# SharpShooter Reports.WPF 基本的な使い方

Last modified on: November 15, 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
ライセンス認証	3
アクティベーション	5
開発	5
手順 1. アプリケーションの作成	5
手順 2.プロジェクトにレポートテンプレートを追加する	7
手順 3. 参照の追加	8
手順 4. XAML コードの追加	8
手順 5. データソースの作成と代入	9
手順 6. レポートテンプレートの編集	10
手順 7. レポートのデザイン	11
手順 8. レポート表示	12



## はじめに

このユーザーガイドには SharpShooter Reports.WPF を使った簡単なレポートテンプレートの作成方 法が含まれています。

このユーザーガイドは SharpShooter Reports.WPF ユーザーのために用意されたものです。

### システムの必要条件

SharpShooter Reports.WPF を使用するには以下が必要となります。

- .NET Framework 4.0
- Visual Studio 2010

### ライセンス認証

ライセンスを入手購入されましたら、[スタート]メニューからライセンス マネージャを起動してライセ ンスをインストールしてください。





ライセンスを追加するには、「ファイルから追加」ボタンをクリックし、ダウンロードした「\*. Elic」 ファイルを選択し、「閉じる」ボタンを押してライセンス マネージャを閉じます。

License manager			-	×
() porpotium	Product name	License type	Registered to	Subscription expiration date
Perpetuum	SharpShooter Reports.WPF	Registered	The Desire	
license manager				
B	Add from file Add license	Remove license Renew your license		Close

ライセンスがないと、**Perpetuum Software** コンポーネントの使用中に「体験版」を表すポップアップ ダイアログが表示されます。

PerpetuumSoft.Reporting.WPF.ReportViewer	
SharpShooter Reports.WPF : ReportViewer requires 'RSS.WPF' option in license	
You have 21 days trial left.	
Agreement	
End User License Agreement for SharpShooter Reports.WPF	
Perpetuum Software LLC SharpShooter.Reports.WPF SOFTWARE COMPONENT PRODUCT Copyright (C) 2011 Perpetuum Software LLC END-USER LICENSE AGREEMENT FOR SHARPSHOOTER.REPORTS.WPF SOFTWARE COMPONENT PRODUCT	
IMPORTANT - READ CAREFULLY: This Perpetuum Software LLC End-User License Agreement ("EULA") is a legal agreement between you, a developer of software applications ("Developer End User") and Perpetuum Software LLC ("Vendor") for SharpShooter.Reports.WPF™ SOFTWARE COMPONENT PRODUCT, its relevant controls, source code, demos, intermediate files, media, printed materials, and "online" or electronic documentation ("PRODUCT") contained in the installation file.	
By installing, copying, or otherwise using the PRODUCT, the Developer End User agrees to be bound by the terms of this EULA. The PRODUCT is in "use" on a computer when it is loaded into temporary memory (i.e. RAM) or installed into permanent memory (e.g. hard disk, CD-ROM, or other storage device) of that computer. If the Developer End User does not agree to any part of the terms of this EULA, THE DEVELOPER END USER CAN NOT INSTALL, USE, DISTRIBUTE, OR REPLICATE IN ANY MANNER. ANY PART, FILE OR PORTION OF THE PRODUCT, OR USE THIS	

ビューアを使用すると透かしが表示されます。



# アクティベーション

SharpShooter 製品を一度もお使いになったことがない場合は、最初の起動時に本ソフトウェアのアク ティベーションが必要となります。本製品のアクティベーションを行うには、表示されたウィンドウに お客様のお名前と Email をご入力ください。

ActivationForm	X
This is your first time running Sh continue. Please, enter your nam button.	arpShooter product, it should be activated before you may ne (optional), email adress (required) and press "Activate"
Name	
*Email	
	Activate Cancel

アクティベーションのフォームが入力されると本製品のアクティベーションは完了し、お使いいただけ るようになります。

×
Product was successfully activated. Thank you.
ОК

### 開発

### 手順1.アプリケーションの作成

**Visual Studio** を開き、(.NET Framework 4 を使用して)新規の WPF アプリケーション プロジェク トを作成します。



New Project						and the second s	? ×
Recent Templates		.NET Fra	mework 4 🔹 Sort by: Default	• • •	Search In	stalled Templates	٩
Installed Templates		plates		Visual C#	<b>Type:</b> Visual C#	/isual C#	
<ul> <li>Visual Basic</li> <li>Visual C#</li> </ul>				Visual C#	Window	vs Presentation Found	ation client
Windows		<b>S</b>	WPF Application	Visual C#	appricat		
Web ▷ Office		c.	Console Application	Visual C#	=		
Cloud		<u></u>	ASD NET Web Application	Vieual C#			
<ul> <li>SharePoint</li> </ul>		ĒĊ	Astract web Application	visual C#			
Silverlight Test		C#	Class Library	Visual C#			
WCF			ASP.NET MVC 2 Web Application	Visual C#			
Vorktiow ▶ Visual C++			ASP.NET MVC 3 Web Application	Visual C#			
<ul> <li>Visual F#</li> <li>Other Project Type</li> </ul>	s	EC <sup>8</sup>					
▶ Database		€ <mark>C</mark> #	Silverlight Application	Visual C#			
Modeling Projects     Test Projects		C#	Silverlight Class Library	Visual C#			
Online Templates		<b>c</b> ≇	Silverlight Business Application	Visual C#			
			WCF RIA Services Class Library	Visual C#	Ŧ		
Name:	WPFSample						
Location: c:\			\visual studio 2010\Projects	•	Browse.	•	
Solution name:	WPFSample				Create d	irectory for solution	
					Add to s	ource control	
						ОК	Cancel

クライアント プロファイルをデフォルトのフレームワークとしてお使いの場合は、プロジェクトのプロ パティでフルバージョンのフレームワークに変更してください。



WPFSample $\times$	MainWindow.xaml MainWindow.xaml.cs	
Application	Configuration: N/A    Platf	form: N/A
Build		
Build Events	Assembly name:	Default namespace:
	WPFSample	WPFSample
Debug	Target framework:	Output type:
Resources	.NET Framework 4 Client Profile	Windows Application 👻
Services	.NET Framework 2.0 .NET Framework 3.0 .NET Framework 3.5	Accurate
Settings	.NET Framework 3.5 Client Profile	Assembly Information
Reference Pa	ths Install other framework 4	
Signing		
Security	<ul> <li>Icon and manifest</li> <li>A manifest determines specific settings for an app</li> </ul>	lication. To embed a custom manifest, first add it to
Publish	your project and then select it from the list below. Icon:	
Code Analysi	s (Default Icon)	<b>II</b>
	Manifest:	
	Embed manifest with default settings	
	Resource file:	
		,

# 手順 2.プロジェクトにレポートテンプレートを追加する

プロジェクトに新しい項目「SharpShooter Reports Template」を追加します。



Add New Item - WPFSample			२ <mark>- </mark> २ -
Installed Templates	Sort by: Default	Sharp	Shooter ×
<ul> <li>✓ Visual C# Code Data General Web Windows Forms WPF Reporting RSS for SL SharpShooter Reports Silverlight SVforRS Workflow Search Results</li> <li>Online Templates</li> </ul>	SharpShooter Reports Template	Visual C#	e: Visual C#
Name: ReportTemplate	1.rst		
			Add Cancel

### 手順 3.参照の追加

プロジェクトに次のアセンブリ参照を追加します。

- PerpetuumSoft.Charts
- PerpetuumSoft.Framework
- PerpetuumSoft.Framework.Export
- PerpetuumSoft.Framework.Model
- PerpetuumSoft.Instrumentation
- PerpetuumSoft.Reporting
- PerpetuumSoft.Reporting.Export.Excel
- PerpetuumSoft.Reporting.Export.Html
- PerpetuumSoft.Reporting.Export.Pdf
- PerpetuumSoft.Reporting.Export.Rtf
- PerpetuumSoft.Reporting.Export.Xps
- PerpetuumSoft.Reporting.WPF
- PerpetuumSoft.Reporting.Xaml.Export
- PerpetuumSoft.Reporting.Export.OpenXML

### 手順 4. XAML コードの追加

MainWindow.xaml に(黄色で強調表示されている)次のコードを追加します。



```
<Window x:Class="WPFSample.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:reports="clr-
namespace:PerpetuumSoft.Reporting.WPF;assembly=PerpetuumSoft.Reporting.WPF"
        Title="ReportViewer" Height="auto" Width="auto">
    <Grid>
        <reports:ReportViewer x:Name="reportViewer">
            <reports:ReportViewer.Source>
                <reports:ReportSource x:Name="reportSource"</pre>
Source="c:\Projects\WPFSample\WPFSample\ReportTemplate1.rst">
                </reports:ReportSource>
            </reports:ReportViewer.Source>
        </reports:ReportViewer>
    </Grid>
</Window>
```

ファイルの絶対パスが Source として使用されます。対象のプロパティ フィールドからテンプレートフ ァイルのパスをコピーできます。

注意:ファイル参照を使用してテンプレートがデザインされるので、リソースアドレスを使用する場合、実行時にリソースの編集は行えません。

最終アプリケーションのリソースの参照を変更できます。更に、XML テンプレートビューをソースのレ ポートソースとして設定できます。

It is possible to change the reference for the resource in the final application. Besides that you can set the XML template view as Source ReportSource.

### 手順 5. データソースの作成と代入

I

ビジネスオブジェクト一覧をデータソースとして使用します。プロジェクトに新しい Employee クラス を追加してください。クラス構造は次の通りです。

```
internal class Employee
{
    public string FirstName { get; set; }
    public string LastName { get; set; }
    public string Title { get; set; }
    public string Phone { get; set; }
    public DateTime BirthDate { get; set; }
    public string Address { get; set; }
}
```

(例えば Loaded フォームのイベントハンドラで)製品のコレクションを代入し、reportSource にデ ータソースのコレクションを追加します。

Fill the collection of products and add the Data Source collection to reportSource (for example, in the Loaded form event handler)

```
List<Employee> employeeList = new List<Employee>();
employeeList.Add(new Employee()
```





```
{
      FirstName = "Maria",
       LastName = "Anders",
      Title = "Sales Representative",
      Phone = "(71)555-5598",
       BirthDate = new DateTime(1960, 5, 29),
      Address = "Obere str. 57"
});
employeeList.Add(new Employee()
{
      FirstName = "Ana",
       LastName = "Trujillo",
       Title = "Owner",
       Phone = "(5)555-4729"
      BirthDate = new DateTime(1971, 7, 1),
      Address = "Avda. de la Constitution 222"
});
employeeList.Add(new Employee()
{
       FirstName = "Antonio",
       LastName = "Moreno",
       Title = "Owner",
       Phone = "(5)555-3932",
       BirthDate = new DateTime(1969, 3, 12),
       Address = "Mataderos 2312"
});
reportSource.DataSources.Add("EmployeeList", employeeList);
```

reportViewer.RenderDocument();

#### 手順 6. レポートテンプレートの編集

レポートを編集する場合、デザイン時でも実行時でも行えます。実行時にレポートテンプレートを編集 する方が便利です。



備考:レポートを編集する場合、実行時にデータ構造やアプリケーションデータに アクセスできるので常に実行時の編集をお勧めいたします。

アプリケーションに、実行時のテンプレートの編集機能を追加します。フォームに「デザイナ」ボタン を貼り付け、このボタンのクリックイベントに次のハンドラを追加します。

```
private void designTemplate_Click(object sender, RoutedEventArgs e)
{
    reportSource.DesignTemplate();
    reportViewer.RenderDocument();
}
```

デザイン時にレポートテンプレートを編集したい場合は、ソリューション エクスプローラで対象の項目 をダブルクリックしてください。



注意:デザイン時にレポートをデザインする場合、レポートデザイナでデータソー スの取得はできません。

### 手順 7. レポートのデザイン

アプリケーションを実行し、「デザイナ」ボタンをクリックします。デザイナでプログラム言語 (CSharp、Visual Basic またはカスタム言語)を選択できます。Document オブジェクトの ScriptLanguage プロパティから言語を選択します。このサンプルでは CSharp を使用します。

レポートの作成処理を簡素化するために、このサンプルではレポートをテーブルウィザードで作成しま す。挿入タブの「テーブル」ボタンをクリックし、表示されたテーブルウィザードからデータソースを 選択します。

Table Wizard	x
Data source	
EmployeeList 🔹	
Available fields	Selected fields
FirstName	
LastName	
Title	
Phone	>
BirthDate	>>
Address	
	<
	Next > Cancel

FirstName、LastName、Phone、Address を選択し、「選択されたフィールド」に移動し、「次へ」 ボタンをクリックします。デフォルトのフィールド設定を使用し、「次へ」を押します。LastName と FirstName で並べ替えを指定し、「終了」をクリックします。



Table Wizard					23
Data source: EmployeeList					
Available fields		Sort by fields			
Title		Field name	Sort direction	ı	
Phone		LastName	Ascending		
BirthDate		FirstName	Ascending		
Address	>				
					<b>1</b>
					<b>I</b>
	<				
	<<				
		< Back	Finish	Cano	el

この後、必要に応じてテーブル(例:テキストの位置合わせ、タイトルのハイライトなど)を編集できます。

dataBandEmployeeList:DataBand DataSource =	= EmployeeList		
headerEmployeeList:Header			
FirstName	LastName	Phone	Address
detailEmployeeList:Detail			
⊡GetData	⊤GetData	TGetData	TGetData
("EmployeeList.FirstNa	("EmployeeList.LastNa	("EmployeeList.Phone"	("EmployeeList.Addres
end of dataBandEmployeeList			

### 手順 8. レポート表示

テンプレートを保存し、デザイナを閉じます。レポートビューアに作成されたレポートが表示されます。



e	95	i 📥 🚔 📩 🖡	¥) 🤌 🎧 🗋 🗈	[ 🕐 २ २ २ ६	Search	. <del>← →</del>
		FirstName	LastName	Phone	Address	
		Maria	Anders	(71)555-5598	Obere str. 57	
		Antonio	Moreno	(5)555-3932	Mataderos 2312	E
		Ana	Trujillo	(5)555-4729	Avda. de la Constitution 222	
	<					~
T T	1 of 1	<u>इ.स.</u>			 	

エンドユーザーにテンプレートの編集をさせたくない場合は、「デザイナ」ボタンを削除し、 ReportSource.Source をリソースのパスに変更できます(追加されたテンプレートはデフォルトでリ ソース型を持っています)。