

SharpShooter Reports.Web Configuring Reporting Engine

Last modified on: September 5, 2011



Table of Content

Introduction.....	3
Step 1. Creation of a project.....	3
Step 2. Adding references to the assemblies.....	4
Step 3. Adding reporting engine class	6
Step 4. Adding data source	8
Step 5. Adding data to a data source	10
Step 6. Adding slot for a report.....	11
Step 7. Creation of a report with Wizard	15
Step 8. Setting up a report.....	18
Step 9. Adding navigation	20



Introduction

The purpose of this tutorial is to cover the basic steps of creation of the class which implements the server part. This server part is necessary for the use of SharpShooter Reports.Web. This step-by-step tutorial provides minimum required skills necessary for work with the component, such as:

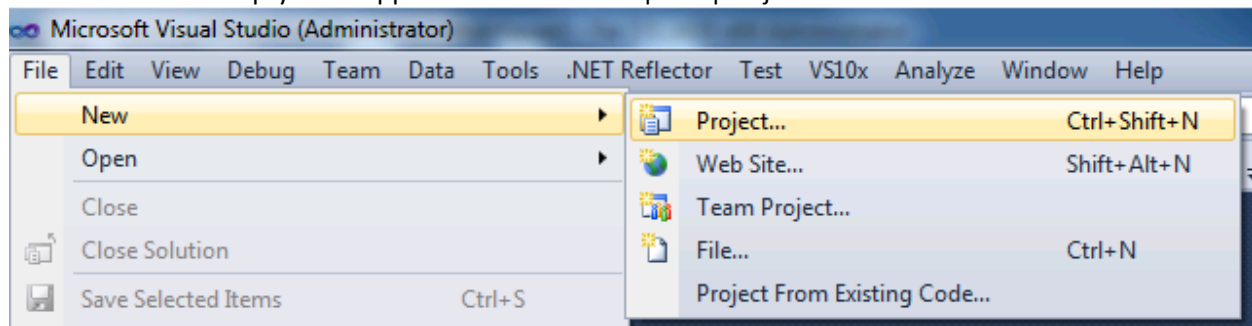
- addition of a class to a project,
- class configuration,
- addition of Dataset,
- addition of “ReportManager”,
- addition of a data source,
- a report creation.

Let’s examine the creation of a class on the basis of the “ASP.NET Empty Web Application” project. The steps described in this application are similar for such projects as:

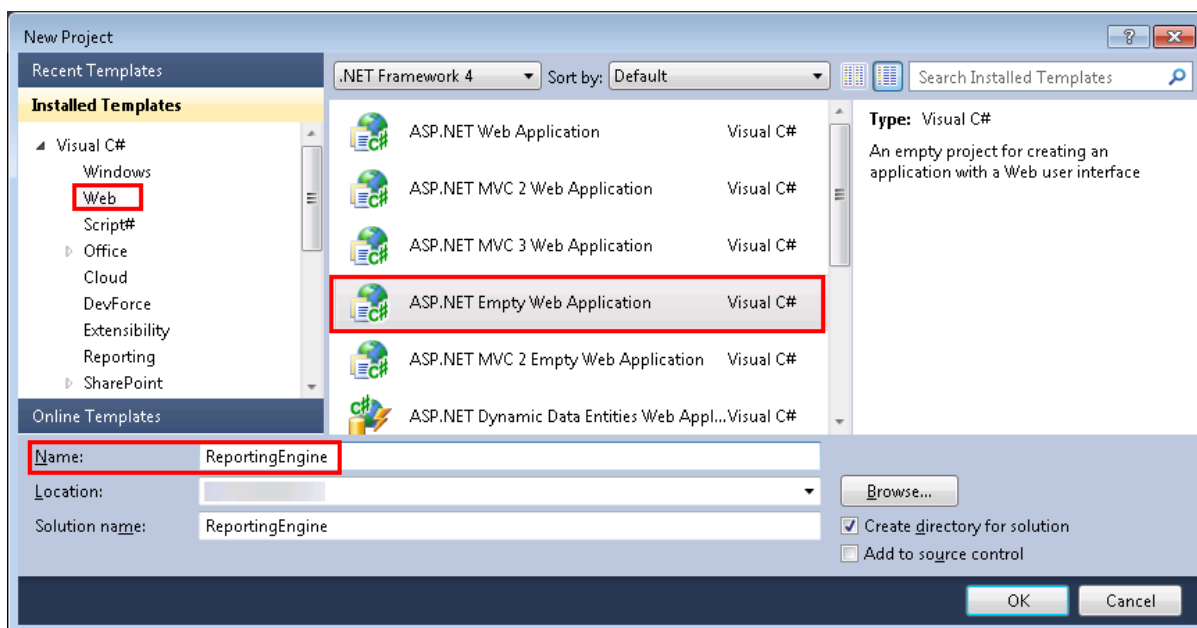
- ASP.NET Web Application
- ASP.NET MVC Web Application

Step 1. Creation of a project

Create “ASP.NET Empty Web Application”. Select File|New|Project... in the Visual Studio menu.



Select “ASP.NET Empty Web Application” in the list of projects. Enter “ReportingEngine” in the “Name” field.

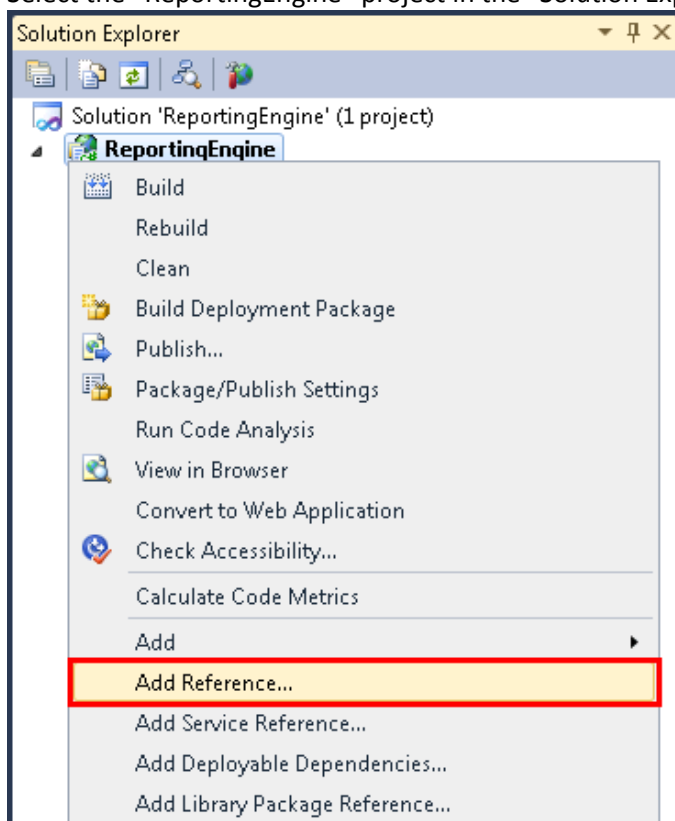


Step 2. Adding references to the assemblies

Add the references to the following assemblies to the project:

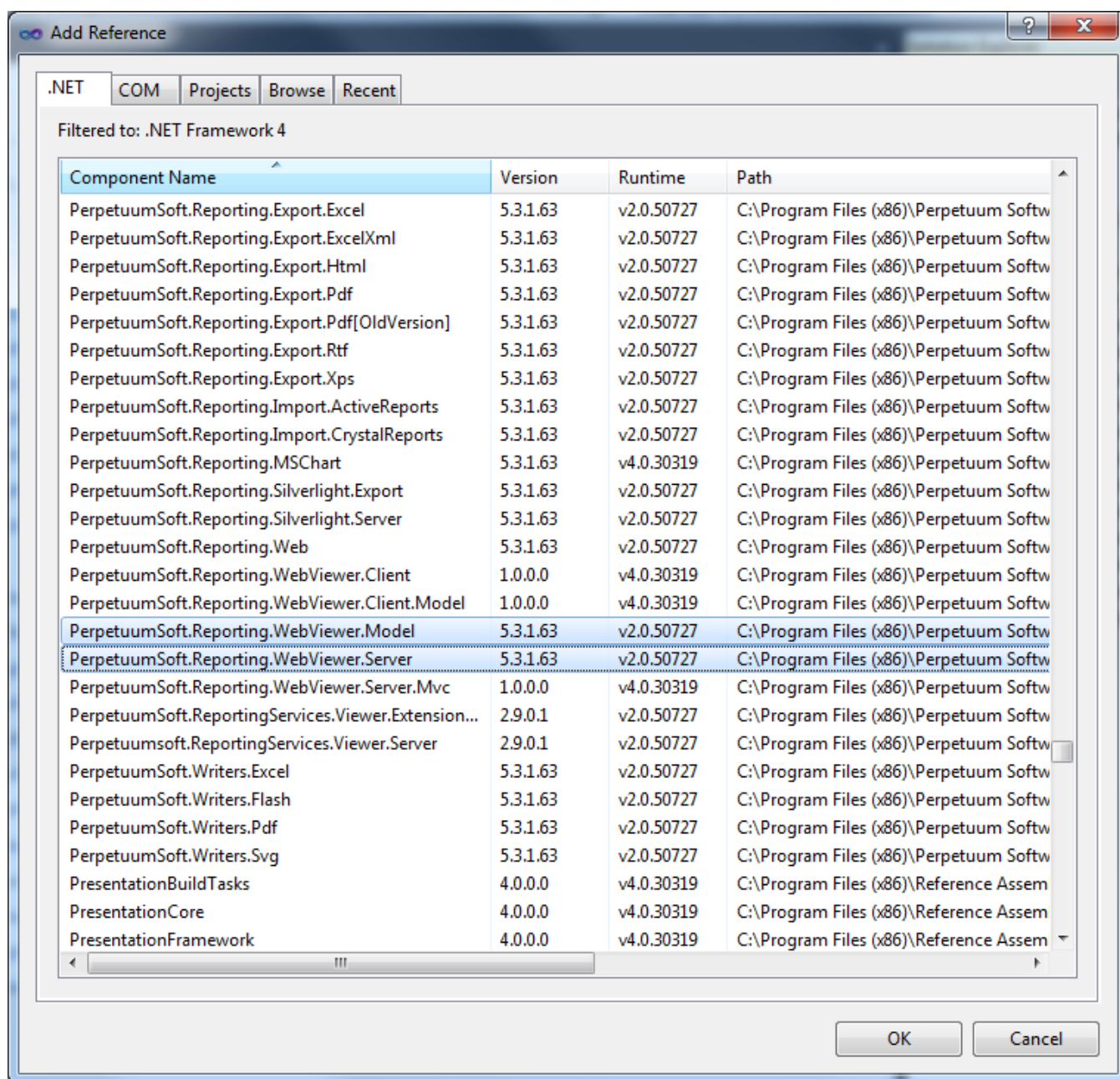
- PerpetuumSoft.Reporting.WebViewer.Model.dll
- PerpetuumSoft.Reporting.WebViewer.Server.dll
- System.ServiceModel.dll

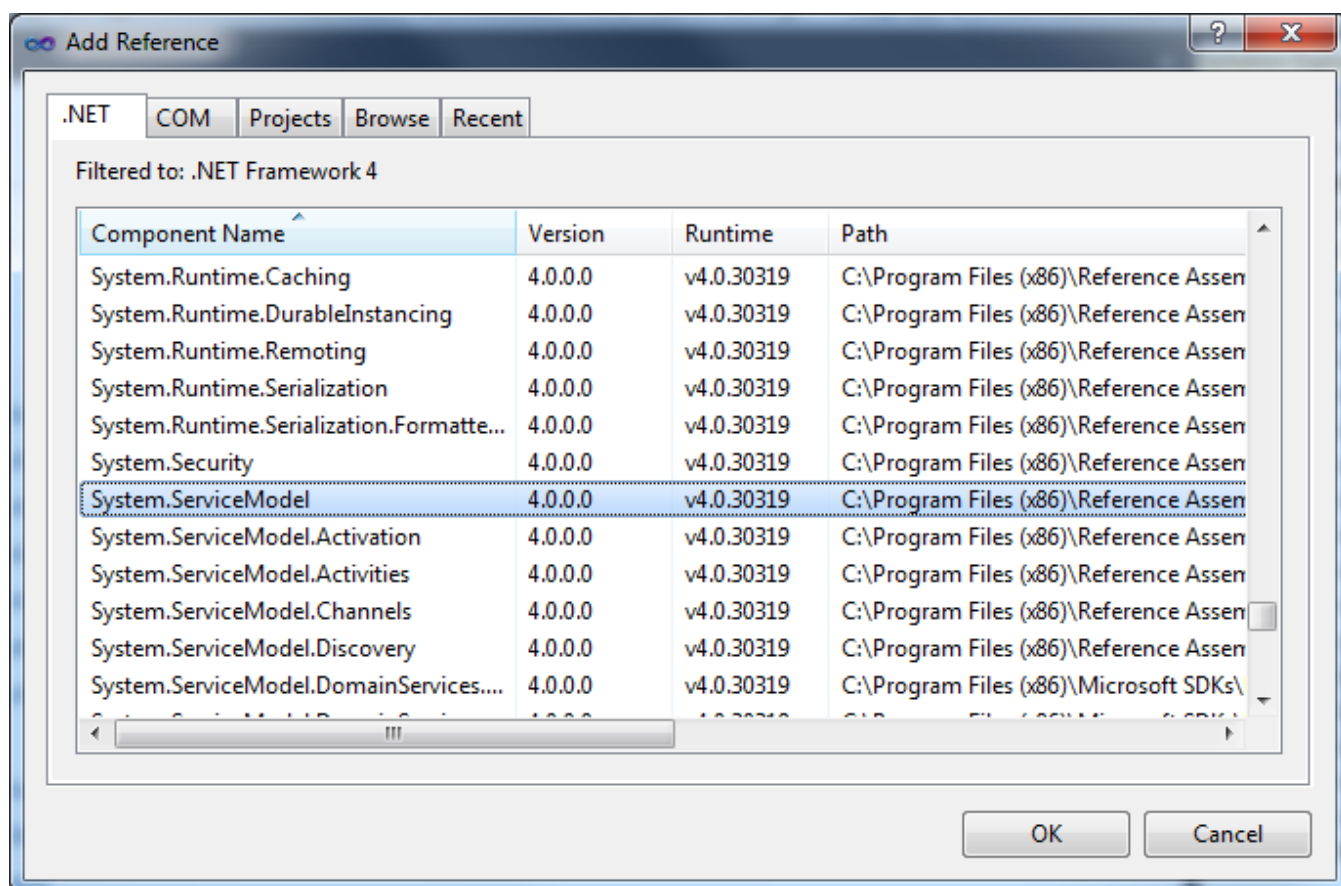
Select the "ReportingEngine" project in the "Solution Explorer". Select "Add Reference" in the context menu.





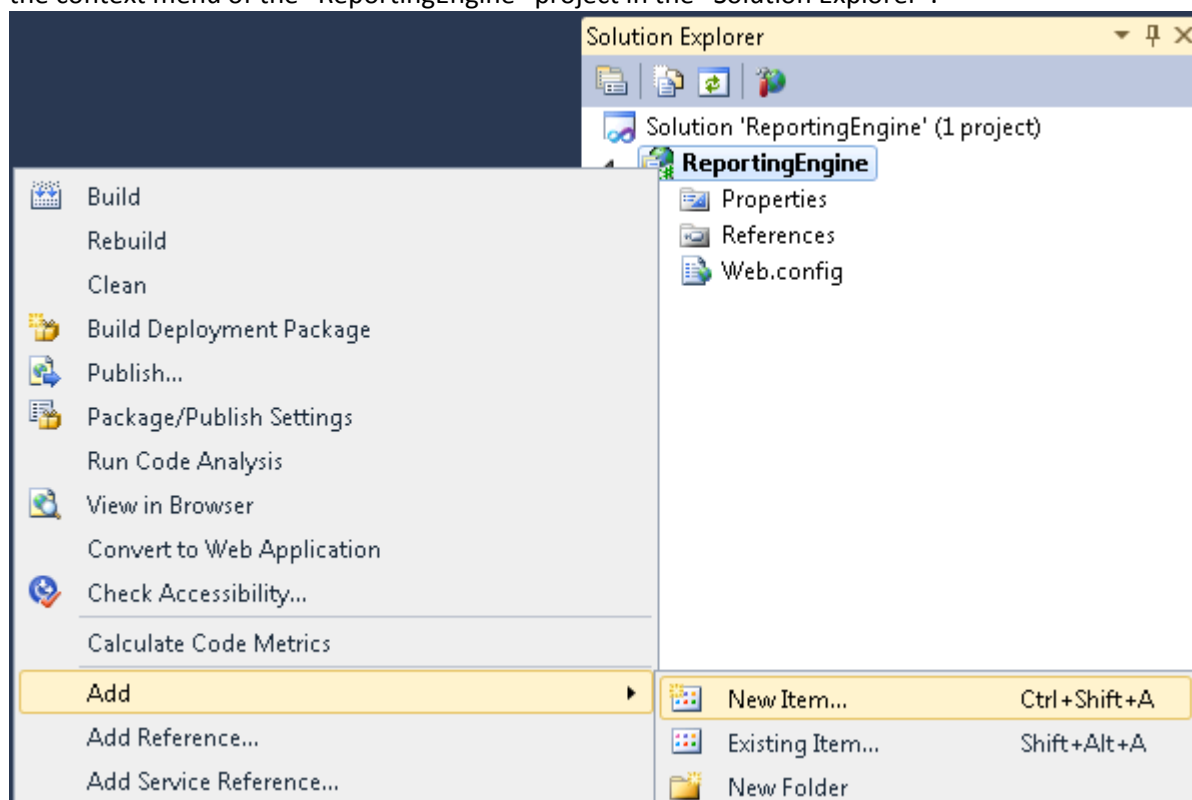
Select the PerpetuumSoft.Reporting.WebViewer.Model.dll, PerpetuumSoft.Reporting.WebViewer.Server.dll and System.ServiceModel.dll files.





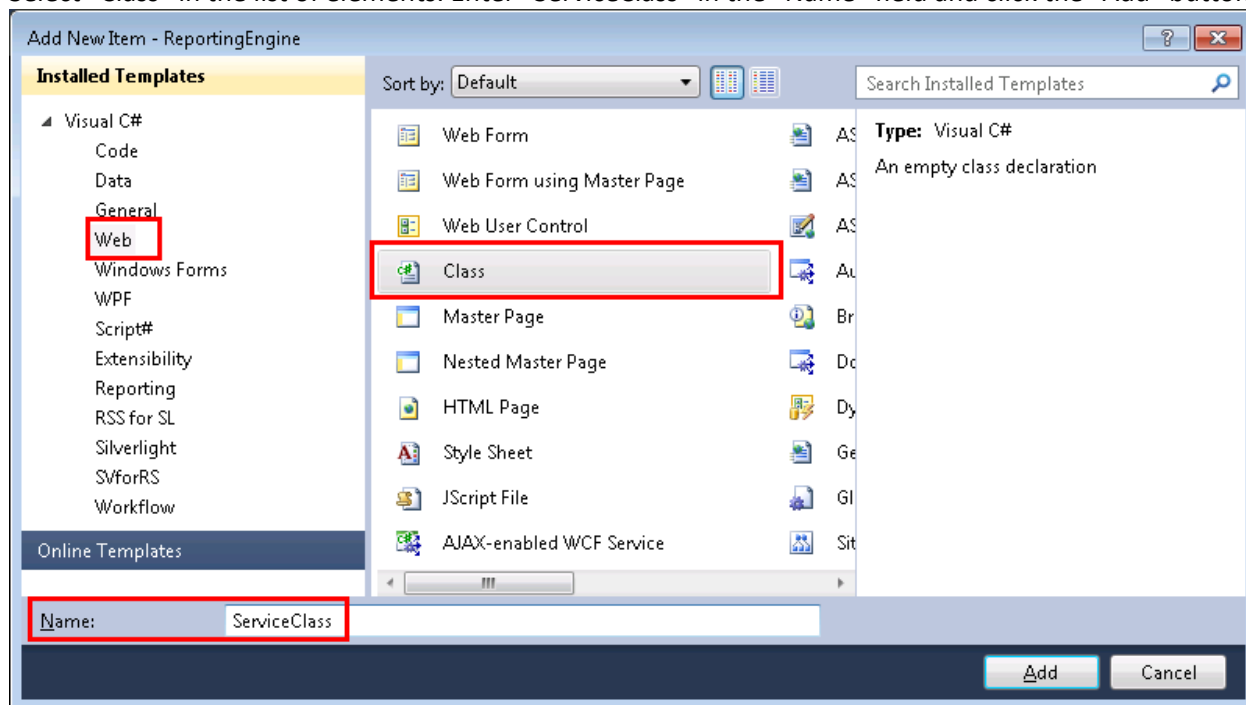
Step 3. Adding reporting engine class

Add a class to the project. The class implements the logic of the server part work. Select “Add\New Item...” in the context menu of the “ReportingEngine” project in the “Solution Explorer”.





Select “Class” in the list of elements. Enter “ServiceClass” in the “Name” field and click the “Add” button.



Open the added class. To do this, double click on the “ServiceClass.cs” element in “Solution Explorer”. Add the “System.ServiceModel.Activation”, “System.Data” and “PerpetuumSoft.Reporting.WebViewer.Server” namespaces to the “ServiceClass”. To do this, use the using directive:

```
using System.ServiceModel.Activation;  
using System.Data;  
using PerpetuumSoft.Reporting.WebViewer.Server;
```

Inherit the added class from the “ReportServiceBase” class. Thus, you get ready implementation and ability to change standard behavior. You should mark the service class with “AspNetCompatibilityRequirements” attribute. After this the service will have access to ASP.NET context during the realization of the server part. The ASP.NET context caches document’s data if other cache mechanism is not implemented.

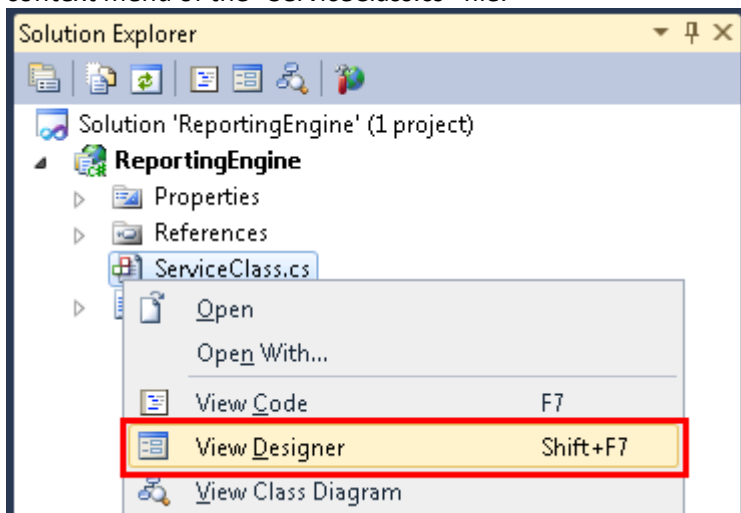
Add the “InitializeComponent” function and the constructor, which invokes this function, to the “ServiceClass” class.

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Web;  
using System.ServiceModel.Activation;  
using System.Data;  
using PerpetuumSoft.Reporting.WebViewer.Server;  
namespace ReportingEngine  
{  
    [AspNetCompatibilityRequirements(RequirementsMode = AspNetCompatibilityRequirementsMode.Required)]  
    public class ServiceClass : ReportServiceBase  
    {  
        public ServiceClass()  
        {  
            InitializeComponent();  
        }  
  
        private void InitializeComponent()  
        { }  
    }  
}
```

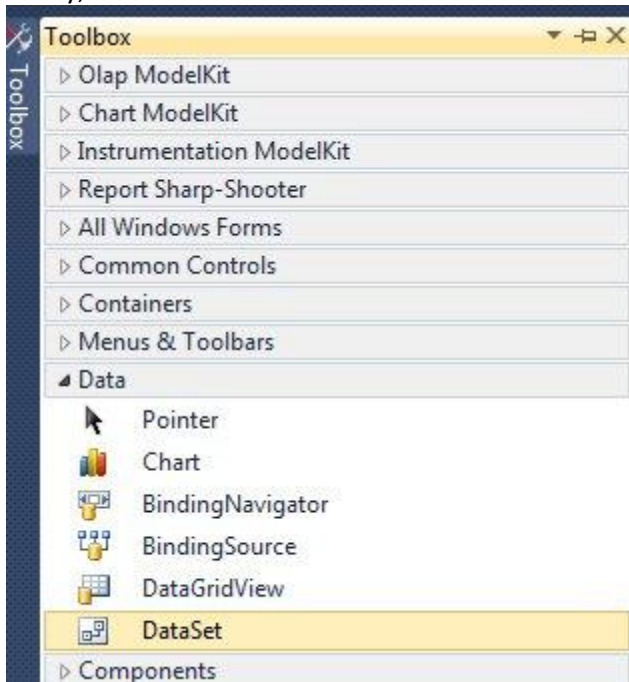


Step 4. Adding data source

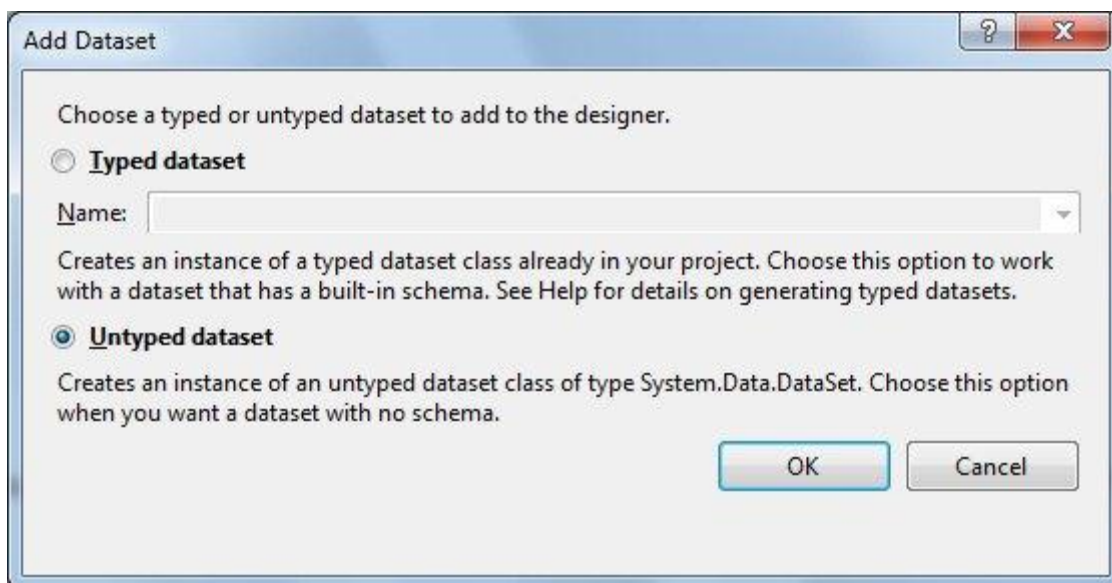
Open the designer for the “ServiceClass”. Select the “View Designer” item in the “Solution Explorer” in the context menu of the “ServiceClass.cs” file.



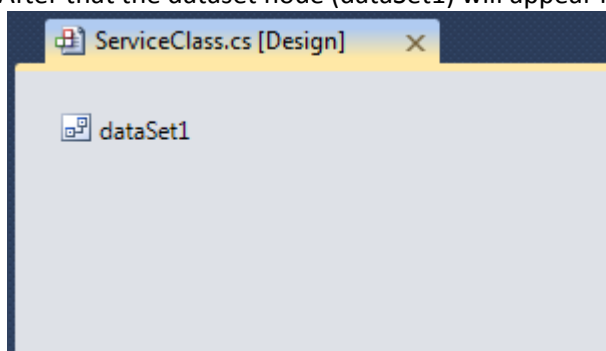
Firstly, create data source structure. Add “Dataset” from the “Toolbox” (double click on the Dataset in Toolbox).



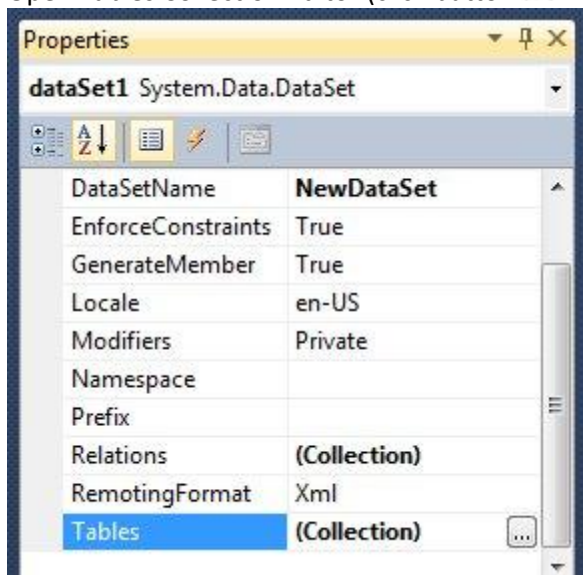
And select the Untyped dataset.



After that the dataset node (dataSet1) will appear in the designer.

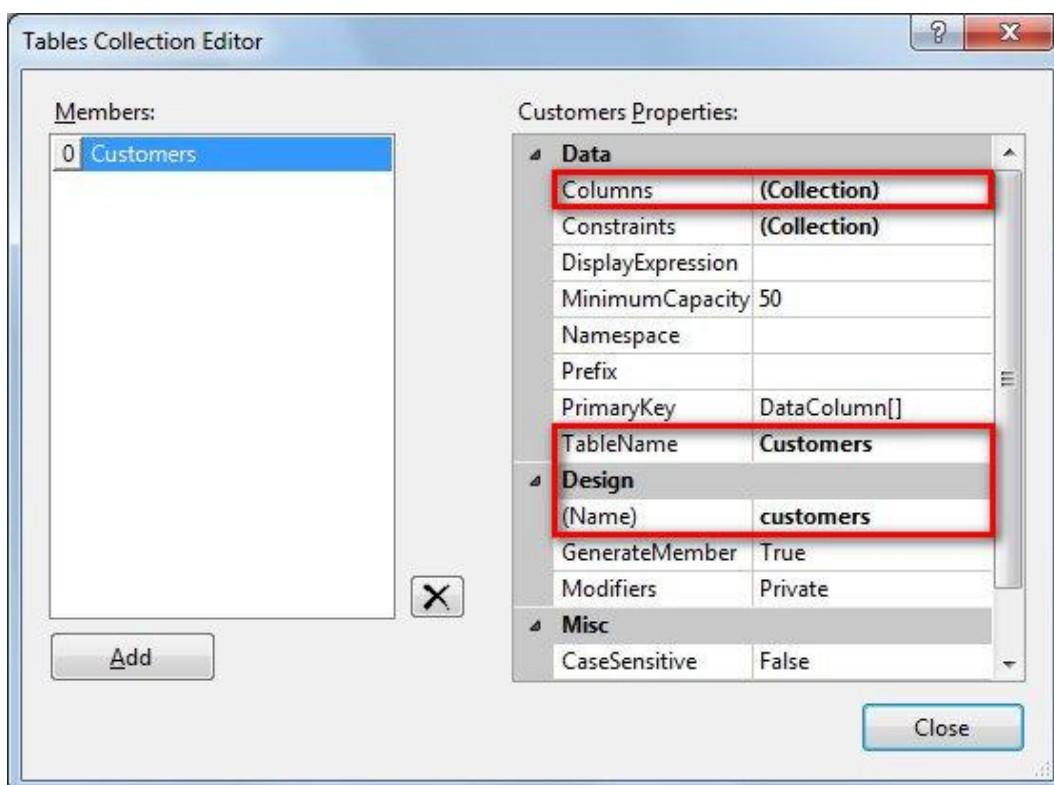


Open Tables Collection Editor (click button  in property Tables).

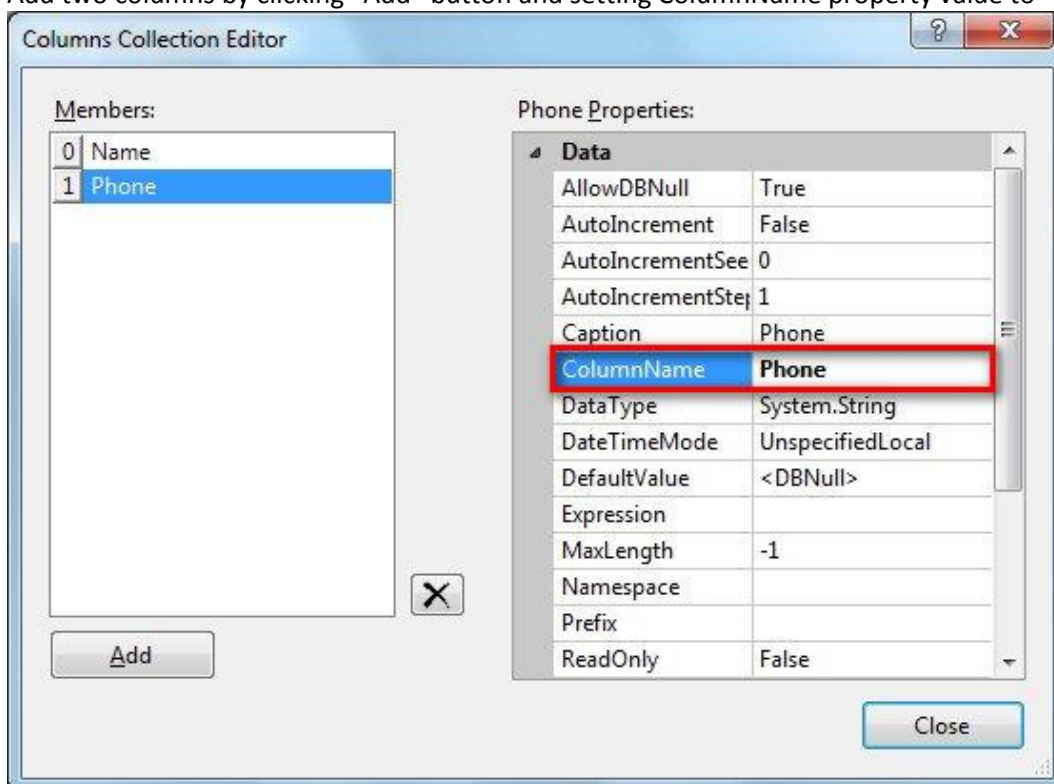


Add Customers table to the dataSet1 (click "Add" button and set TableName property value to "Customers", set Name property value to "customers").

After that open Columns Collection Editor (click button in property grid of Tables Collection Editor)

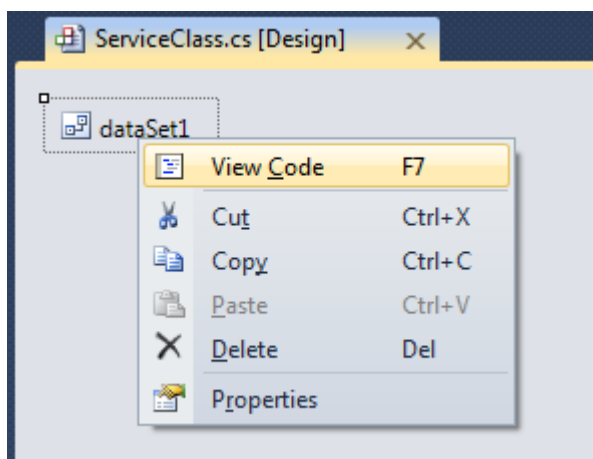


Add two columns by clicking “Add” button and setting ColumnName property value to “Name” and “Phone”.



Step 5. Adding data to a data source

Data structure is defined, and it's necessary to fill Customers Table with the data. To view source code right-click on designer area and click “View Code” item in contextual menu.



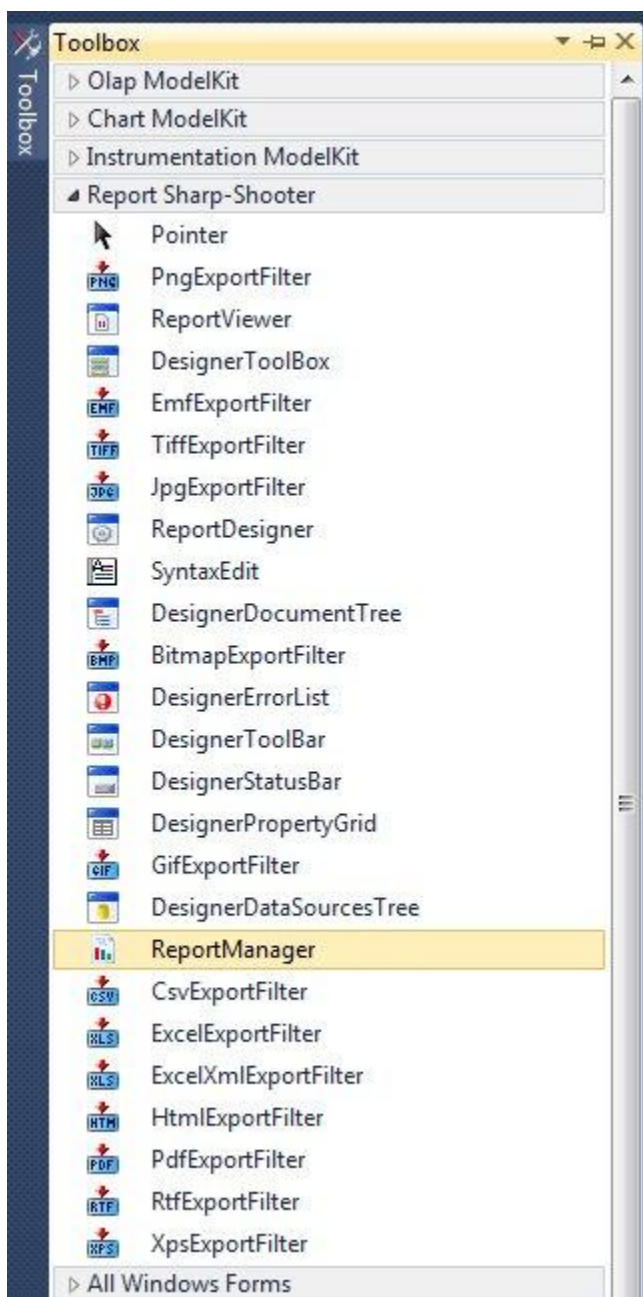
You should fill data sources with the values via overriding OnLoadData method of the ServiceClass class.

```
protected override void OnLoadData(IDictionary<string, object> parameters, string reportName, PerpetuumSoft.Reporting.Components.ReportSlot reportSlot)
```

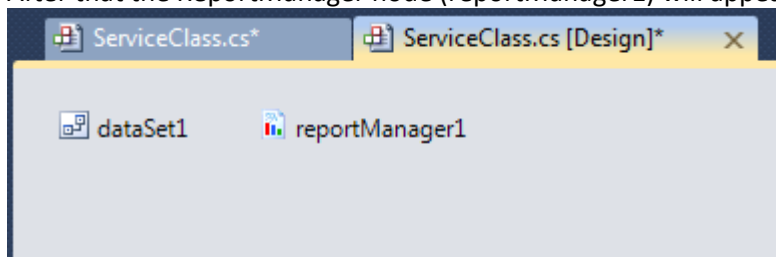
```
{  
    base.OnLoadData(parameters, reportName, reportSlot);  
    DataRow row = customers.NewRow();  
    row["Name"] = "Johnson Leslie";  
    row["Phone"] = "613-442-7654";  
    customers.Rows.Add(row);  
    row = customers.NewRow();  
    row["Name"] = "Fisher Pete";  
    row["Phone"] = "401-609-7623";  
    customers.Rows.Add(row);  
    row = customers.NewRow();  
    row["Name"] = "Brown Kelly";  
    row["Phone"] = "803-438-2771";  
    customers.Rows.Add(row);  
}
```

Step 6. Adding slot for a report

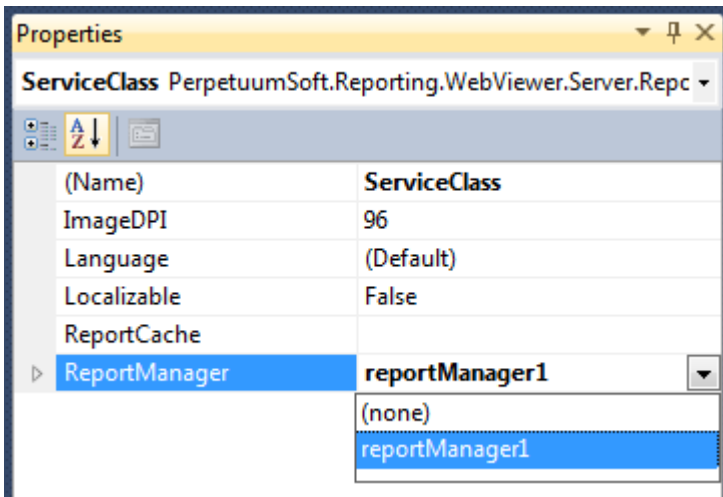
Now add the ReportManager component (double click on "ReportManager" in ToolBox); this component is responsible for report generation.



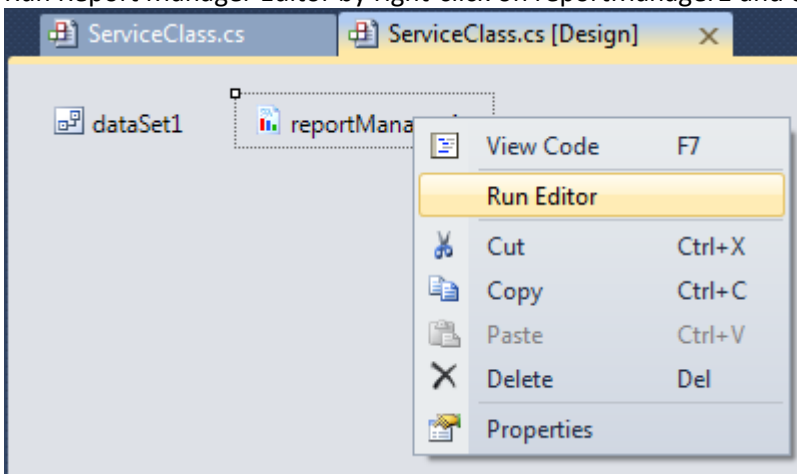
After that the ReportManager node (reportManager1) will appear in the designer.



Set the ReportManager property of the ServiceClass class. In order to do that, open ServiceClass properties in the Properties window. Select reportManager1 from the list:

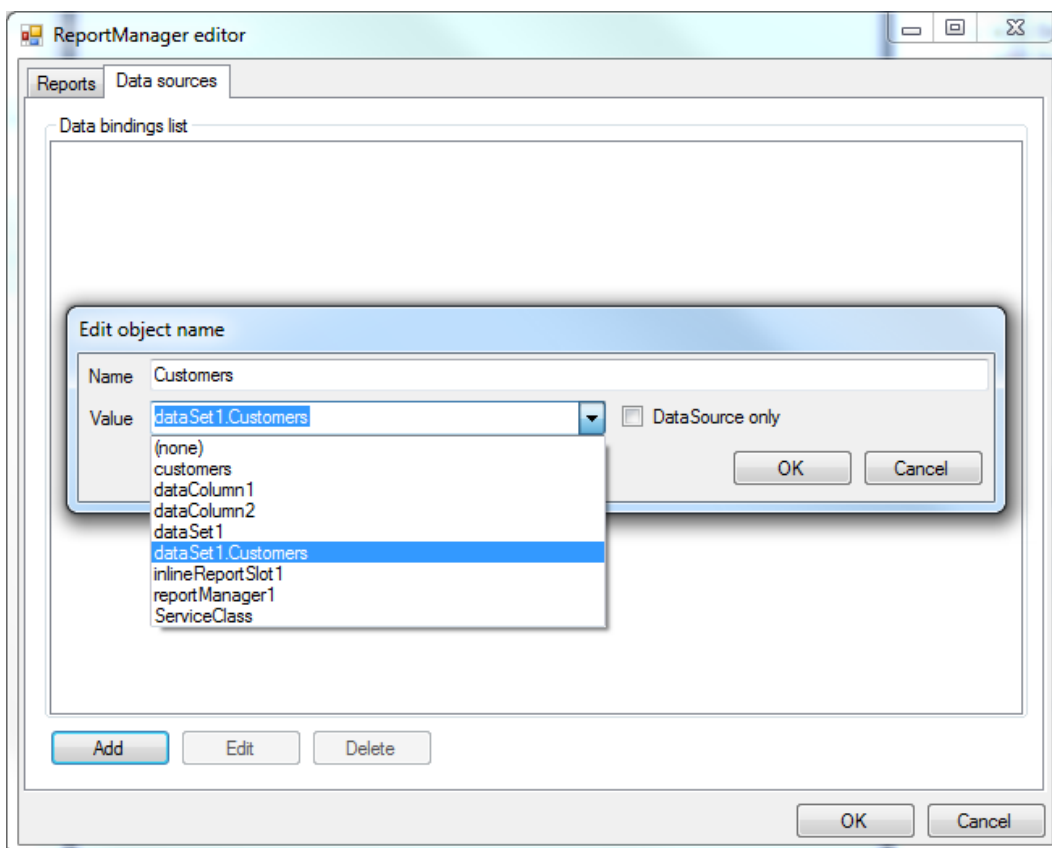


Run Report Manager Editor by right-click on reportManager1 and choosing Run Editor

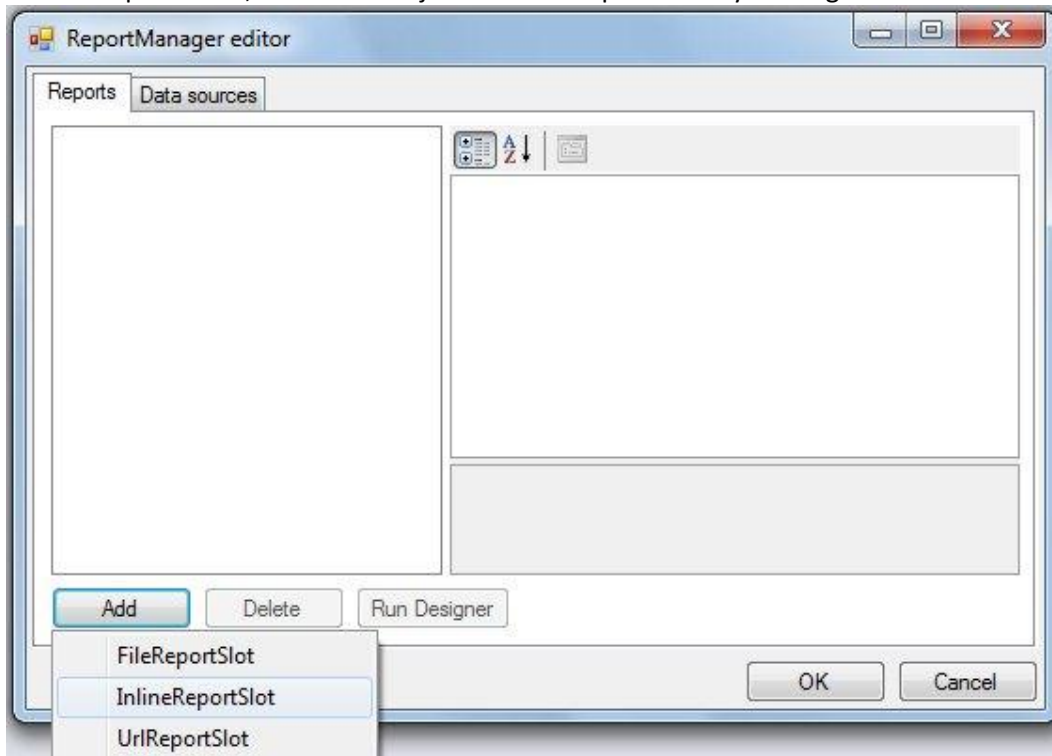


Before you start creating report template add a data source by which the report will be generated.

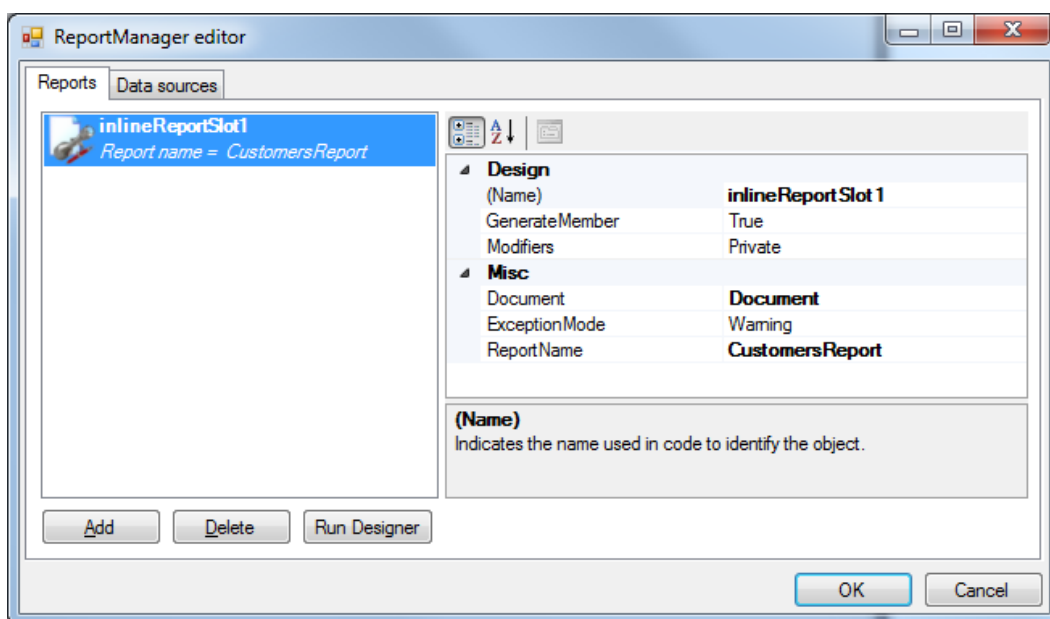
Add Customers table to the Data binding list located on the "Data sources" tab (click "Add" button, in appeared form "Edit object name" set Name value to Customers, select dataSet1.Customers in the combo box list).



In the “Reports” tab, add a new object – “InlineReportSlot” by clicking the “Add” button.

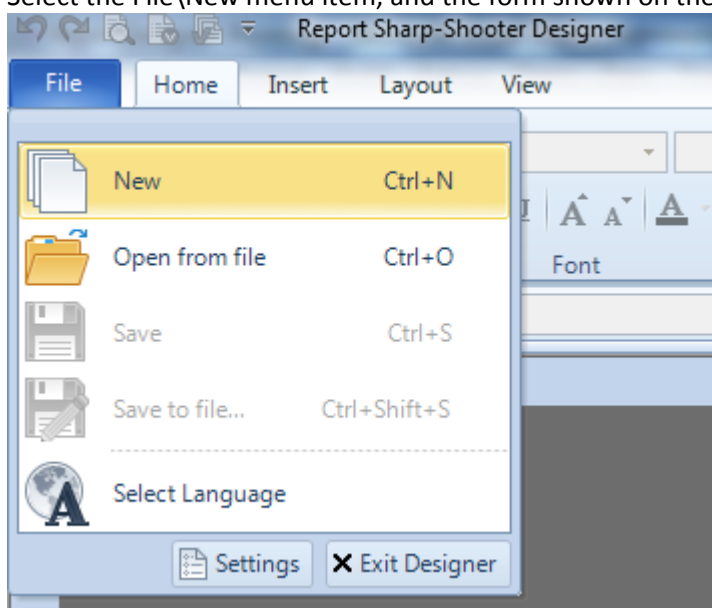


Set the ReportName property value to CustomersReport. Afterwards you will get the required document from the Report Manager exactly by that name. Then press the “Run Designer” button to launch report designer.

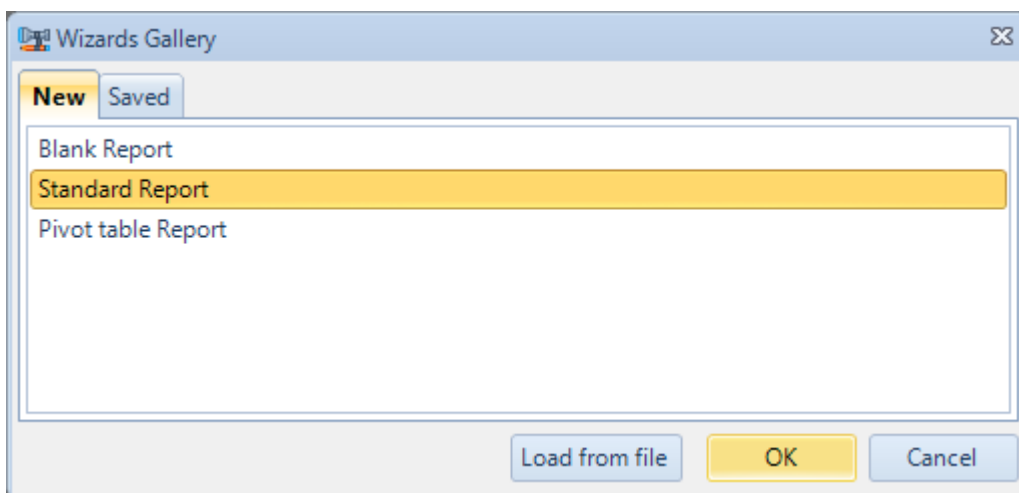


Step 7. Creation of a report with Wizard

Select the File\New menu item, and the form shown on the screen below will appear.

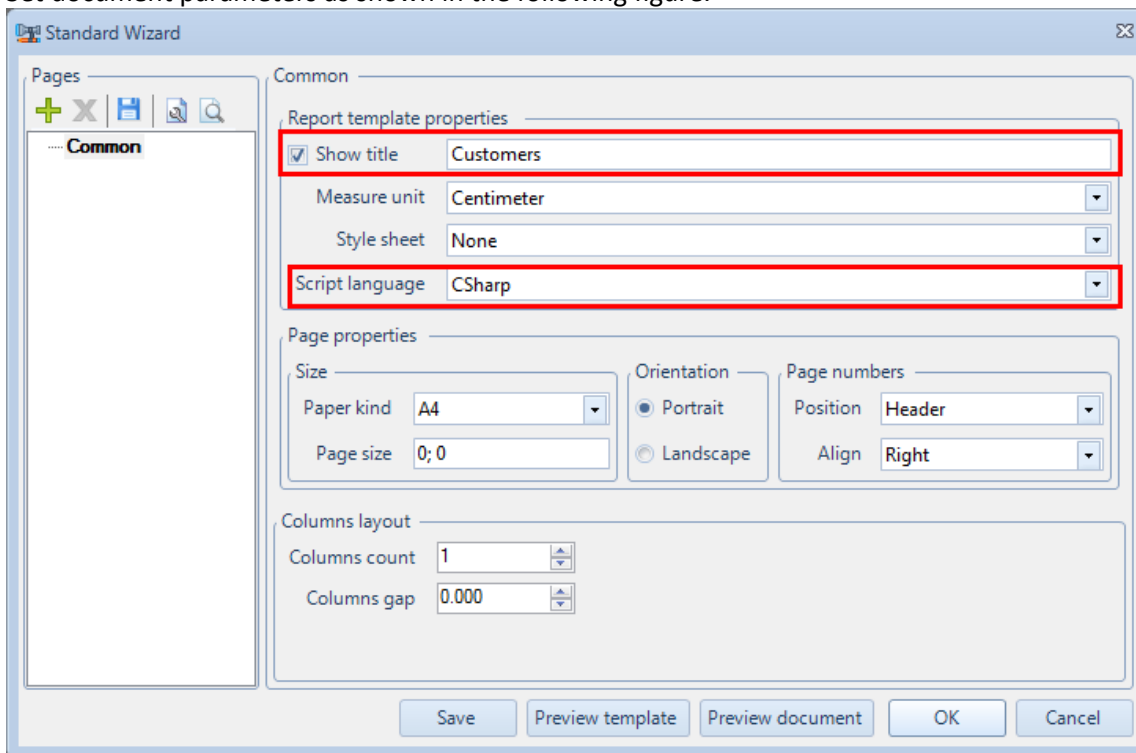


Select Standard Report in the list under the “New” tab and press OK.

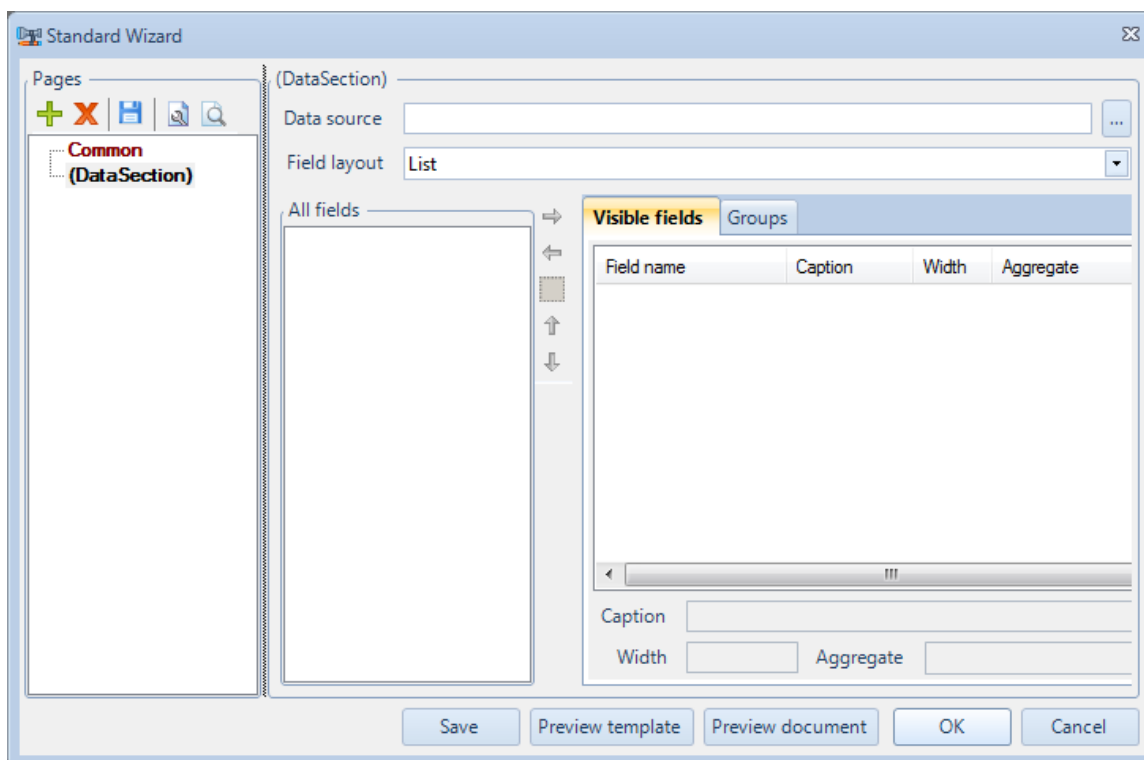



The Standard Wizard window will appear on the screen.

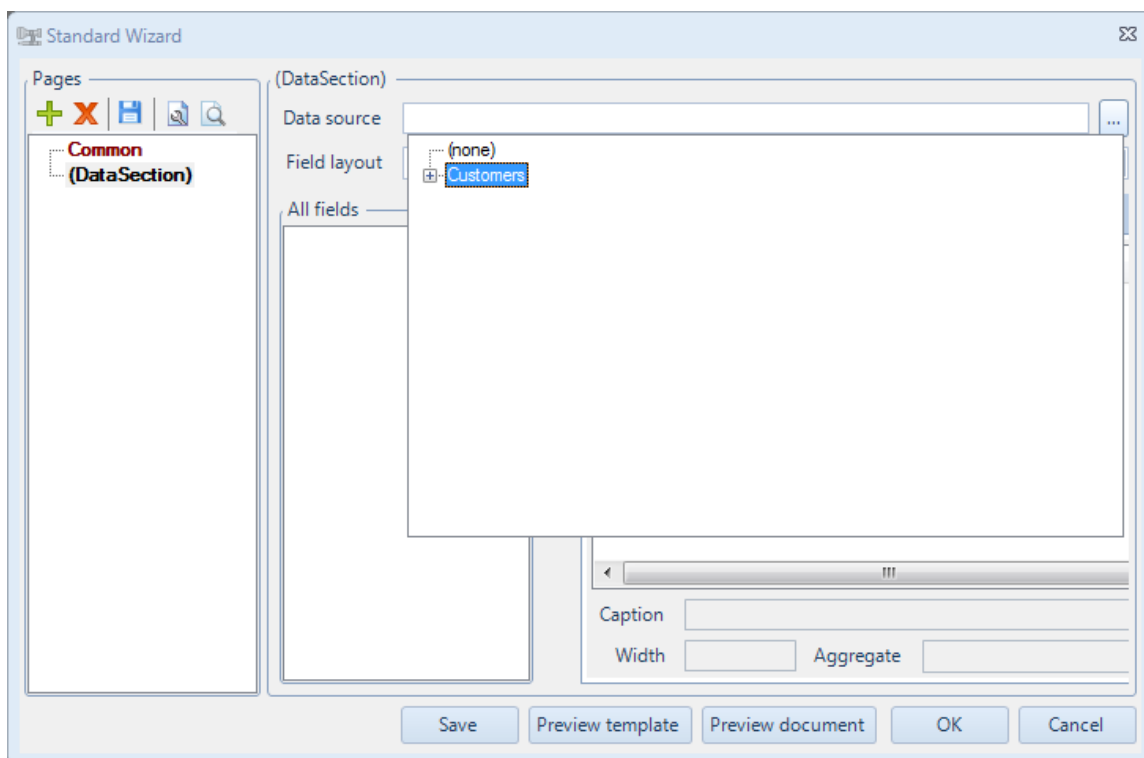
Set document parameters as shown in the following figure.



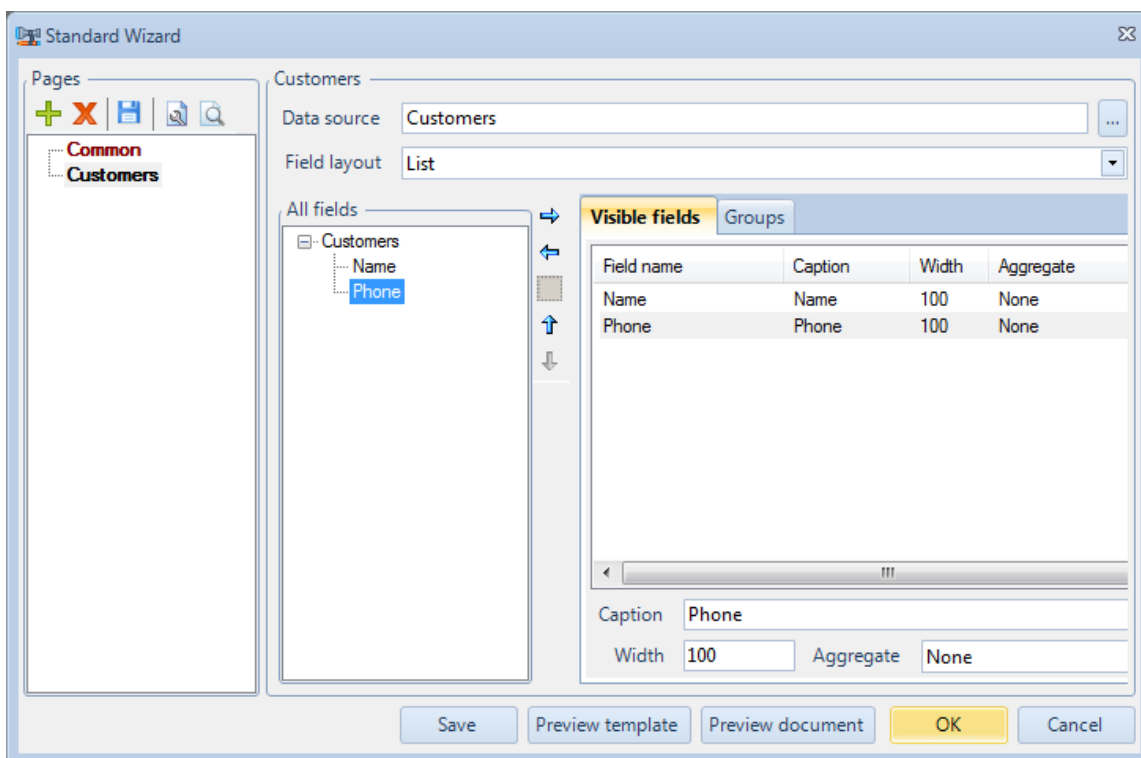
Add data source using the “Add” button (+).



Click  button, in appeared tree view select "Customers" by double click.



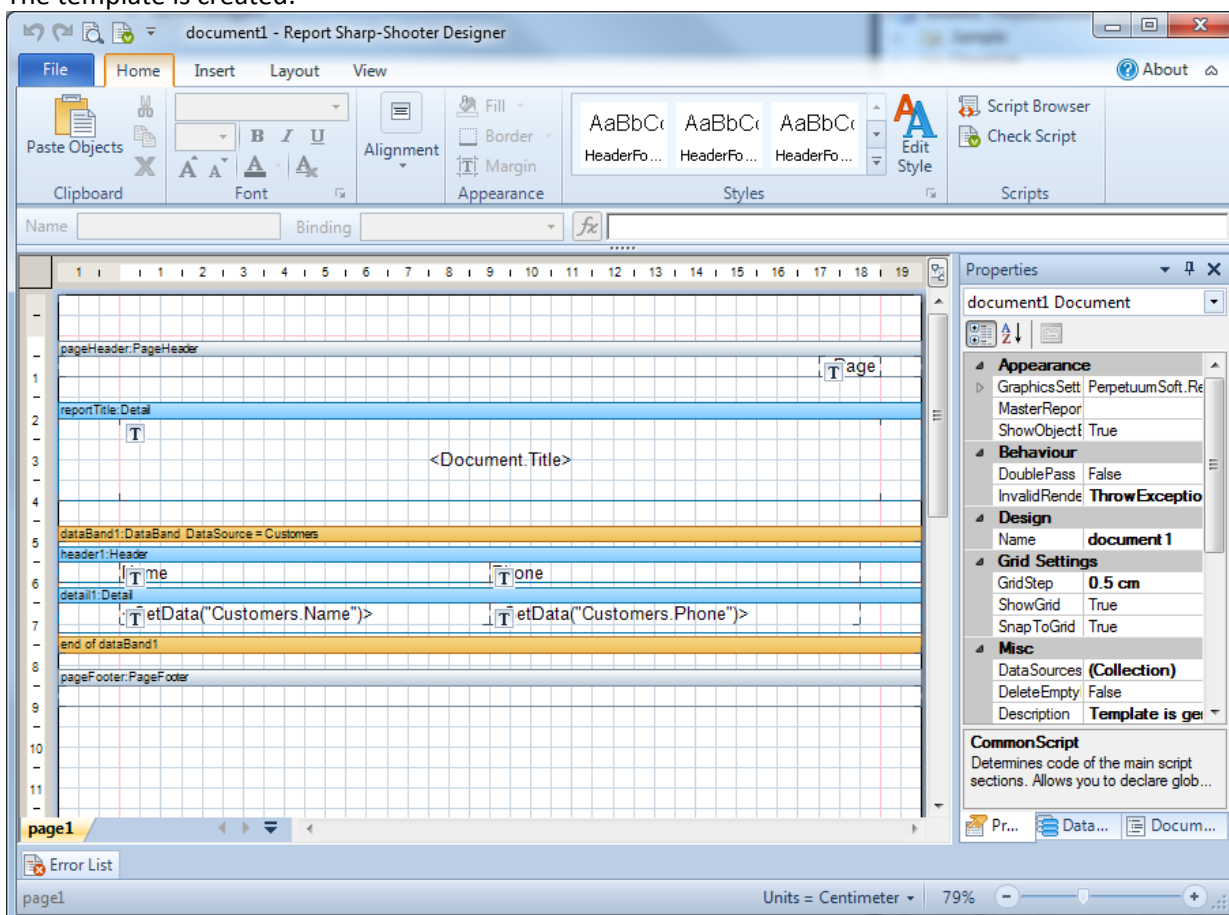
Select fields you want to output in the report (you should move both fields Name and Phone):



Press "OK".

Step 8. Setting up a report

The template is created.

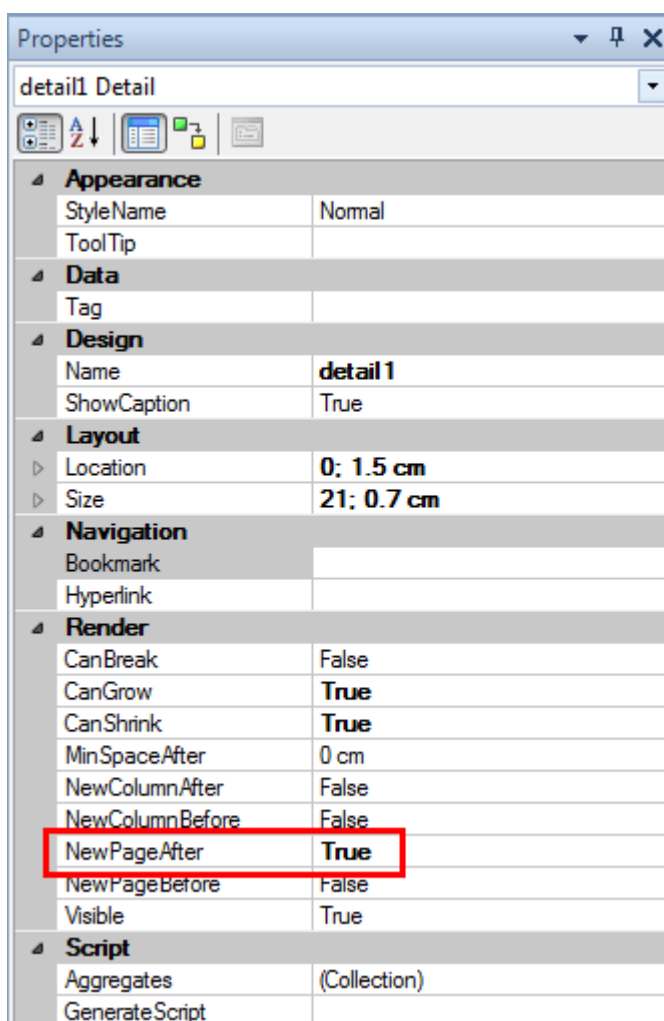


Set the “RepeatEveryPage” property to True for the header1 element in the “Properties” window.

The screenshot shows the Properties window for the 'header1 Header' element. The 'RepeatEveryPage' property is highlighted with a red box and is set to 'True'. Other properties include StyleName (HeaderFooter1), Name (header1), ShowCaption (True), Location (0; 0.5 cm), Size (21; 0.5 cm), and various Render and Script properties.

Appearance	
StyleName	HeaderFooter1
ToolTip	
Data	
Tag	
Design	
Name	header1
ShowCaption	True
Layout	
Location	0; 0.5 cm
Size	21; 0.5 cm
Navigation	
Bookmark	
Hyperlink	
Render	
CanBreak	False
CanGrow	True
CanShrink	True
MinSpaceAfter	0 cm
NewColumnAfter	False
NewColumnBefore	False
NewPageAfter	False
NewPageBefore	False
RepeatEveryColumn	False
RepeatEveryPage	True
Visible	True
Script	
Aggregates	(Collection)
GenerateScript	

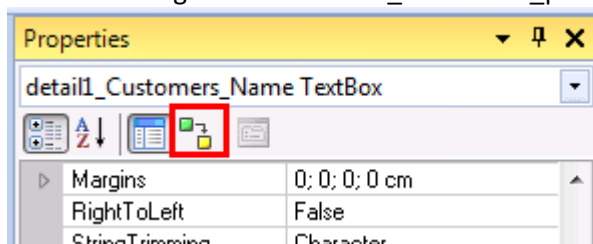
Set the “NewPageAfter” property to True for the detail1 element in the “Properties” window.




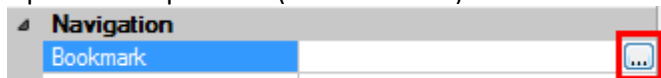
Step 9. Adding navigation

Let's add navigation to a report. Bookmarks tree will be used for navigation through the report.

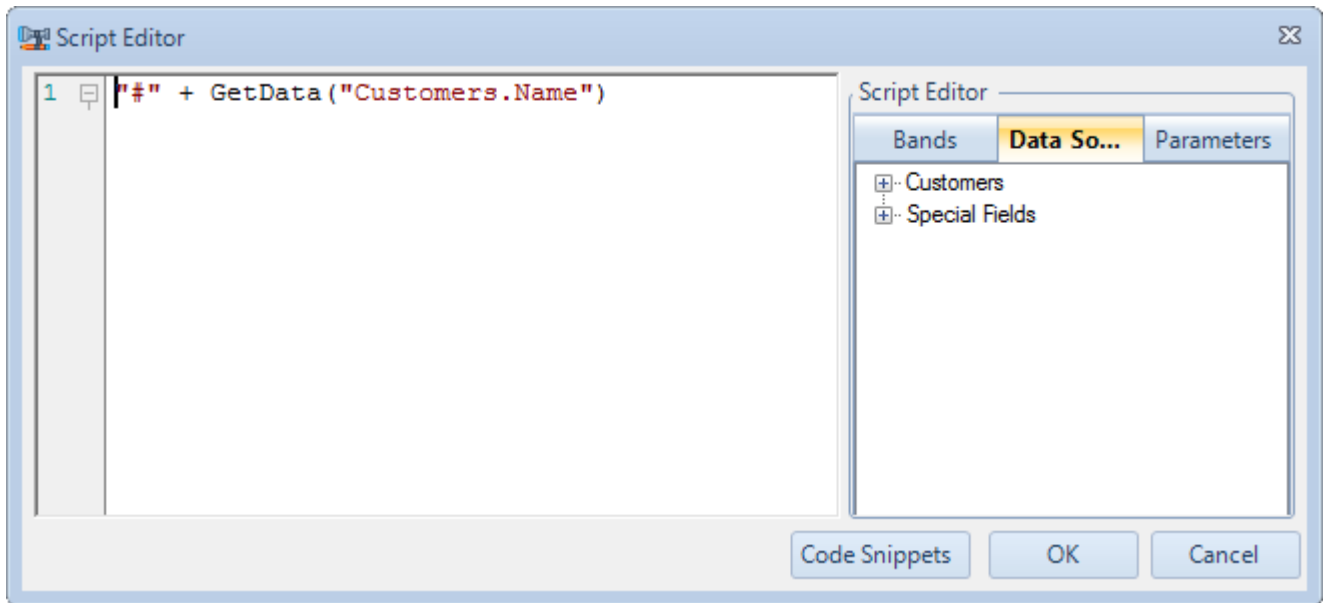
Select "Bindings" for the detail1_Customers_phone element in the "Properties" window.



Open the script editor (the  button) for "Bookmark" property.



Enter "#" + in the script editor. Expand the "Customer" data source in the "Data Source" tab. Double click on the "Name" element in order to add the script code which will get customer name.



Script Editor

Bands Data So... Parameters

- Customers
- Special Fields

Code Snippets OK Cancel

Now you should save the template and close designer.

If you have any questions regarding the integration with SharpShooter Reports.Web don't hesitate to contact us at support@perpetuumsoft.com.