



# SharpShooter Reports.Silverlight

## 基本的な使い方

---

Last modified on: August 16, 2012

※本ドキュメント内のスクリーンショットは英語表記ですが SharpShooter Reports JP(日本語版)では日本語で表示されます。



## 目次

はじめに.....	3
システムの必要条件.....	3
サンプルアプリケーションの作成.....	3
手順 1.....	3
手順 2.....	4
手順 3.....	6
手順 4.....	8
手順 5.....	19
手順 6.....	22
終わりに.....	23



## はじめに

このユーザーガイドは、**SharpShooter Reports.Silverlight** の使い方を説明し、本コンポーネントを使用するために必要な情報を提供することを目的としています。**SharpShooter Reports.Silverlight** を使った Web アプリケーションの作成方法について順を追って説明します。サービスの作成および設定方法、レポートのデザイン方法、アプリケーションのページにレポートビューアを統合する方法を説明します。

## システムの必要条件

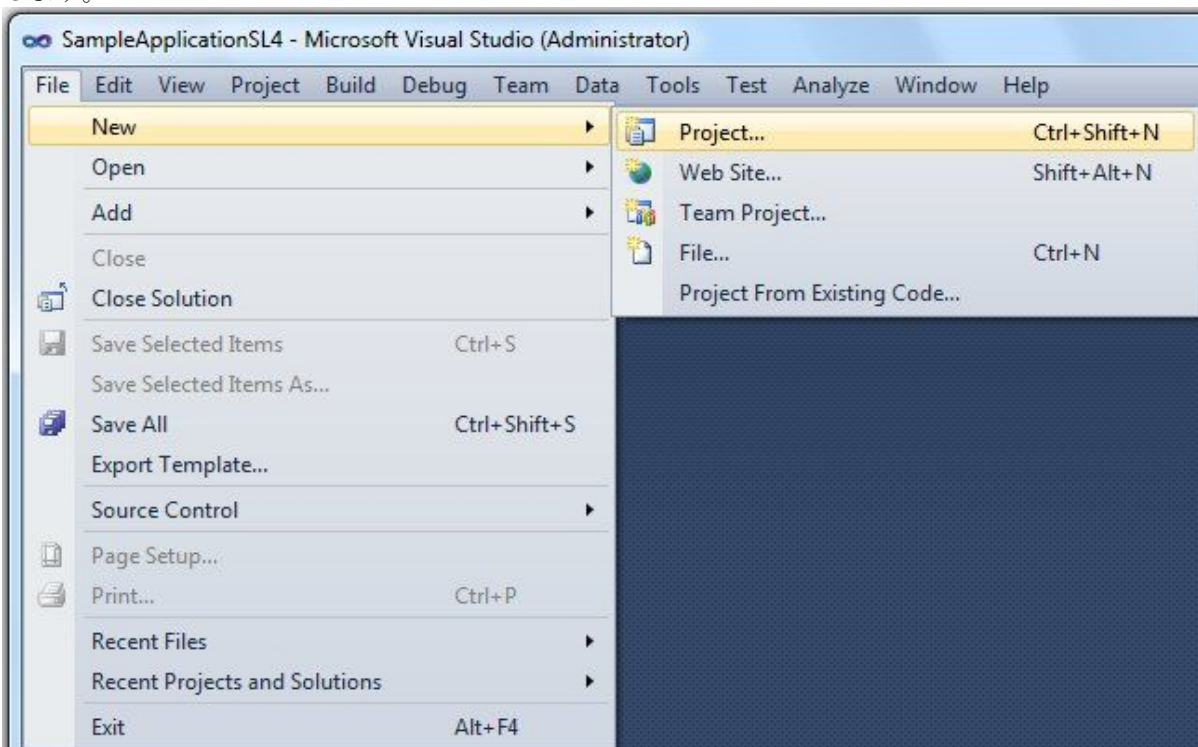
サンプルを動かすには、以下のソフトウェアをインストールする必要があります。

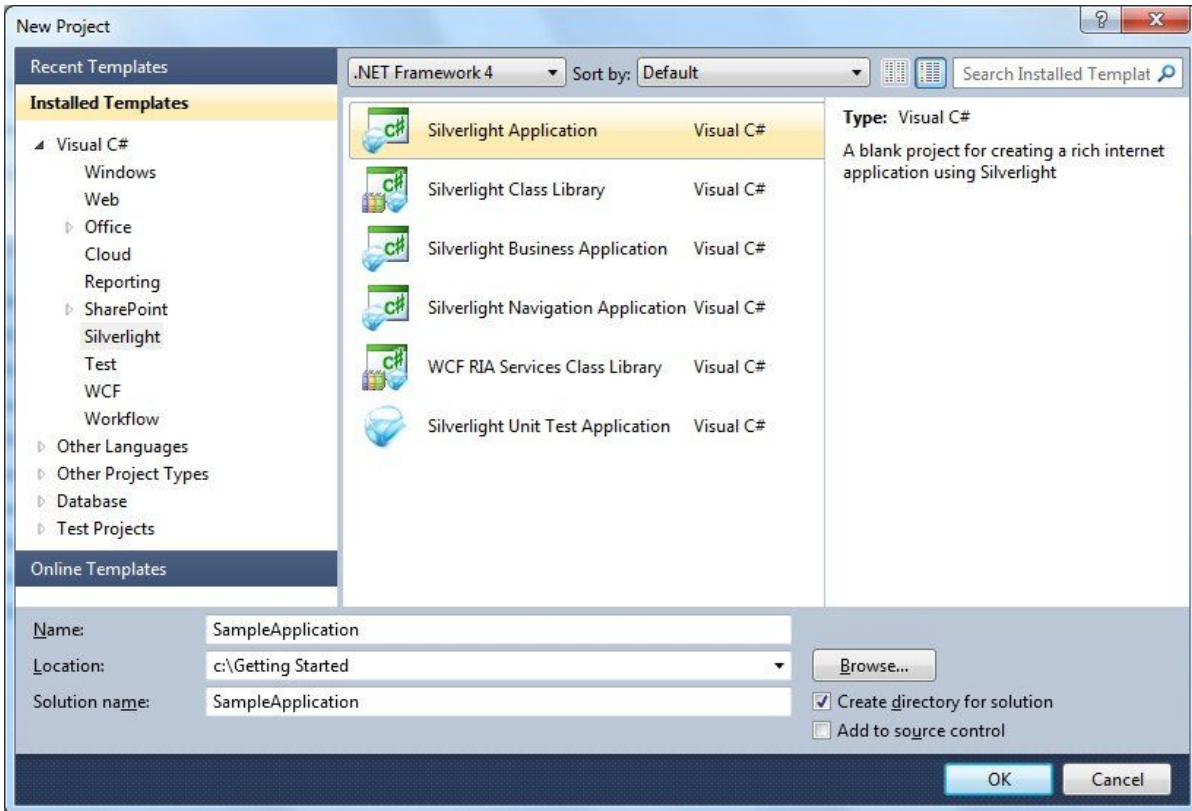
- MS Visual Studio 2010
- .NET Framework 3.5
- ASP.NET 2.0
- Silverlight 4.0
- Microsoft Silverlight 4 Toolkit
- SharpShooter Reports.Silverlight 6.0.0.0 またはそれ以上

## サンプル アプリケーションの作成

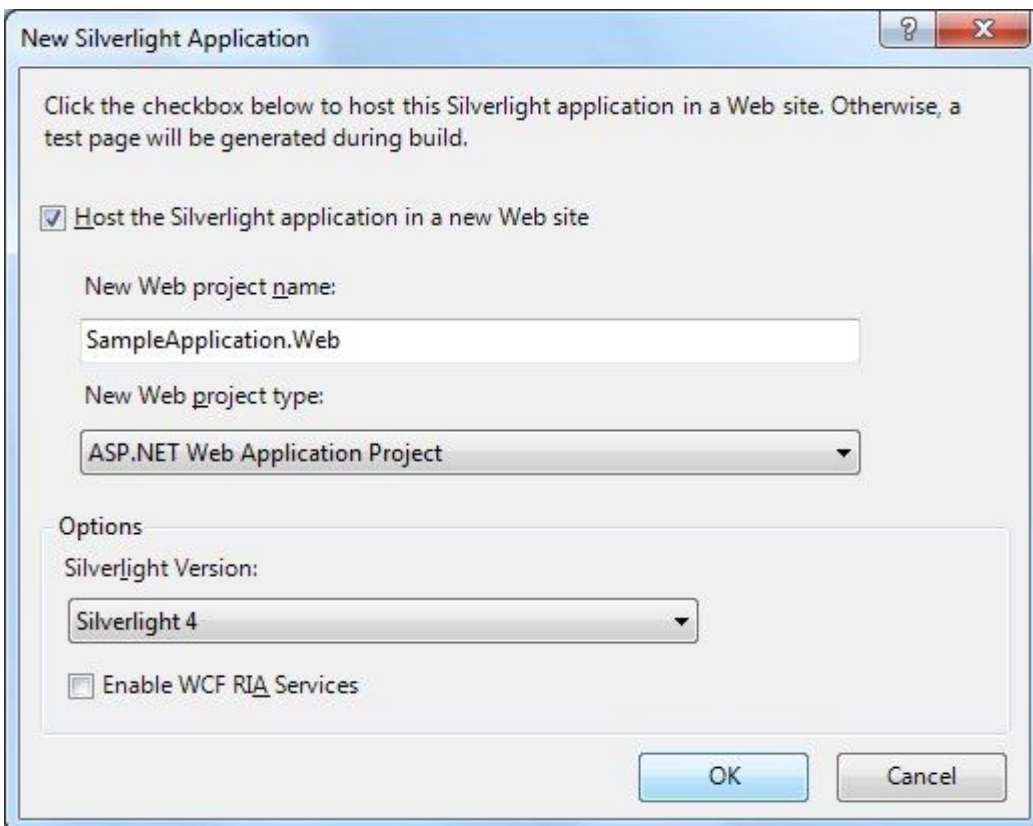
### 手順 1

新規プロジェクト「Silverlight アプリケーション」を作成し、プロジェクト名を「SampleApplication」にします。





プロジェクトの作成時に「Silverlight アプリケーションを新しい Web サイトで ホストする」にチェックを付け、Web プロジェクト名を「SampleApplication.Web」にします。



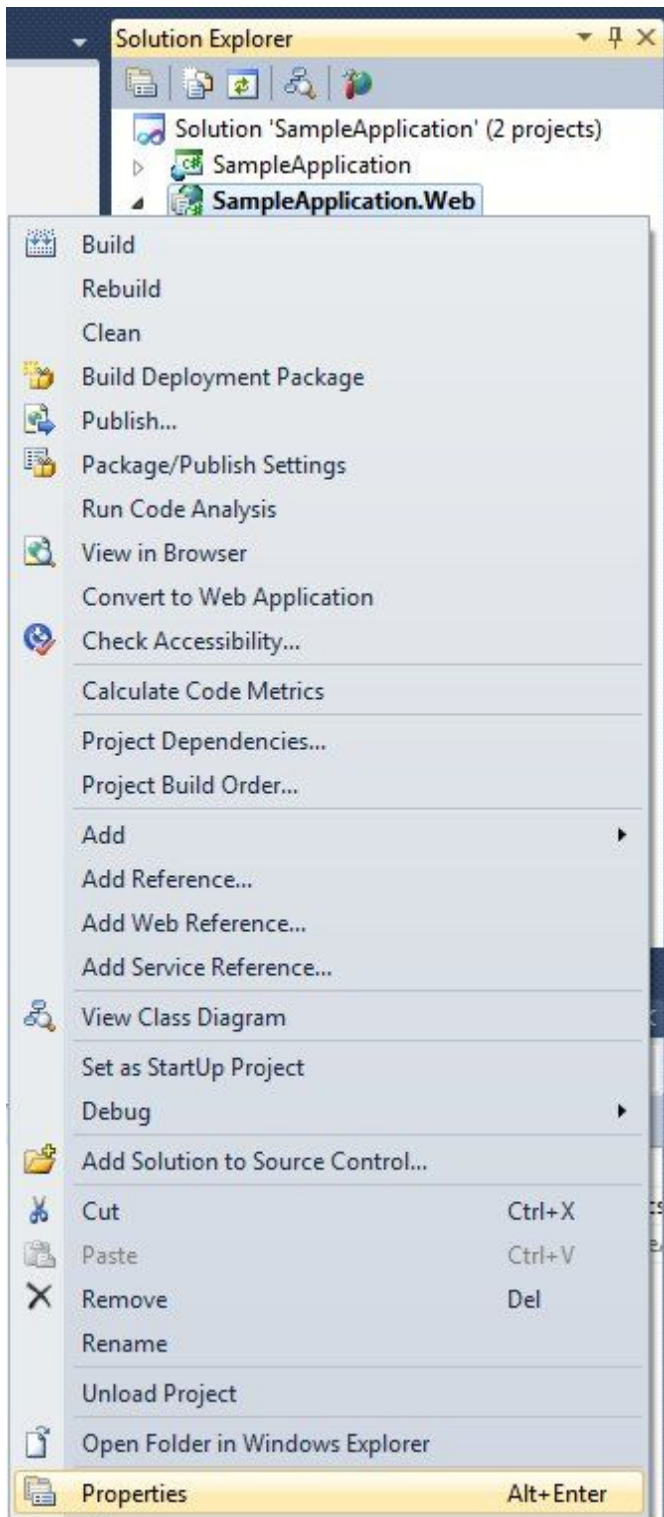
そうすると VS が、2つのプロジェクトを含んだ新しいソリューションを作成します。

## 手順 2

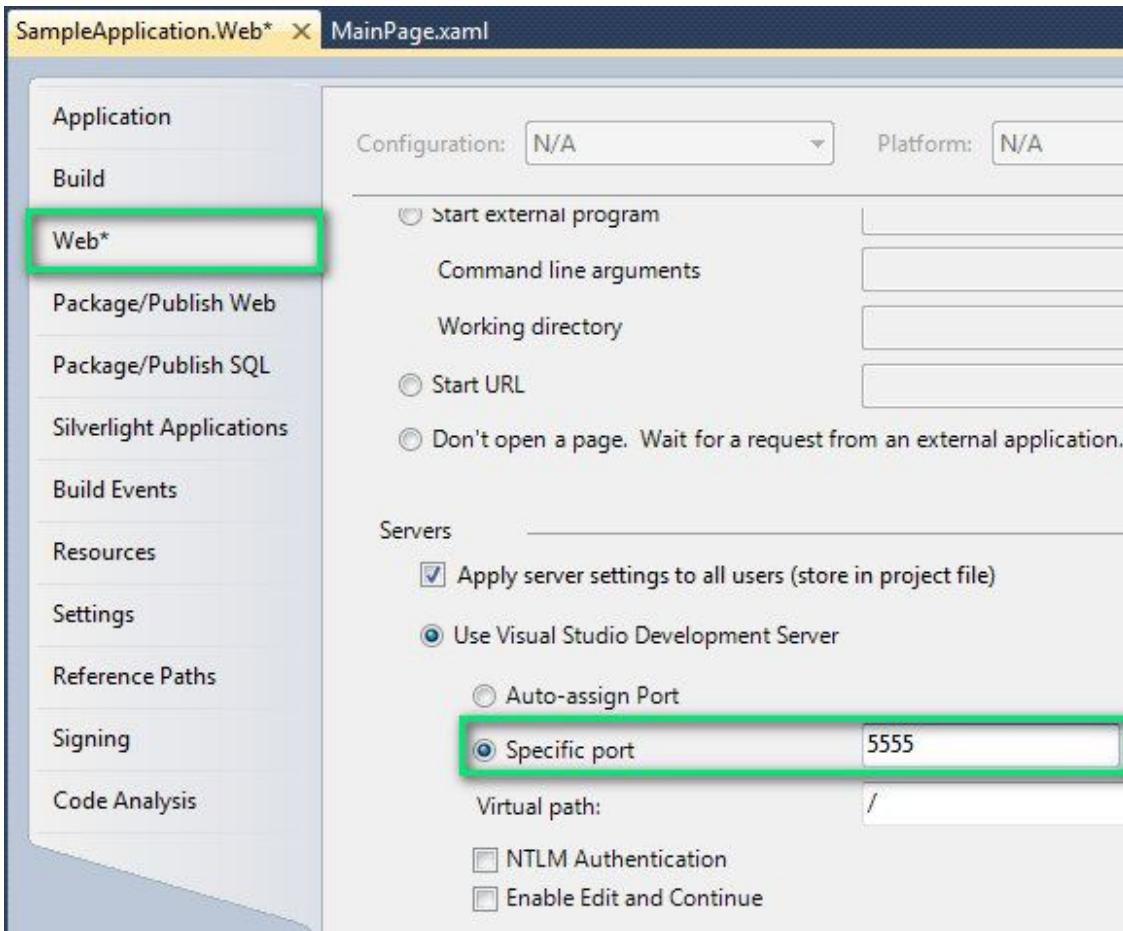
静的ポート 5555 を使用するために Web アプリケーションを設定します。



「SampleApplication.Web」の [プロパティ] を開きます。



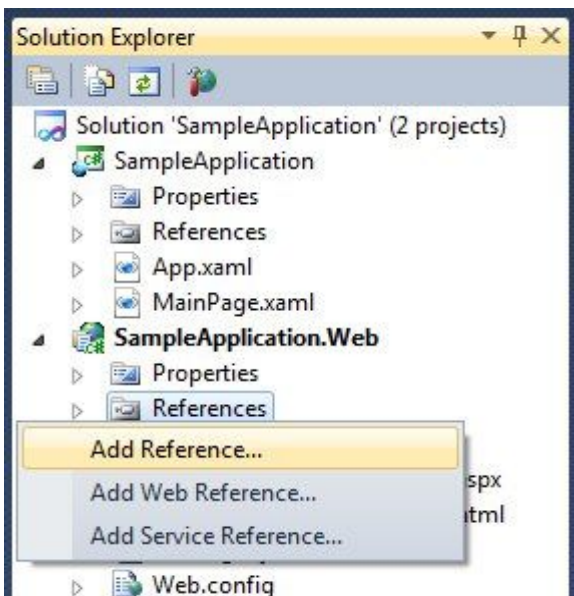
「SampleApplication.Web」の [プロパティ] の Web タブで「ポートを指定する」にチェックを付けて、5555 を設定します。



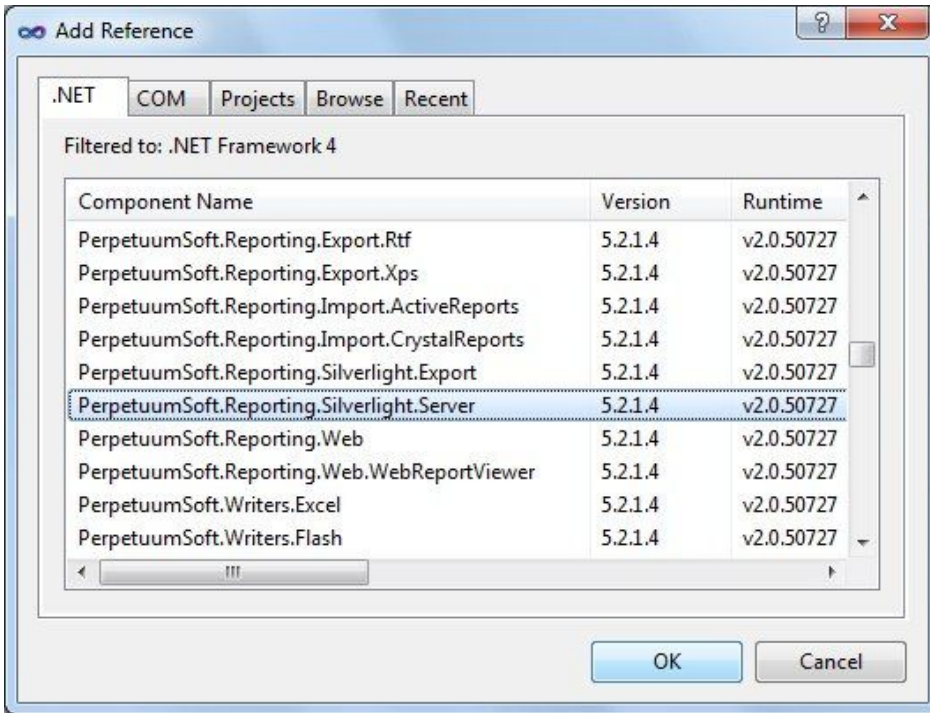
### 手順 3

#### WCF サービスの追加と設定

クライアントとサーバーの情報をやり取りできるようにするには特別なサービスが必要です。このサービスを作成する前に、**SampleApplication.Web** プロジェクトの参照設定を右クリックし、ポップアップメニューから [参照の追加] を選択して **PerpetuumSoft.Reporting.Silverlight.Server.dll** の参照を追加しなければなりません。

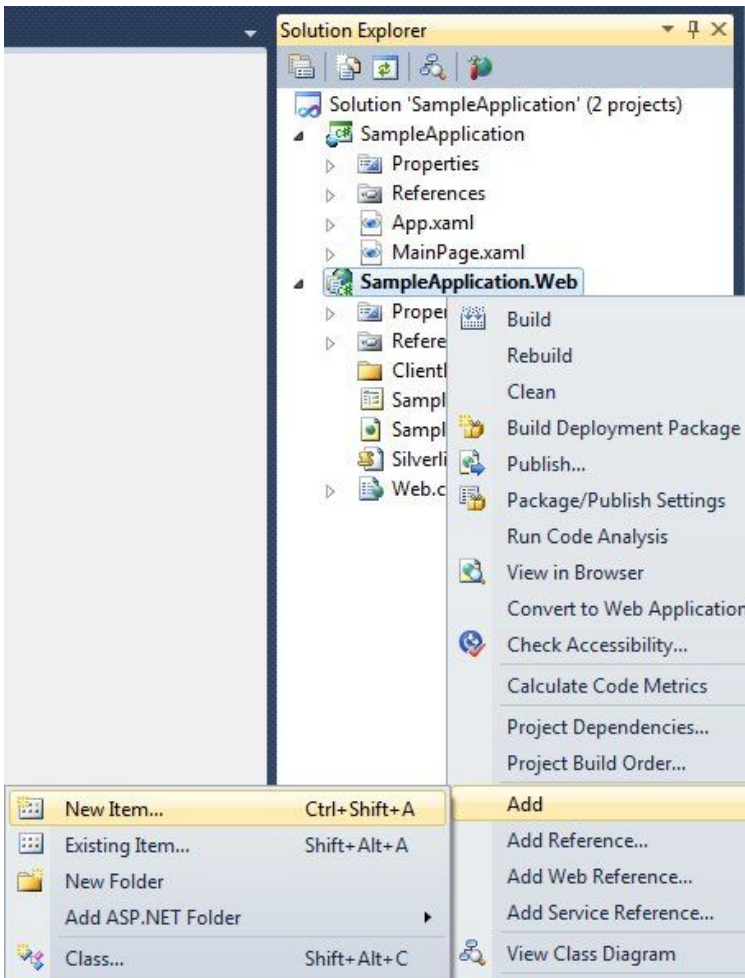






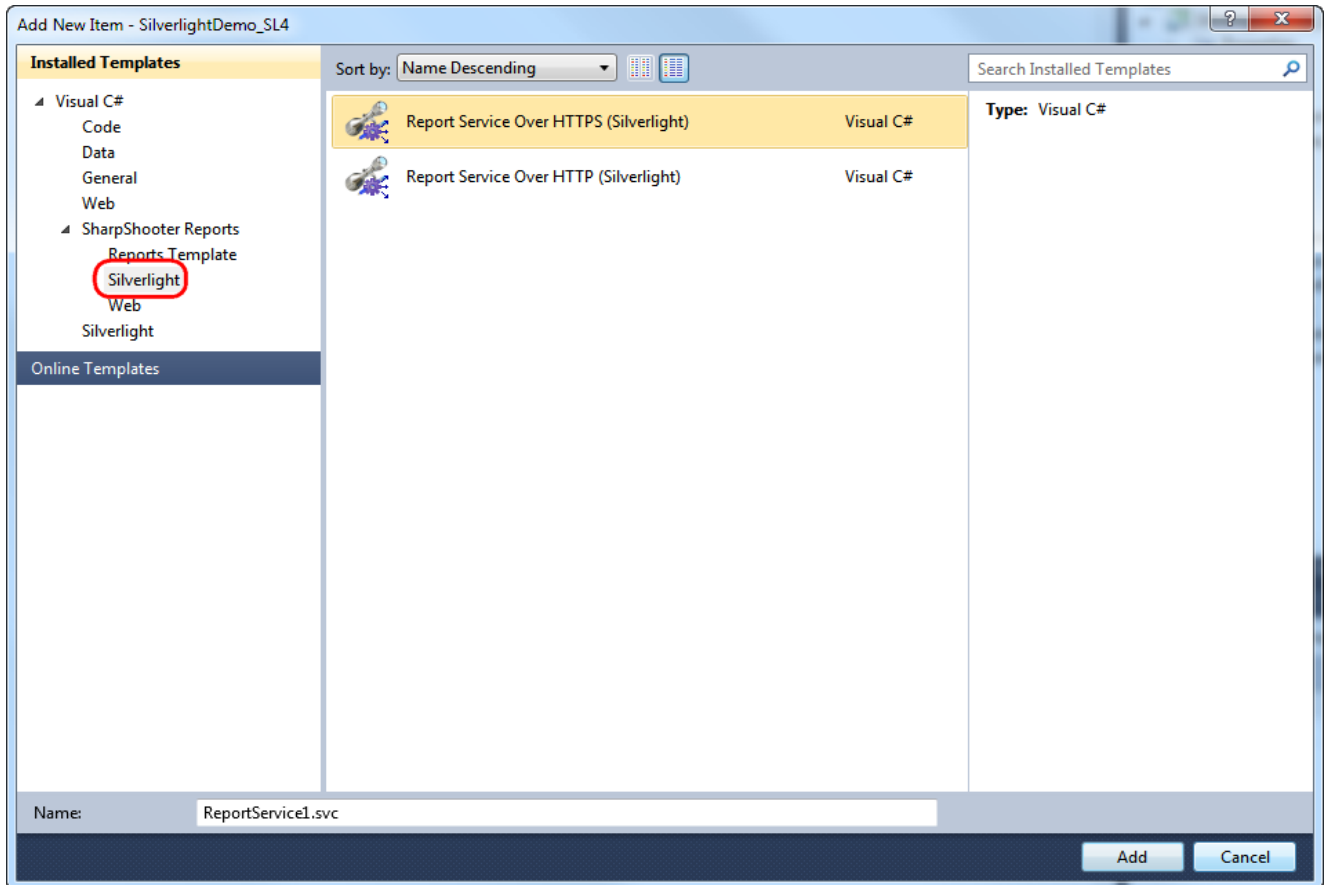
SampleApplication.Web プロジェクトに「Report Service Over HTTP」を追加し、名前を「ReportService1.svc」にします。

それを行うには、SampleApplication.Web を右クリックし、ポップアップメニューから [追加]->[新しい項目] を選択します。





プロジェクトに、「Report Service Over HTTP」テンプレートを追加します。



ReportService クラスは、Silverlight のレポートビューアの WCF サービスの実装を含んでいる PerpetuumSoft.Reporting.Silverlight.Server.ReportServiceBase クラスを拡張します。

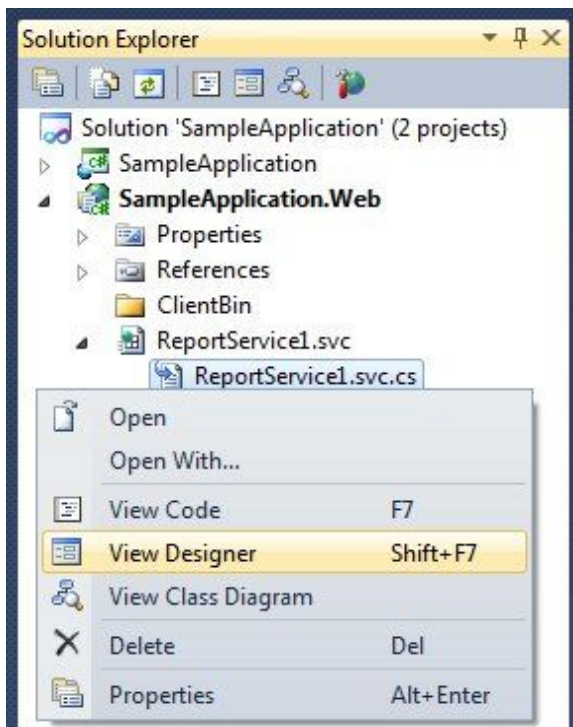
#### 手順 4

レポートテンプレートの作成

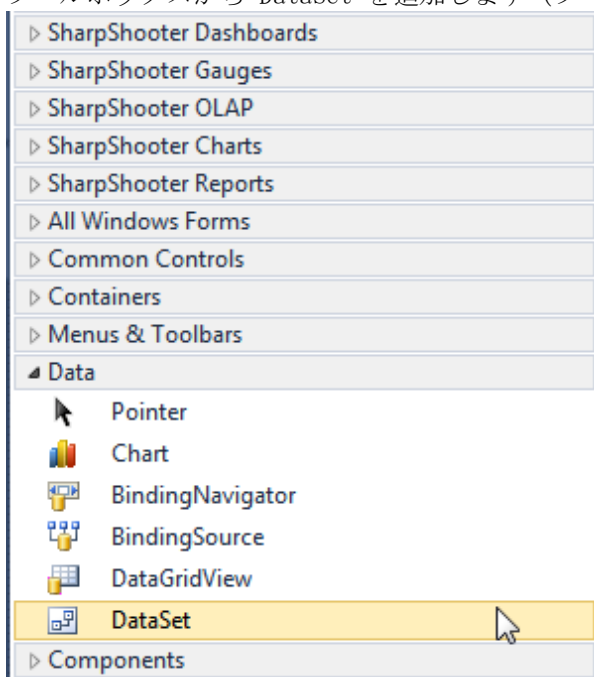
まず、データソース構造を作成します。デザイナーでサービスを開きます。

それには、ReportService1.svc.cs を右クリックしてポップアップメニューから [デザイナーの表示] を選択します。

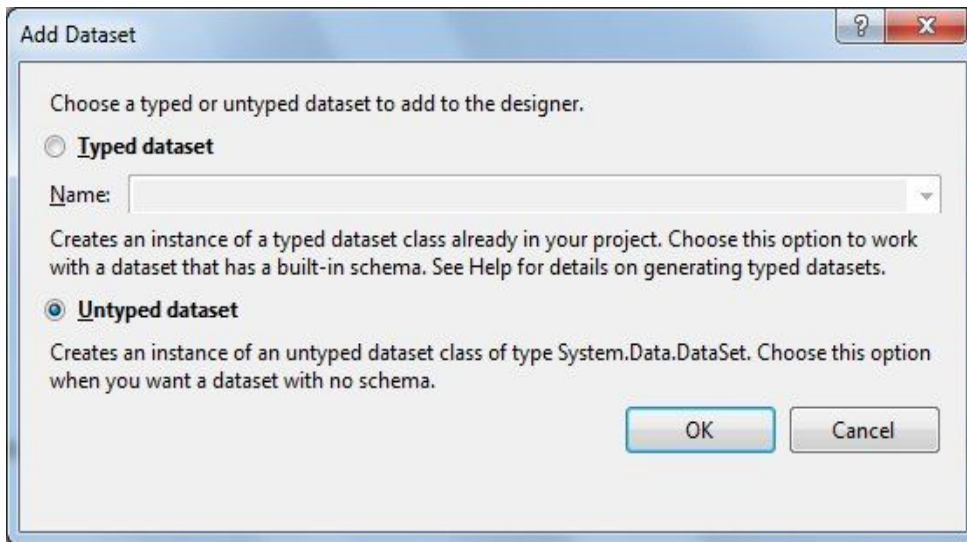




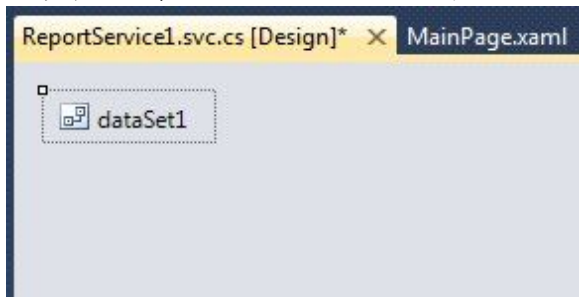
ツールボックスから DataSet を追加します（ツールボックスの DataSet をダブルクリックします）。



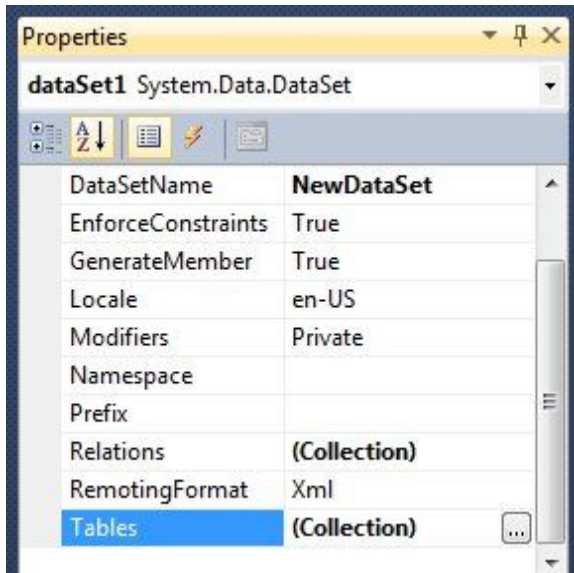
そして、「型指定のないデータセット」を選択します。




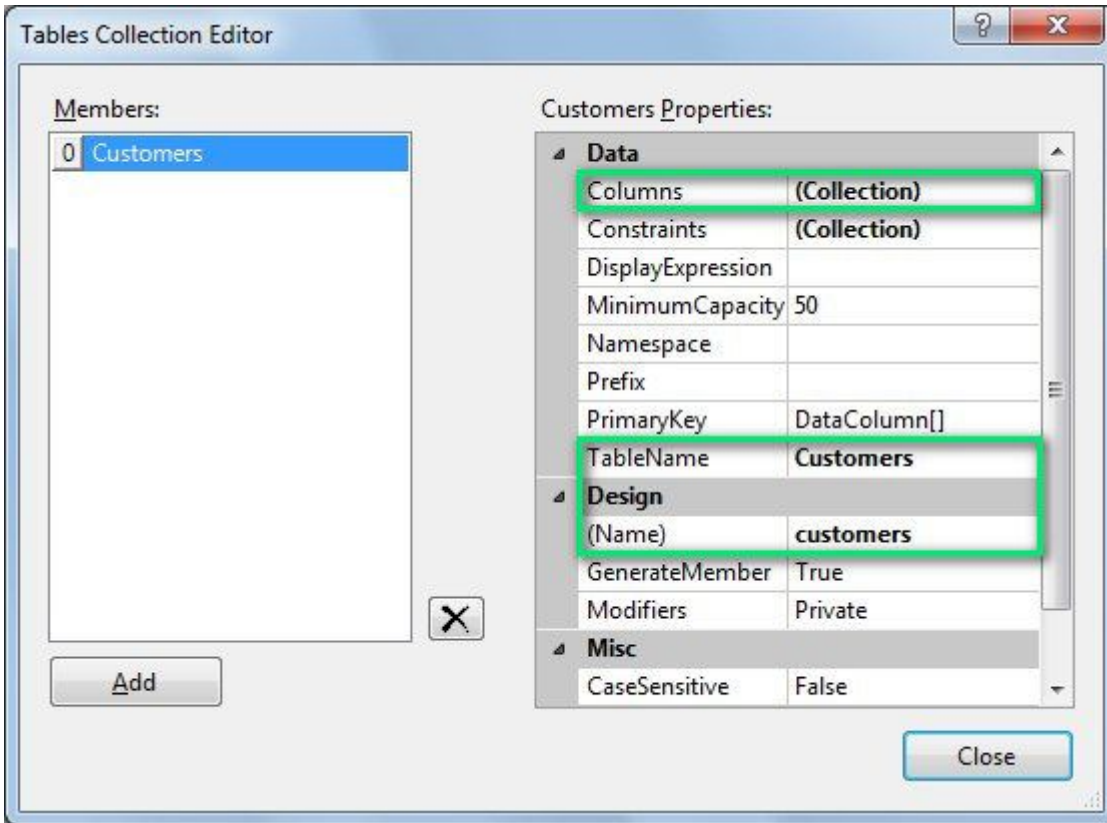
そうすると、デザイナーにデータセットのノード (**dataSet1**) が表示されます。



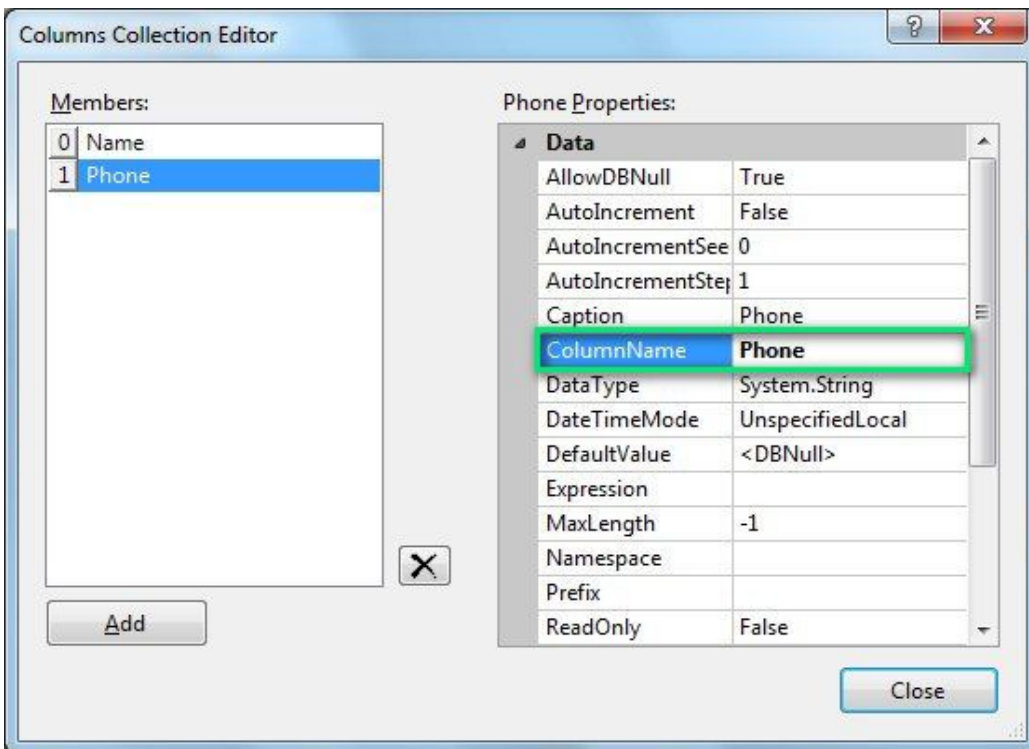
テーブル コレクション エディタ (Tables プロパティの  ボタンをクリック) を開きます。



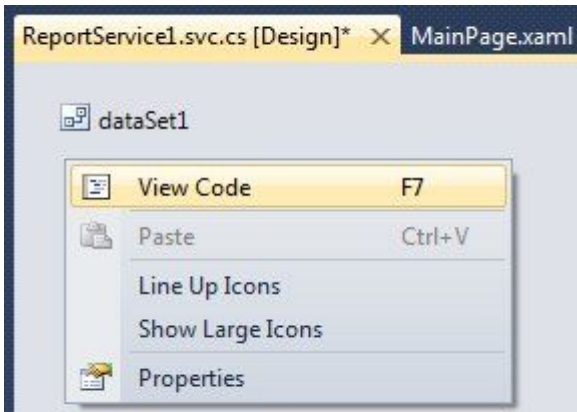
dataSet1 に「Customers」テーブルを追加します（「追加」ボタンをクリックし、TableName プロパティの値を「Customers」に、Name プロパティの値を「customers」に設定します）。そして、列コレクション エディタ (Tables コレクション エディタのプロパティグリッドの  ボタンをクリック) を開きます。



「追加」ボタンをクリックして列を2つ追加し、ColumnName プロパティの値をそれぞれ「Name」と「Phone」に設定します。



データ構造が決まったので、Customers テーブルにデータを設定します。ソースコードを表示するには、デザイナの領域を右クリックし、コンテキストメニューから [コードの表示] をクリックします。



ReportService クラスの OnLoadData オーバーライドメソッドを使用してデータソースに値を設定してください。

[C# の場合]

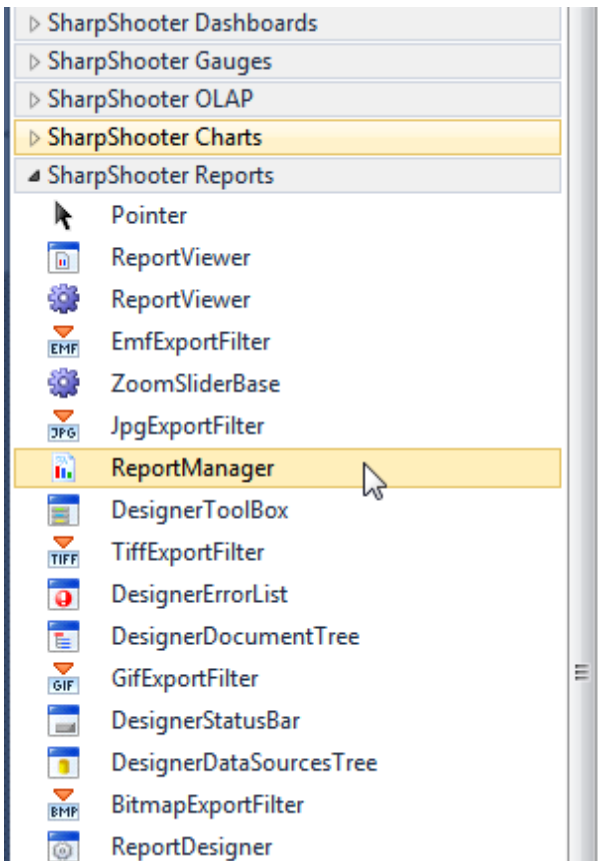
```
protected override void OnLoadData(IDictionary<string, object> parameters, string reportName,
PerpetuumSoft.Reporting.Components.ReportSlot reportSlot)
{
    base.OnLoadData(parameters, reportName, reportSlot);
    DataRow row = customers.NewRow();
    row["Name"] = "Johnson Leslie";
    row["Phone"] = "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);
}
```

[VB.NET の場合]

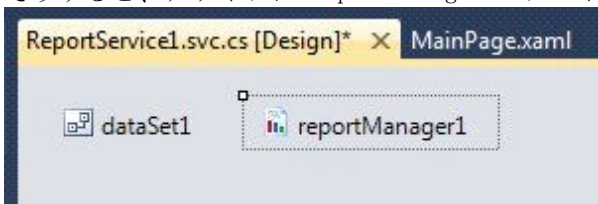
```
Protected Override Sub OnLoadData(ByVal parameters As
System.Collections.Generic.IDictionary(Of String, Object), ByVal reportName As String, ByVal
reportSlot As PerpetuumSoft.Reporting.Components. ReportSlot)
MyBase.OnLoadData(parameters, reportName, reportSlot)
Dim row As DataRow = customers.NewRow()
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("Phne") = "803-438-2771"
customers.Rows.Add(row)
End Sub
```

備考：問い合わせた（クエリされた）ドキュメント情報とそのパラメータは OnLoadData メソッドのパラメータに渡されます。

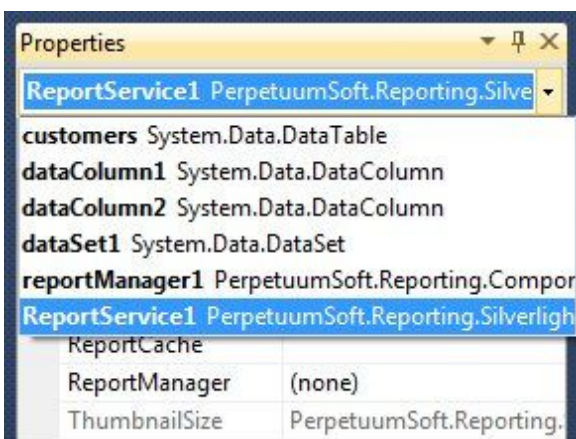
では、ReportManager コンポーネントを追加します（ツールボックスで ReportManager をダブルクリックします）。このコンポーネントはレポート生成を行います。



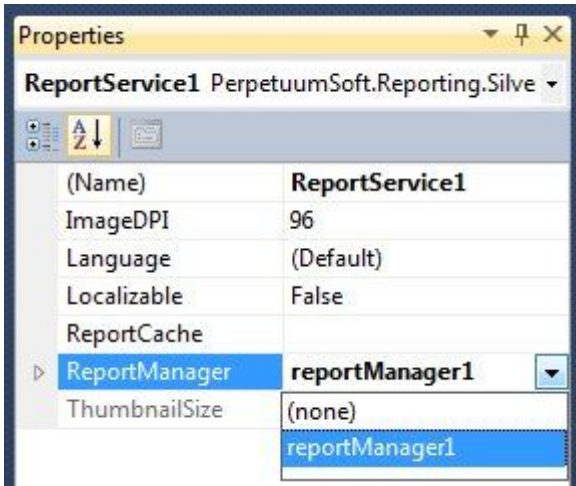
そうすると、デザイナーに ReportManager のノード (reportManager1) が表示されます。



ReportService サービスの ReportManager プロパティを設定します。それには、[プロパティ] ウィンドウから ReportService のプロパティを開きます。

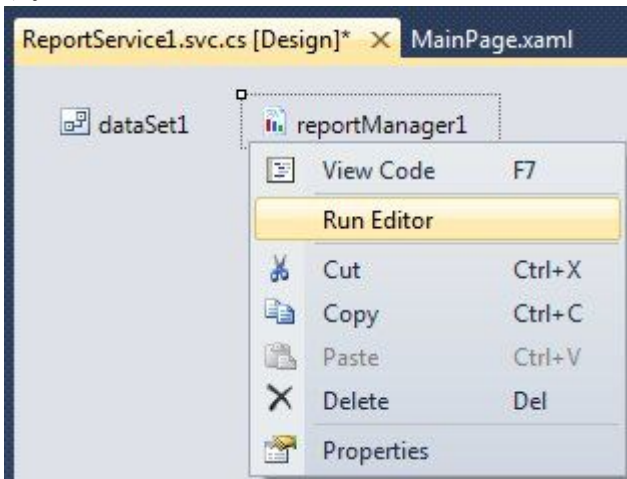


プロパティ一覧から reportManager1 を選択します。



では、レポートを作成します。

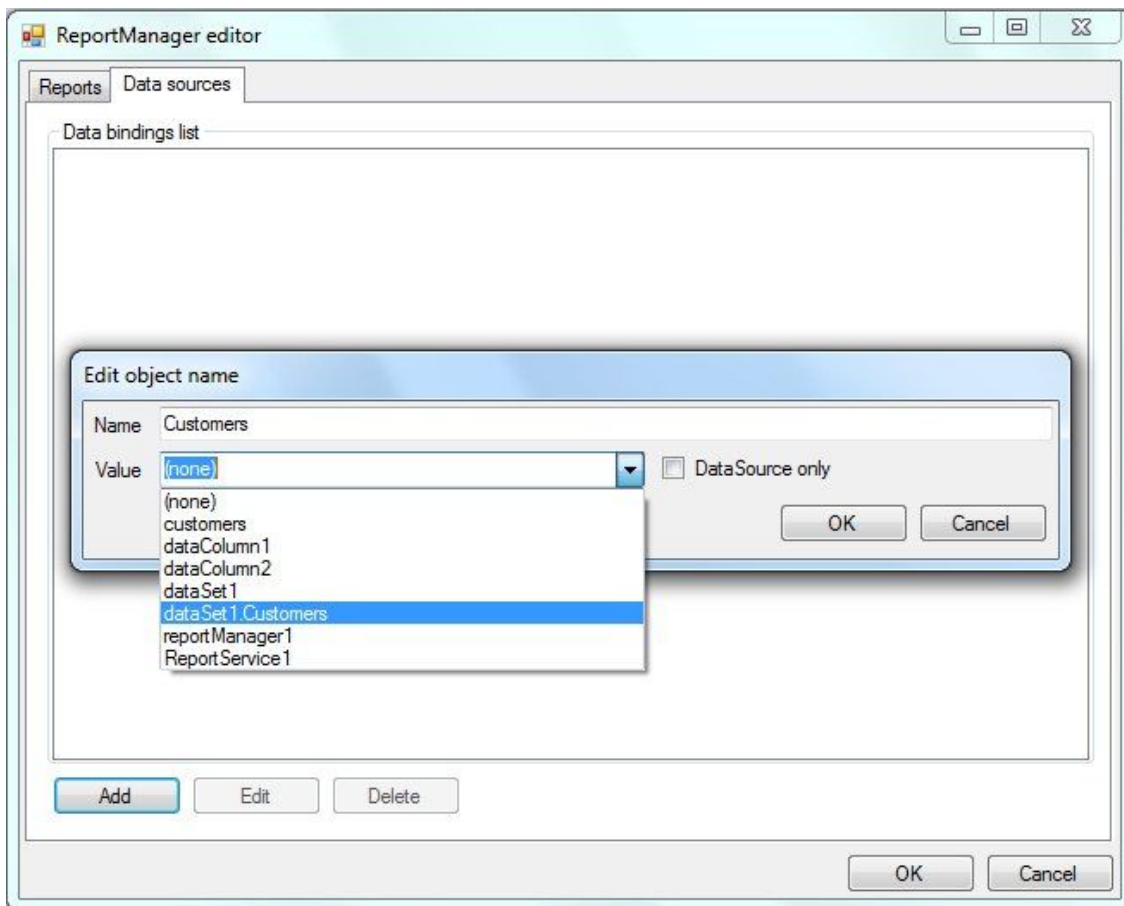
reportManager1 を右クリックし「エディタの起動」を選択して、レポートマネージャのエディタを起動します。



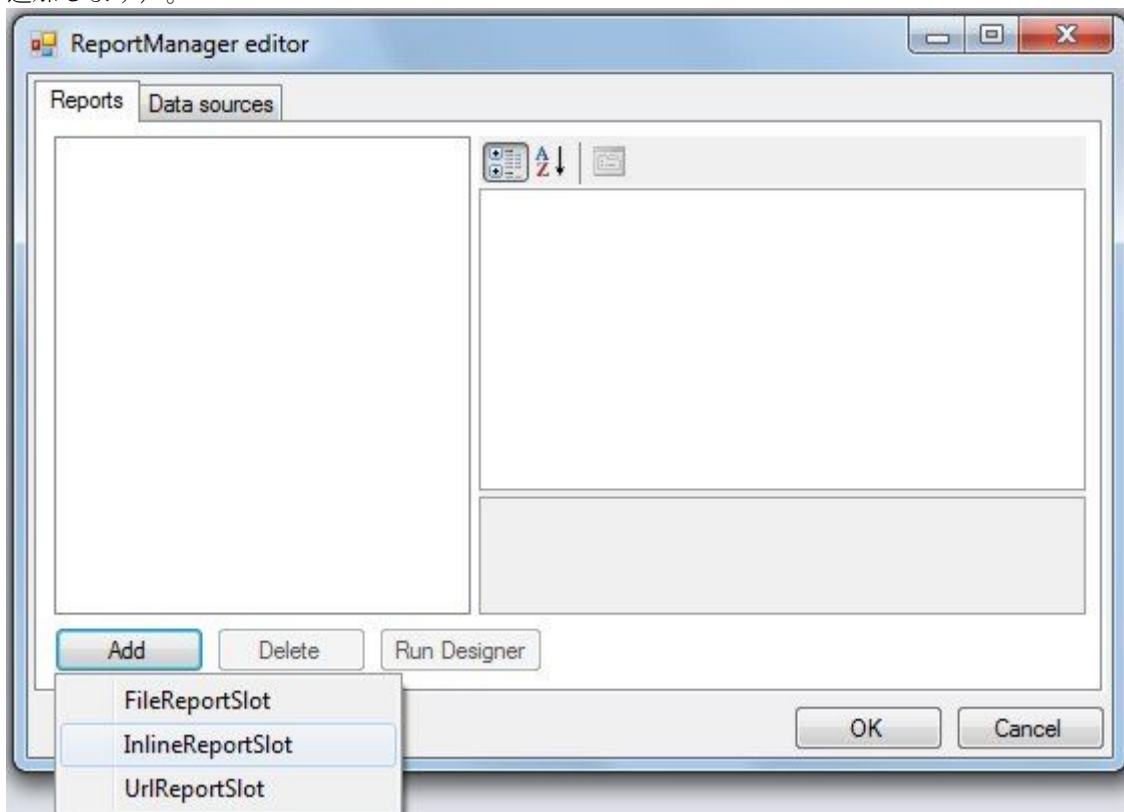
レポートテンプレートを作成する前に、レポートを生成するデータソースを追加します。

「データソース」タブのデータバインドリストに「Customers」テーブルを追加します（「追加」ボタンをクリックすると表示される「オブジェクト名の編集」の「名称」を "Customers" に、「値」にはコンボボックスから dataSet1.Customers を選択します）。

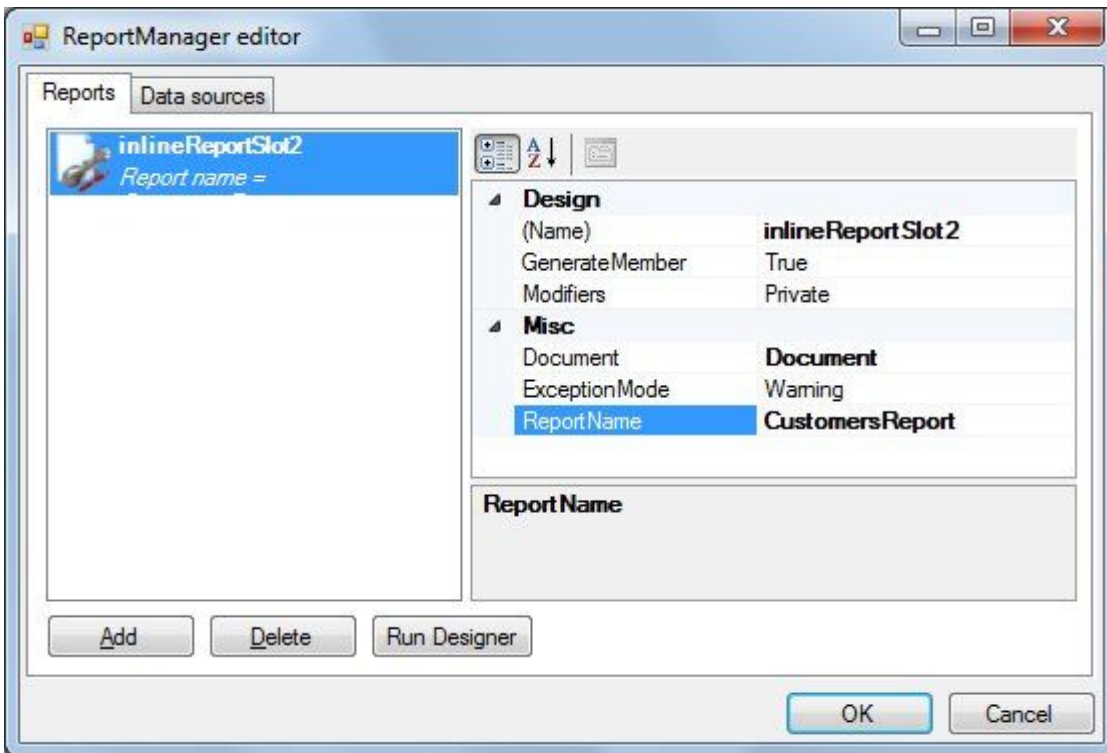




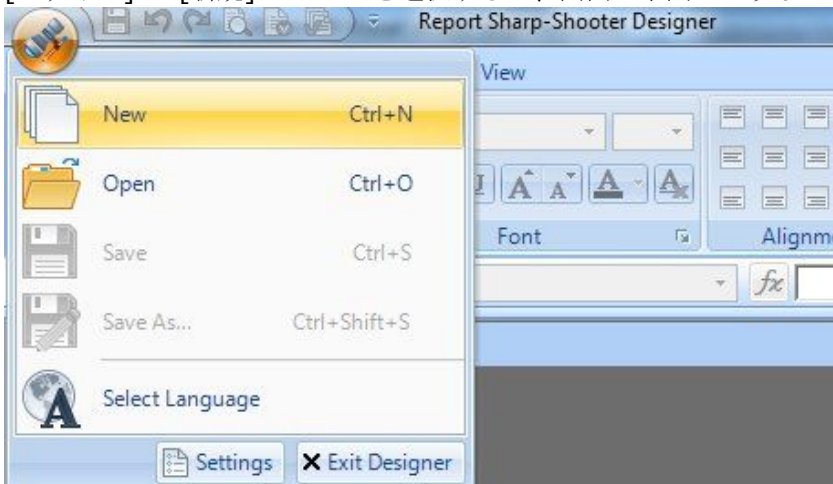
「レポート」タブで新しいオブジェクトを追加します（「追加」ボタンをクリックして **InlineReportSlot** を追加します）。



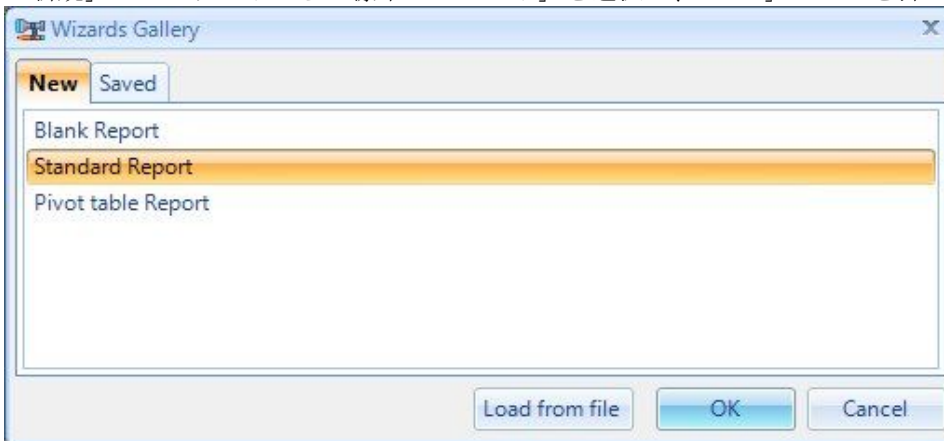
**ReportName** プロパティの値を「**CustomersReport**」に設定します。以後、レポートマネージャからこの名前でドキュメントを取得します。次に、「デザイナーの起動」ボタンを押してレポートデザイナーを立ち上げます。



[ファイル] -> [新規] メニューを選択すると、画面に下図のようなフォームが表示されます。

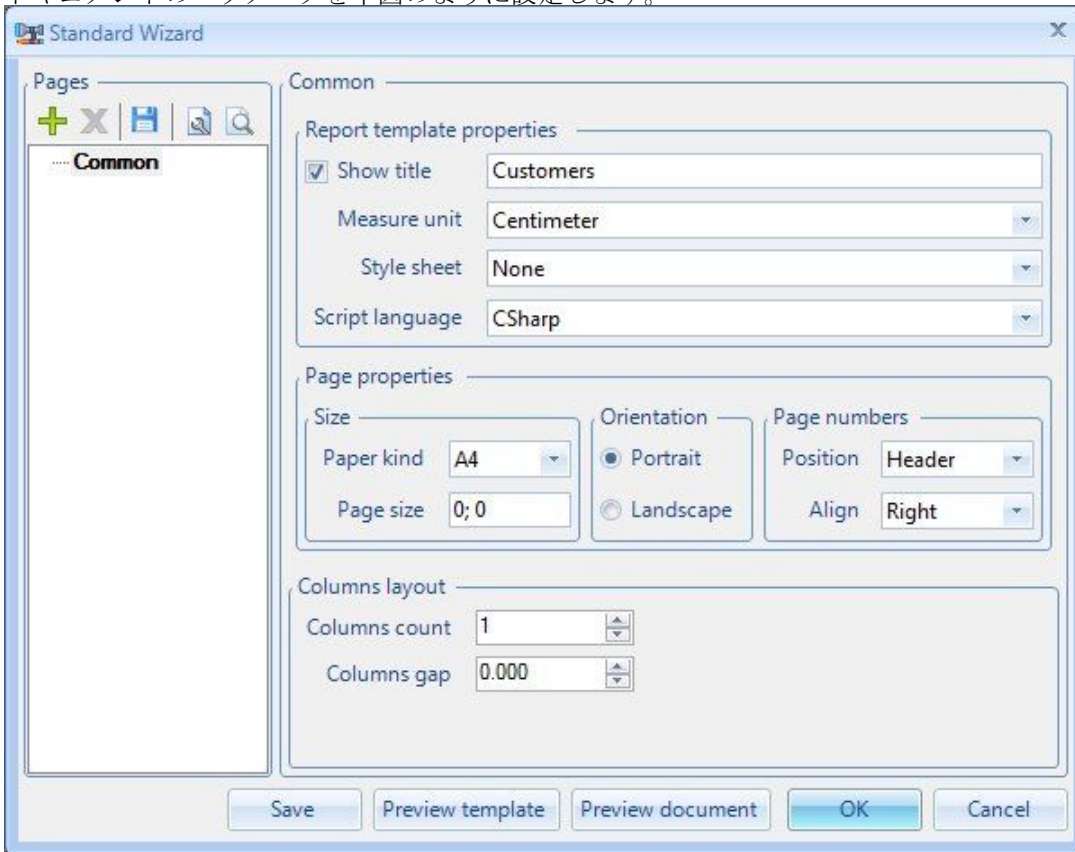


「新規」タブのリストから「標準のレポート」を選択し、「OK」ボタンを押します。

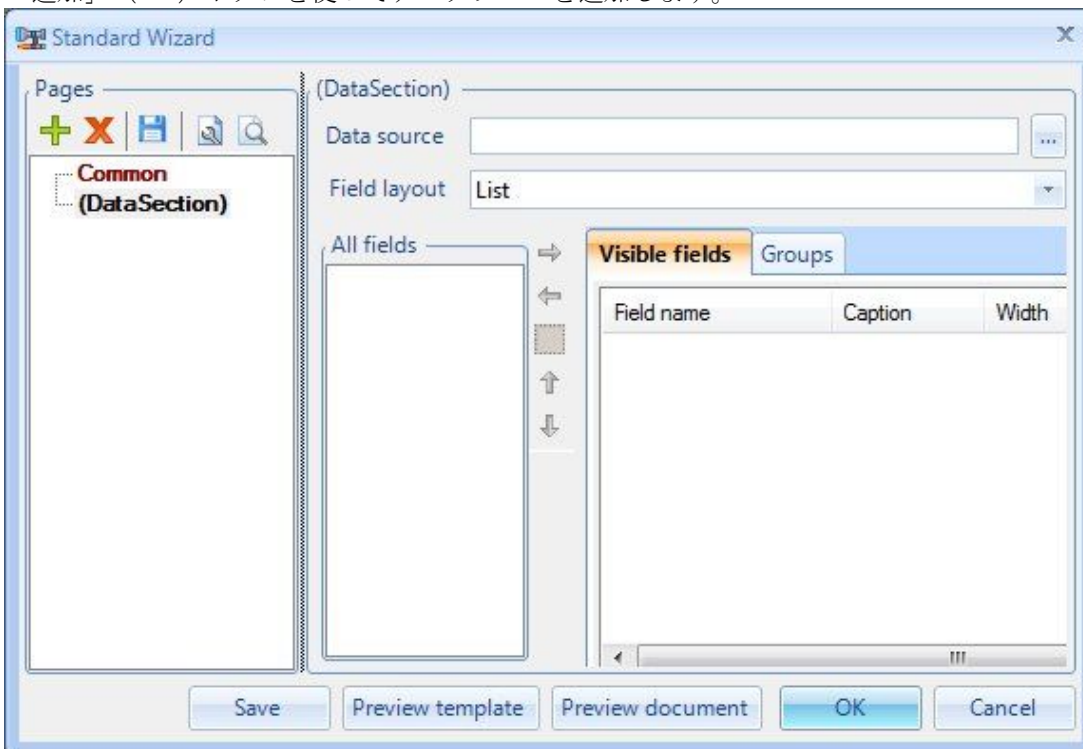


画面に、「スタンダードウィザード」ウィンドウが表示されます。

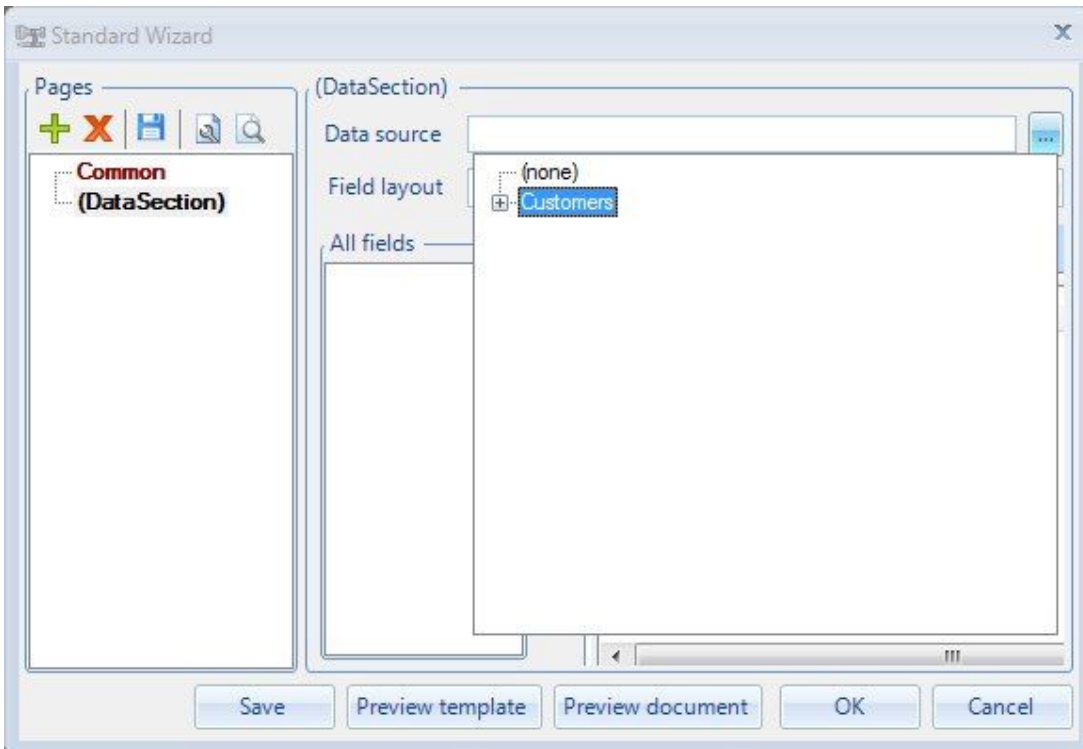
ドキュメントのパラメータを下図のように設定します。



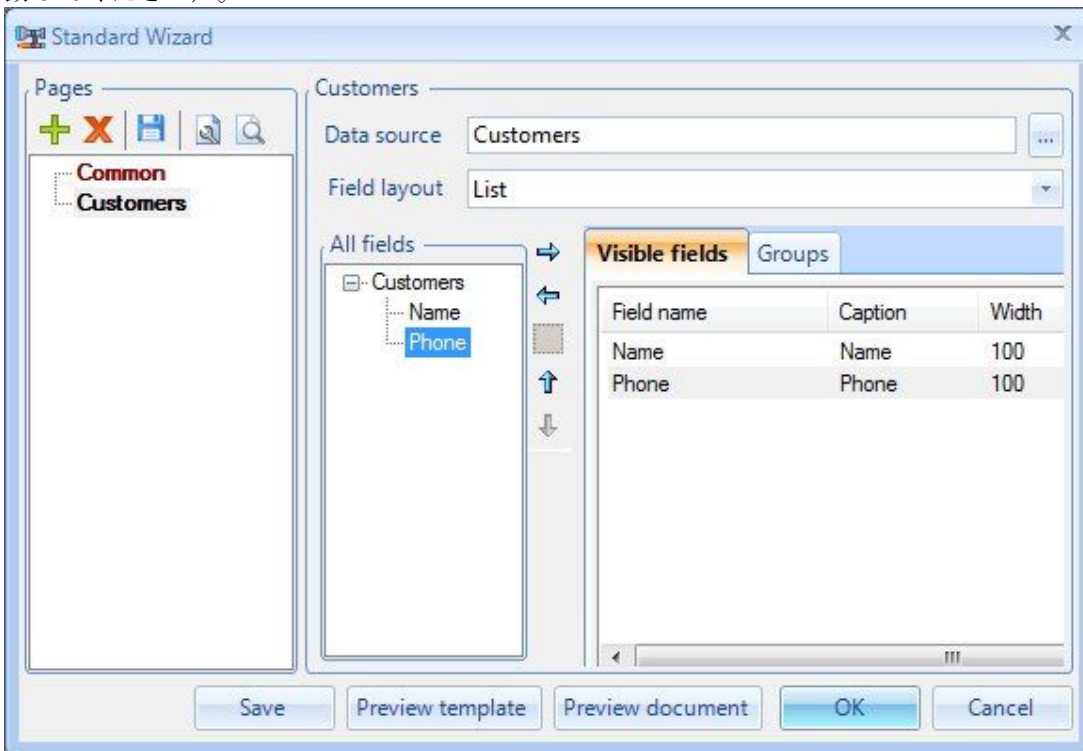
「追加」 (+) ボタンを使ってデータソースを追加します。



データソースフィールドの近くの... ボタンをクリックし、表示されたツリービューから「Customers」をダブルクリックして選択します。



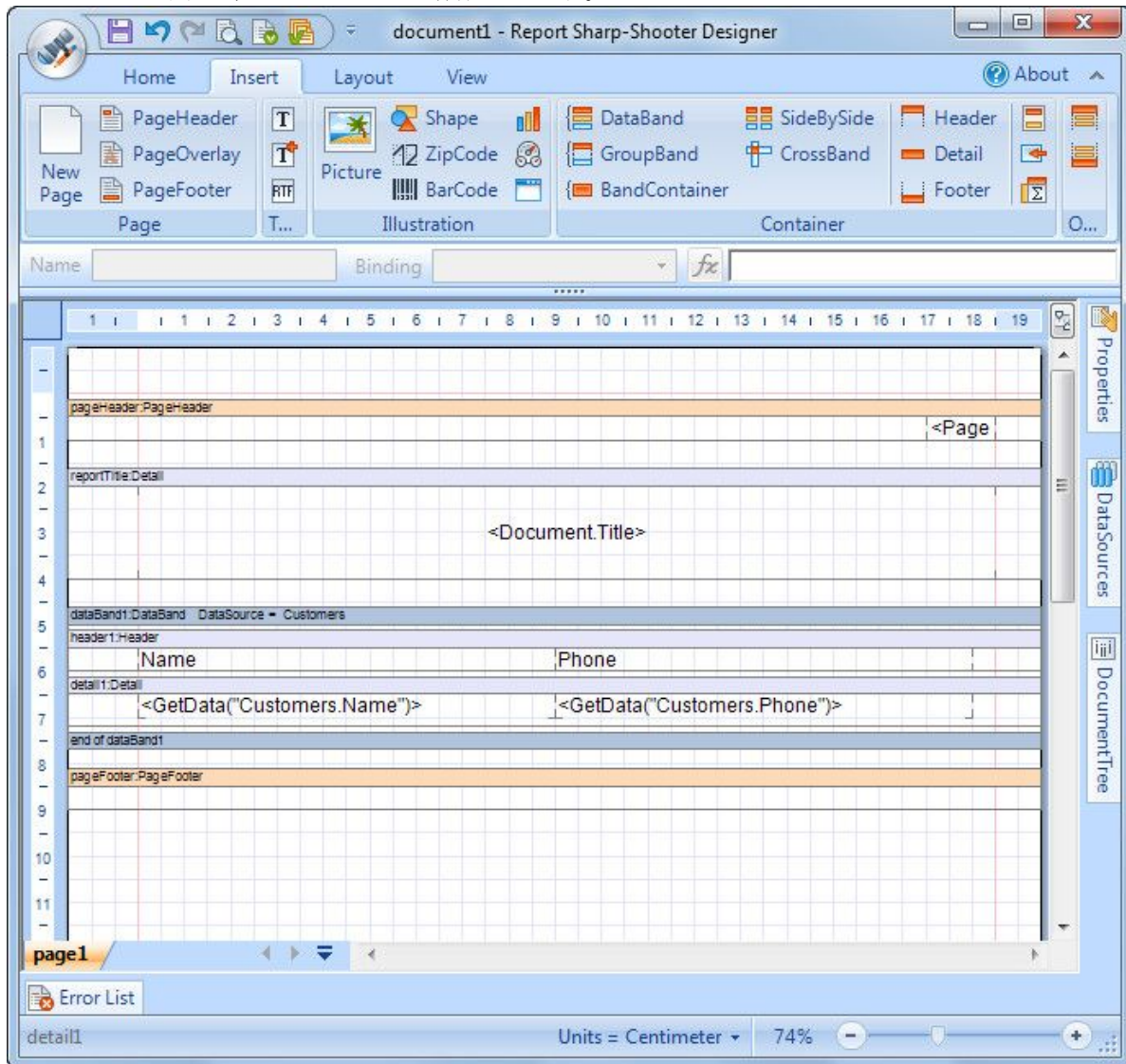
レポートに出力したいフィールドを選択します（フィールドの「Name」と「Phone」フィールドの両方を移動してください）。







OK ボタンを押すと、テンプレートが作成されます。

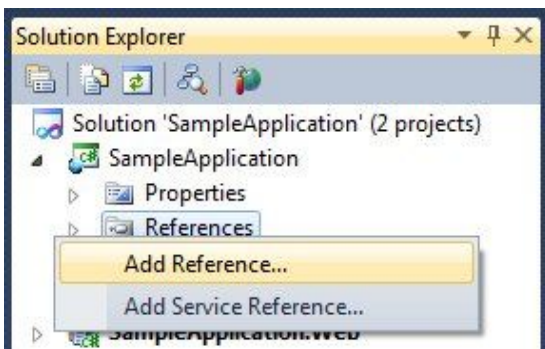


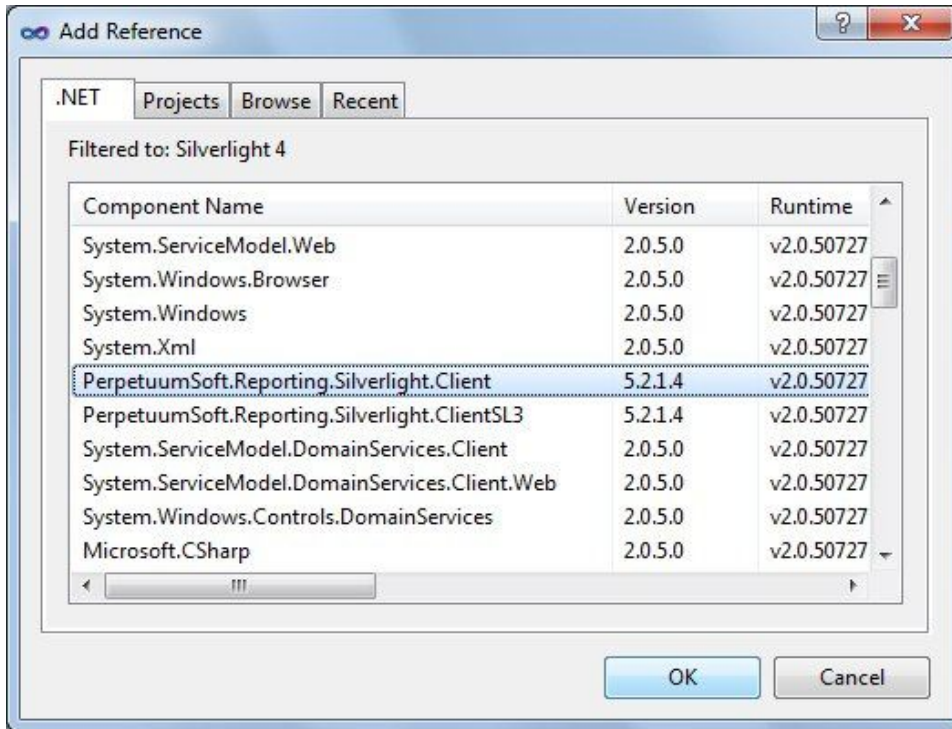
では、テンプレートを保存してデザイナを閉じます。

## 手順 5

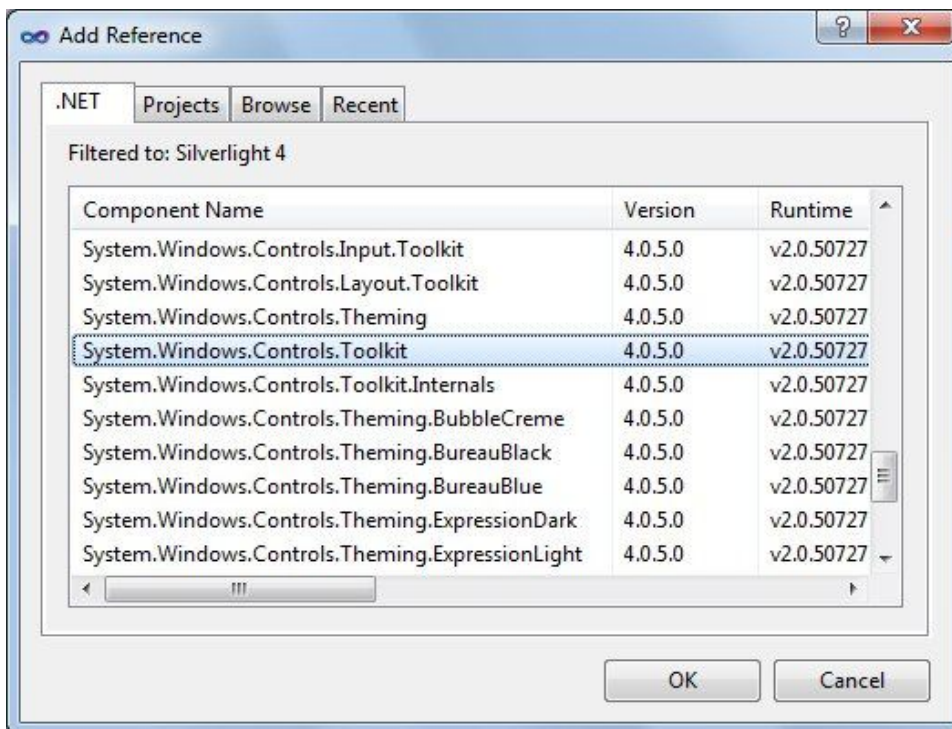
### Silverlight の ReportViewer コンポーネントの追加と設定

レポートを表示するために、**Silverlight** アプリケーションにレポートビューア コンポーネントを追加します。それには、**ReportViewer** を格納している **PerpetuumSoft.Reporting.Silverlight.Client** アセンブリの参照を追加してください（ソリューション エクスプローラの **SampleApplication** の参照設定を右クリックし、ポップアップメニューから [参照の追加] を選択してください）。



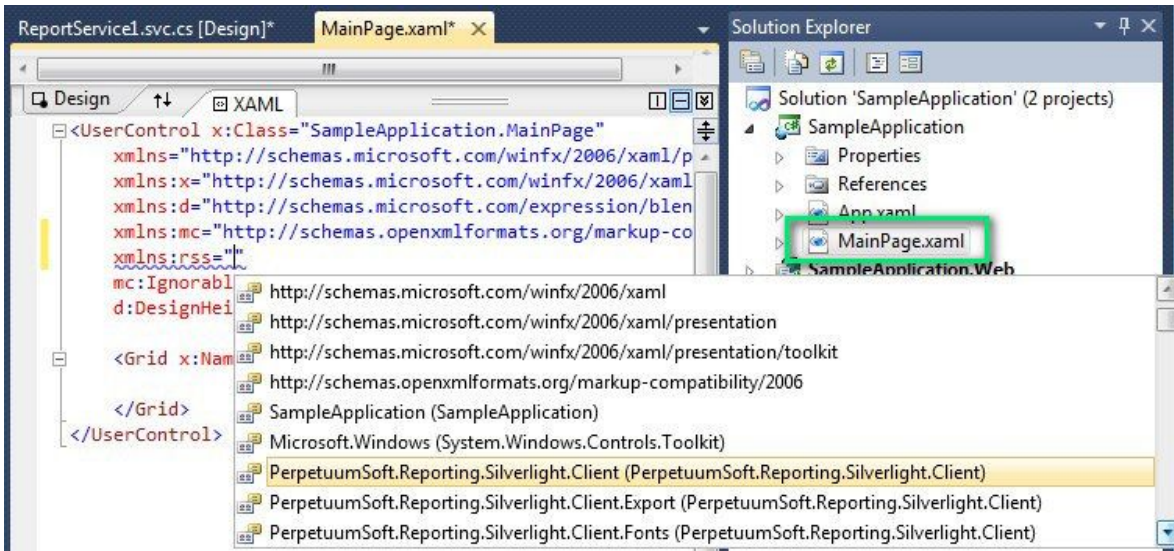


System.Windows.Controls.Toolkit.dll アセンブリの参照も追加します。



デザイナーで MainPage.xaml を開き、PerpetuumSoft.Reporting.Silverlight.Client アセンブリの "rss" xml 名前空間を追加します。





次に、Grid セクションに ReportViewer コンポーネントを追加します。

```
<rss:ReportViewer x:Name="ReportViewer1"
    ServiceUrl="http://localhost:5555/ReportService1.svc"
    ReportName="CustomersReport"/>
```

変更後の UserControl セクションは次のようになるはずです。

```
<UserControl x:Class="SilverlightApplication.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:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:rss="clr-namespace:PerpetuumSoft.Reporting.Silverlight.Client;assembly=PerpetuumSoft.Reporting.Silverlight.Client"
    mc:Ignorable="d"
    d:DesignHeight="300" d:DesignWidth="400">
    <Grid x:Name="LayoutRoot" Background="White">
        <rss:ReportViewer x:Name="ReportViewer1" ReportName="CustomersReport"
            ServiceUrl="http://localhost:5555/ReportService1.svc"/>
    </Grid>
</UserControl>
```

MainPage のソースコードを開いて、コード行を次のように変更/追加します。

[C# の場合]

```
public partial class MainPage : UserControl
{
    public MainPage()
    {
        InitializeComponent();
        Loaded += new RoutedEventHandler(MainPage_Loaded);
    }

    void MainPage_Loaded(object sender, RoutedEventArgs e)
    {
        ReportViewer1.RenderDocument();
    }
}
```

[VB.NET の場合]

```
Public Class MainPage
```



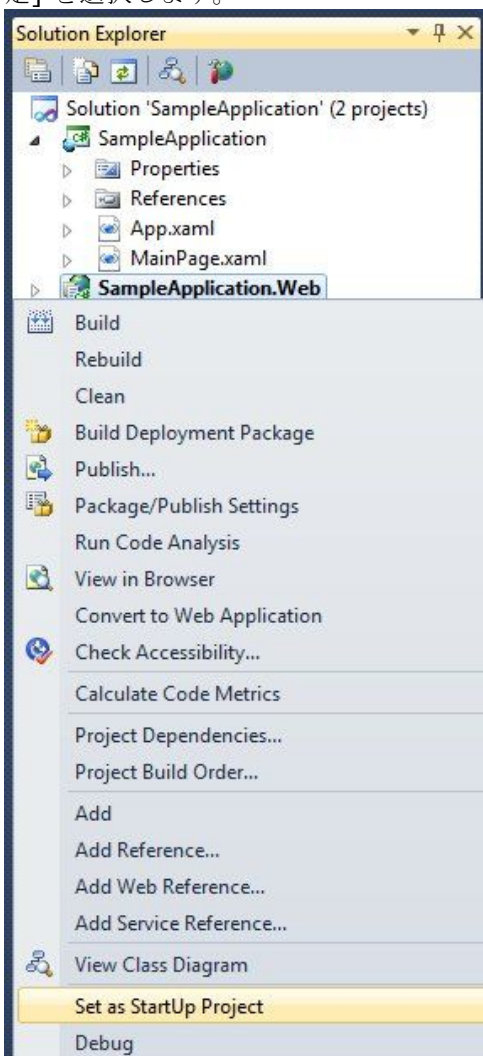
```
inherits UserControl  
Public Sub New()  
InitializeComponent()  
AddHandler Loaded, Address Of MainPage_Loaded  
End Sub  
Sub MainPage_Loaded(ByVal sender As Object, ByVal e As RoutedEventArgs)  
ReportViewer1.RenderDocument()  
End Sub  
End Class
```


備考 : RenderDocument() メソッドを呼び出すと、サーバーにある現在のレポートの生成し、レポートビューアに表示します。

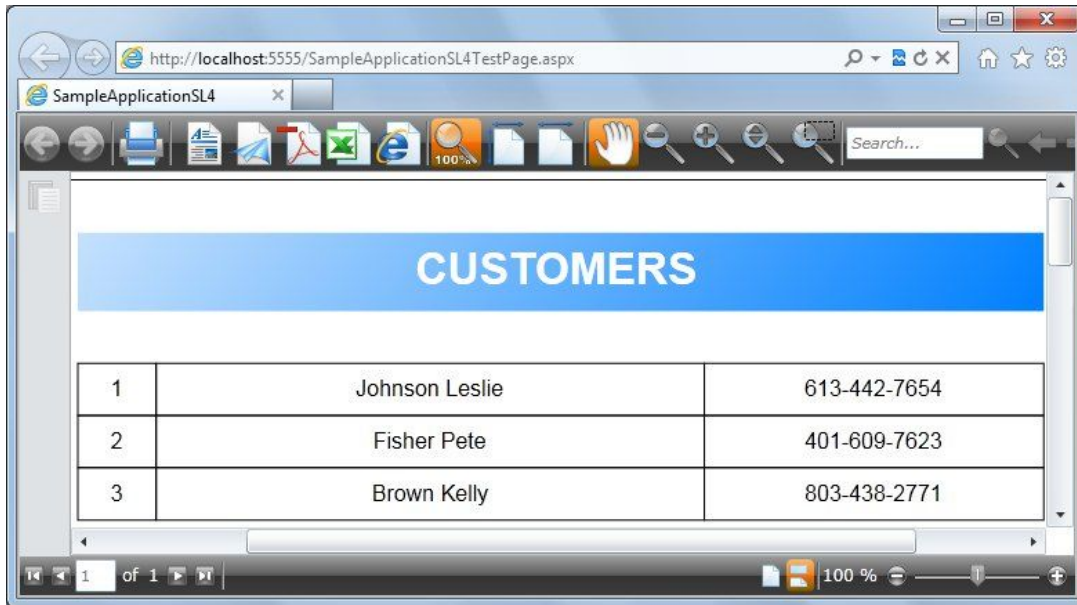
## 手順 6

アプリケーションの起動

SampleApplication.Web アプリケーションを [スタートアップ プロジェクト] に設定します。それには、SampleApplication.Web を右クリックし、ポップアップメニューから [スタートアップ プロジェクトに設定] を選択します。



Visual Studio のメインツールバーにある「デバッグ開始」  ボタンをクリックしてアプリケーションを実行します。



## 終わりに

基本的な手順を説明し、シンプルでとても操作可能なアプリケーションができ上がりました。コードをたくさん記述する必要はなく、既存の実装を使用しただけです。大半はそれで十分です。ご希望の動作がデフォルトで用意されているサンプルとはかなり異なる場合は、**SharpShooter Reports.Silverlight** の多くの動作面やレポートビューアの見た目も必要に応じて変更できます。