.Windows.Controls;

System.Windows.Controls.Toolkit.

System.Windows.Controls.Input.Toolkit.dll.

They are used for document tree and bookmarks tabs visualization.

In order to do this right-click "Sample Application" in "Solution Explorer" and choose item "Add Reference" in context menu.



Open MainPage.xaml in the markup designer and add xml namespace for the PerpetuumSoft.ReportingServices.Viewer.Client assembly.

Then add ReportViewer.

```

Design  ↑  XAML
<UserControl x:Class="SampleApplication.MainPage"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
  xmlns:SLVforSSRS="clr-namespace:PerpetuumSoft.Reporting.Silverlight.Client;assembly=PerpetuumSoft.Reporting.Silverlight.Client"
  xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
  mc:Ignorable="d"
  d:DesignHeight="300" d:DesignWidth="400">
  <Grid x:Name="LayoutRoot" Background="White">
    <SLVforSSRS:ReportViewer x:Name="reportViewer"/>
  </Grid>
</UserControl>
  
```

Open the MainPage.xaml.cs source code and add the following code.

```

public partial class MainPage : UserControl
{
    public MainPage()
    {
        InitializeComponent();
        reportViewer.Loaded += new RoutedEventHandler(reportViewer_Loaded);
    }

    void reportViewer_Loaded(object sender, RoutedEventArgs e)
    {
        reportViewer.ReportName = "/AdventureWorks 2008 Sample Reports/Product
Line Sales 2008";
        if (HtmlPage.IsEnabled == false)
        {
            reportViewer.ServiceUrl =
"http://localhost:5555/ReportService1.svc";
        }
        else
        {
            SilverlightInitParamsHelper initParamsHelper =
                new SilverlightInitParamsHelper(
                    HtmlPage.Plugin.GetProperty("initParams").ToString());
            reportViewer.ServiceUrl = initParamsHelper.ServiceUrl;
        }
        reportViewer.ApplyTemplate();
    }
}
  
```

```

    reportViewer.RenderDocument();
  }
}

```

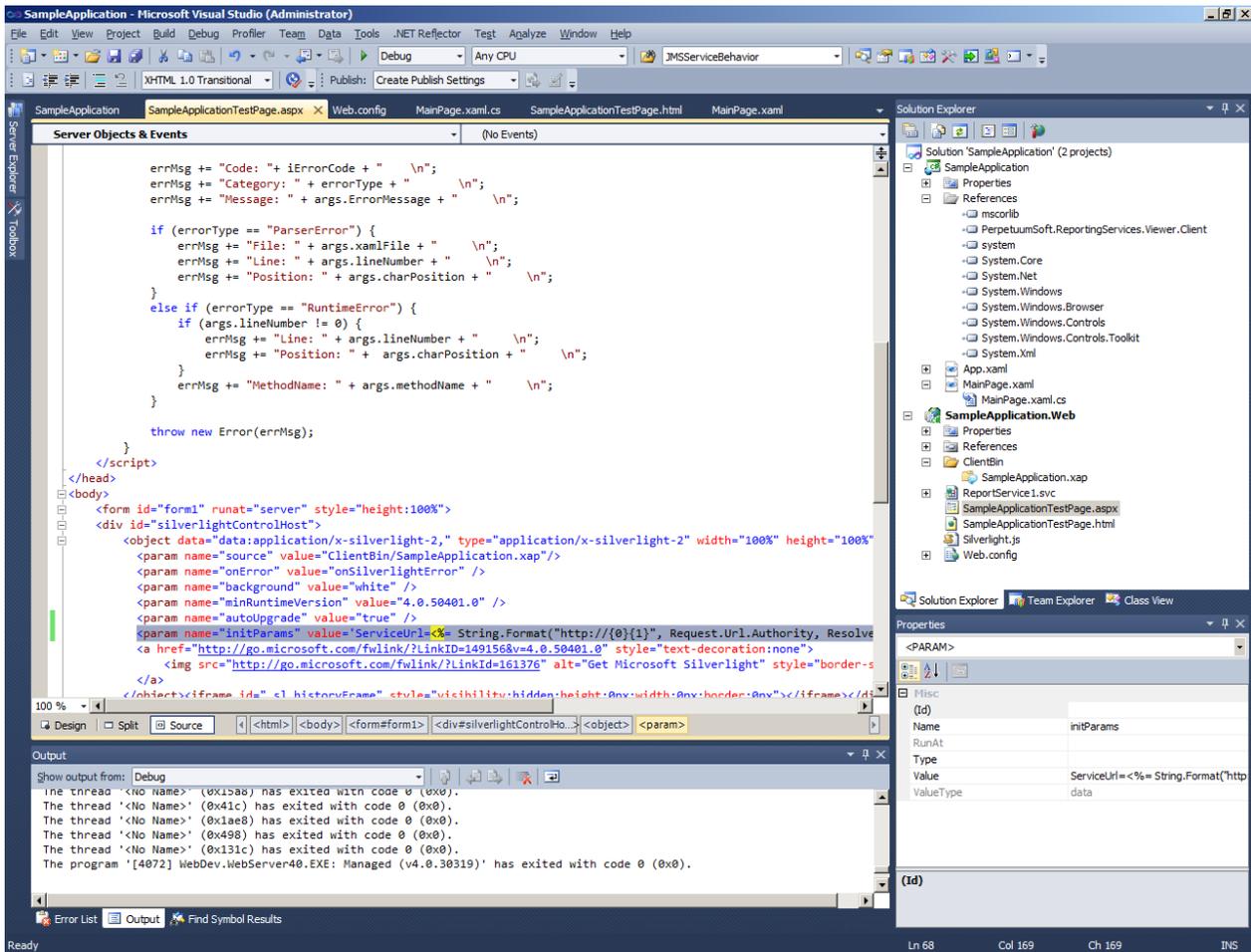
RenderDocument method invocation leads to the rendering of the current report on the server and its displaying in the Report Viewer.

In order to use initParams you should configure it on the server side.

```

<param name="initParams" value='ServiceUrl=<%= String.Format("http://{0}{1}",
Request.Url.Authority, ResolveUrl("~/ReportService1.svc")) %>;DebugMode=Full' />

```



Please pay attention to <param name="InitParams" ...> element. This element determines parameters for ReportViewer.

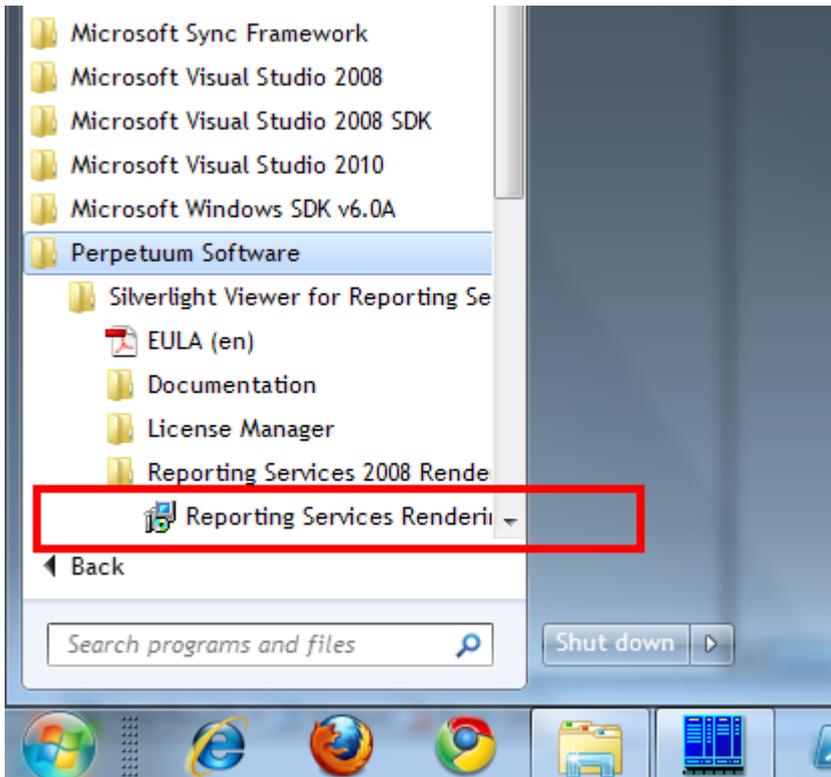
The ServiceUrl attribute specifies Url to ReportService1, the DebugMode attribute specifies the exception display mode. There are two values available "Simple" and "Full". The Full mode can be useful if any errors occur during execution.

Step 6

Install PerpetuumSoft Xaml Rendering extension.

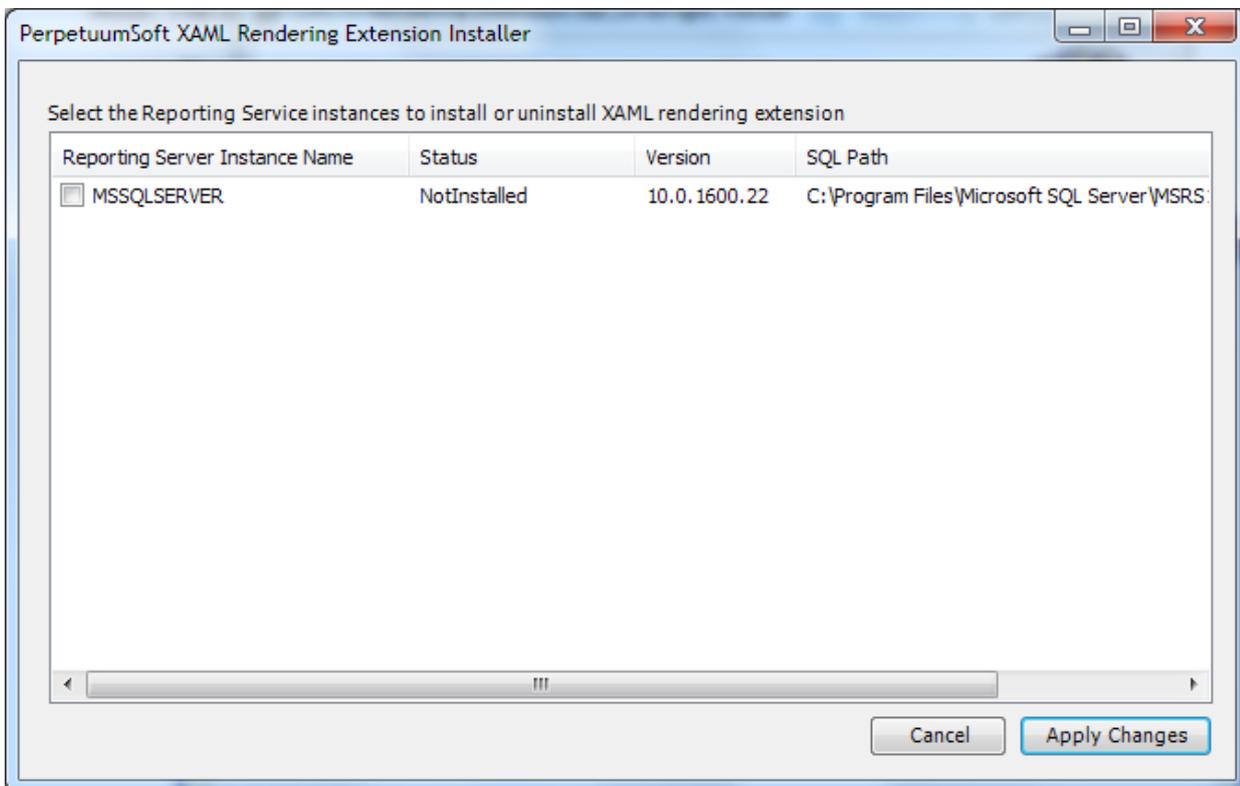
NOTE: The extension must be installed ONLY to Microsoft SQL Server Reporting Services 2008/2008 R2 Developer, Enterprise or Standard editions. Make sure that your SQL Server edition supports custom rendering extensions.

Run PerpetuumSoft.XamlExtension.msi on the machine where Microsoft SQL Server Reporting Services are installed.

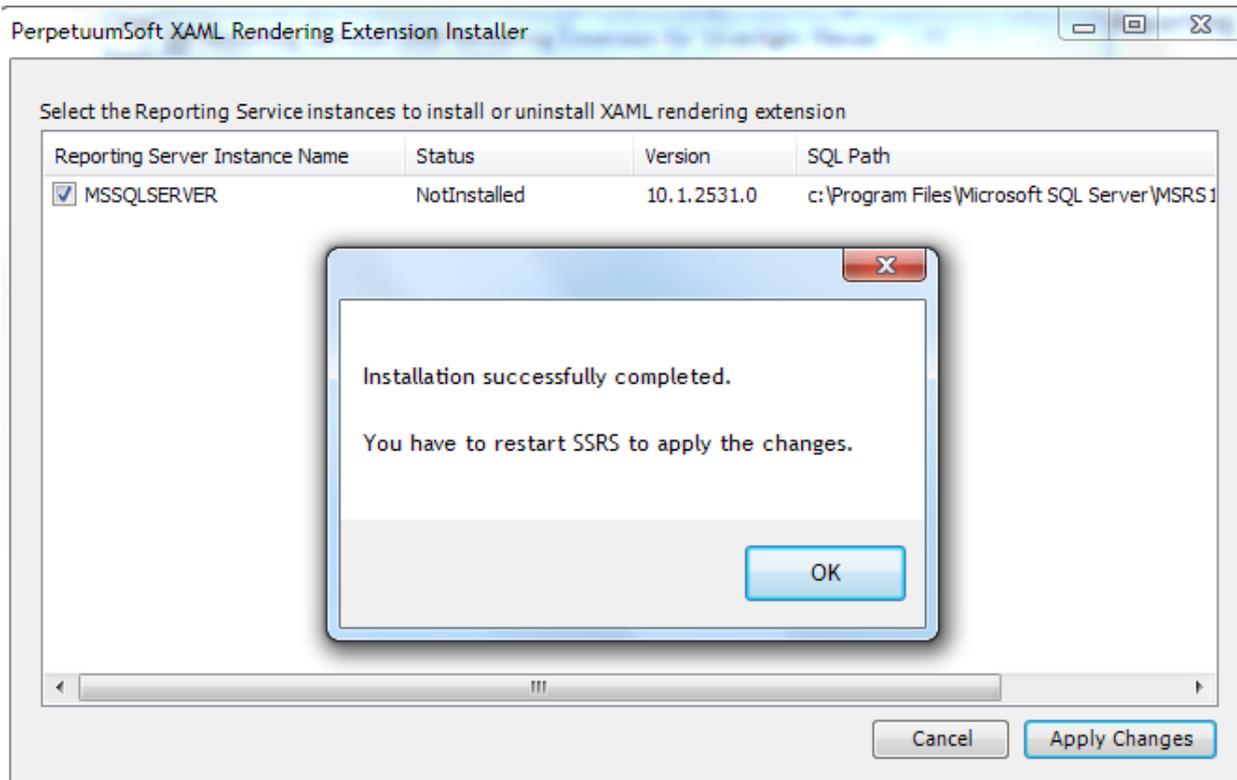


Follow the installer instructions on the screen.

The installer determines your SQL configurations and offers the list of the available Reporting Services instances.



Check/uncheck the instances to install/uninstall Rendering Extension to selected Reporting Services instances and click Apply Changes button.

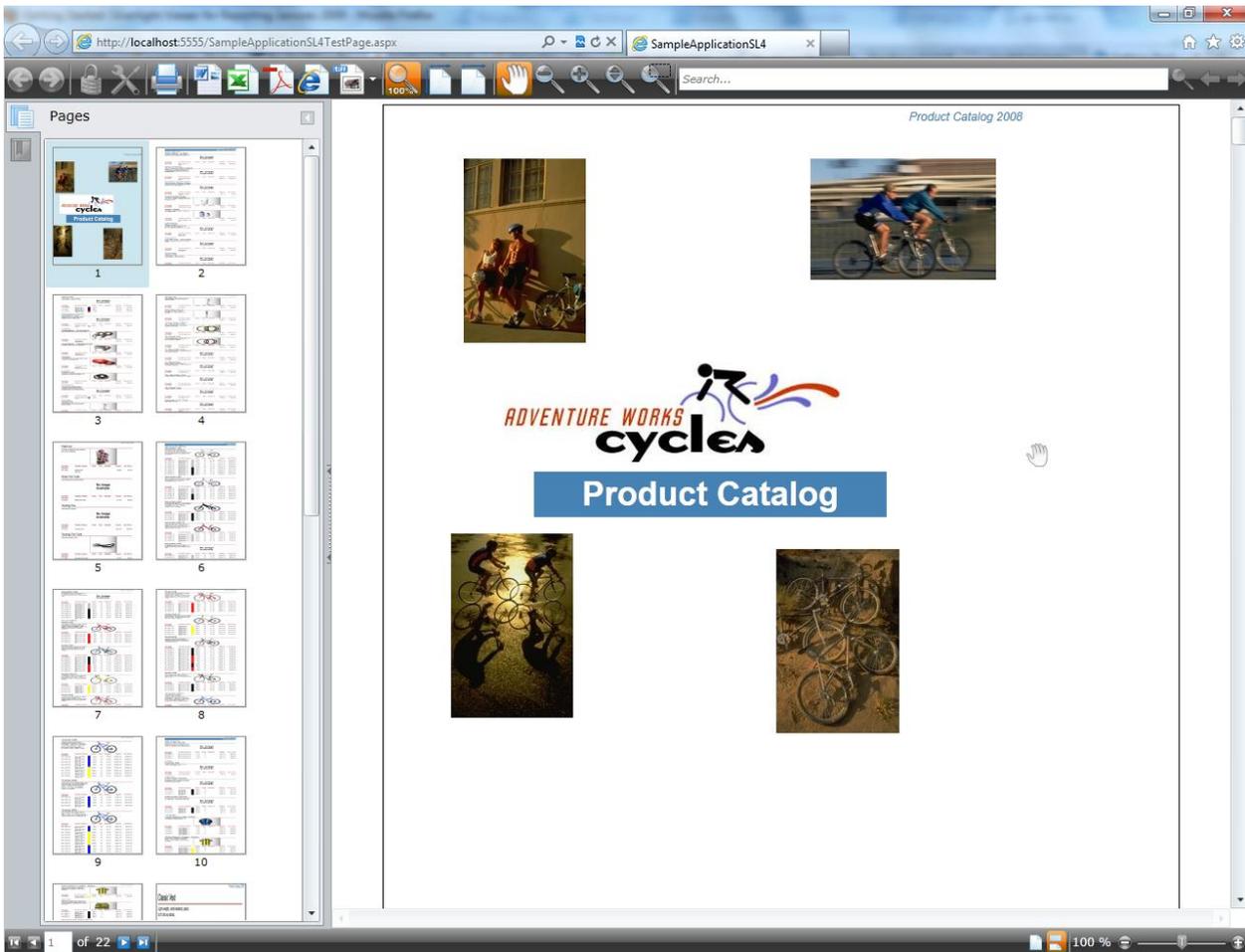


Restart the affected instances of Reporting Services.

Step7

Run Sample Application.

Press Ctrl-F5 in Visual Studio environment to run the application.



If you cannot run application or any errors occurred during execution please contact Perpetuum Software support team by [submitting a ticket](#) or by e-mail support@perpetuumsoft.com.

Please provide as much information as possible in the support request:

You can select the error message, copy it to clipboard and attach to your email.

You may also attach log.txt file (if it exists) which can be found under the {Web Site Root}\Log directory.

Conclusion

We have examined basic steps and got a simple and quite operable application. We didn't have to write thousand lines of code – we only used ready-made implementation. It will be enough in the most cases. If required behavior differs greatly from the one provided by default, you can change not only many aspects of the Silverlight Viewer for Reporting Services 2008/2008R2 work but also the appearance of the report viewer.