**Software FX**

*Any Chart, Anywhere!*

# Chart FX Shines
# for Silverlight™ Today

# Introduction

Microsoft's Visual Studio 2008 introduces a whole new set of .NET technologies that will revolutionize, once again, the way you develop smart client and web -based applications; the most notable being LINQ, WPF and, of course, Silverlight.

For most of us working in and around web application development in the business world, Silverlight is a big step forward; especially if you consider that it provides clean coding practices with extensible languages (e.g. XAML and C#). It also provides some of Visual Studio's rich programming model that we have become accustomed to and can no longer live without; such as class libraries, debugging capabilities and IntelliSense among many others. Silverlight (especially from 1.1 onwards) can be used by .NET developers to create cross platform browser-based content with genuinely interactive client-side functionality that doesn't rely on a sticky tape of AJAX.

Microsoft has been working diligently with third party vendors, like Software FX, to open Silverlight to the vast goodwill that Microsoft partners have to offer. Unfortunately, much of this effort has not been exposed yet as most vendors continue their Silverlight and Visual Studio 2008 integration efforts. At Software FX, we couldn't wait to put our hands on Silverlight and try to build a Chart FX version that you could work with today. In the end, we were able to implement a Silverlight solution based on Chart FX 7 which will allow you to take advantage of tomorrow's technology with tools you are familiar with today. In addition, this paper will help you understand Silverlight from a control vendor perspective. It sheds light on some of the cool aspects of Silverlight as well as some of the integration limitations we found along the way.

We acknowledge that some of these limitations are temporary and will be quickly addressed by Microsoft in subsequent releases of Silverlight, Visual Studio and Blend. In the meantime, you can use this information as an integration guideline for Chart FX with Silverlight 1.1, Visual Studio 2008 Beta 2, and Microsoft Expression Blend 2 (September 2007 Preview).

# Silverlight or WPF?

WPF and Silverlight share some common themes and tools that blur the line between them. For example, Silverlight and WPF are great platforms if you need to incorporate video, or animated transitions between a sequence of images or objects; or if you need to bind to XML data. However, in reality, Silverlight is a WPF subset targeted at web developers building cross-browser, rich, web-based applications.

To put this into perspective, you could say WPF is to a Windows Forms application what Silverlight is to a .NET control functioning in a browser. This means, you should expect WPF to be able to harness the full power of your Windows desktop while Silverlight will allow you to provide cross-browser support but be limited by the security and other restrictions of any browser-based technology. For example, Silverlight runs in a sandbox, so among other things, it can't access the local file system; this means WPF is the better choice for building traditional document-handling applications. In addition, WPF has many hardware accelerated 3D features that are not supported in Silverlight.

Also, there are generational differences between WPF and Silverlight. For example, at the time of this writing, Silverlight had the following limitations:

• Silverlight provides layout support through Canvas only. In essence a Canvas defines the space that allows developers to create and move objects placed inside the application. A Canvas supports absolute positioning only (x, y, width and height). In contrast, WPF provides richer layout classes which, like StackPanel, DockPanel and Grids, support automatic resizing of elements contained within a panel.
• Silverlight does not support popup windows preventing the creation of skinned user interfaces or windows that could fall outside of the Silverlight container in the browser.
• Silverlight does not provide the wealth of user controls available in WPF projects.
• Silverlight does not support data-binding or control templating, features that are used extensively in Chart FX.

Admittedly, Microsoft is actively working on some of these limitations and will probably address them before this paper is actually published. However, these generational restrictions played an important role in the current Chart FX architecture for Silverlight. As explained before, our main goal was to provide developers with a truly rich data visualization experience for their Silverlight applications today.

## Microsoft Expression Blend 2

A quick look at Microsoft Expression Blend 2 and you will immediately realize the ultra cool stuff about Silverlight is not just under the hood in Visual Studio 2008. You will love the timeline, interactions, dynamics, and everything else you don't normally see when you look at code. For those of you who believe developers can't design, no more than designers can develop; Microsoft Expression Blend 2 will let your designers easily complement or build your application visuals without breaking your code... Really!

Most Silverlight detractors would argue that Microsoft's Expression products are lacking in comparison to other designer tools. While it is true Flash, Photoshop and Illustrator may be every designer's bread & butter; Silverlight is a platform for developing entire web-based applications, not just its graphics or animations. In this regards, Microsoft Expression products are not only welcomed but also powerful additions for web developers.

In any case, we not only believe this generational gap between Adobe and Microsoft Expression products will be closing in fast but it will also be complemented with a new set of smarter controls for Silverlight which will ultimately provide the killer functionality that both developers and designers need in their web-based applications.

As soon as Microsoft Expression Blend 2 allows third party control integration, we'll update this article with more information about the Chart FX Silverlight Add-On control and its design-time experience within Blend.

## The Chart FX Silverlight Add-On Foundation

Naturally, we started our Silverlight efforts by inspecting our near-release Chart FX for WPF and determined that we couldn't create our Silverlight offering based on it. Basically, Chart FX for WPF's complex visuals based on 3D, animations and complex layouts made it difficult to adapt it to the Silverlight feature set.

Our second option was to start developing a basic Silverlight charting client from scratch [like the ones you've starting to see coming from other third-party vendors]. However, we couldn't conciliate the fact that although this would ultimately serve as good demoware and a branding vehicle for Chart FX, a Silverlight client supporting basic pie and bar charts would not have been enough to deliver a real Silverlight business solution.

In essence, we wanted to provide Silverlight developers with a Chart FX version with the ability to provide all of the charting scenarios we current support including, but not limited to, Gauges, Maps, Statistical and Financial charts. Simply put, we wanted the entire Chart FX functionality to be available to Silverlight. Needless to say, duplicating such functionality in a dedicated offering for Silverlight would have been an insurmountable development effort.

In addition, if you consider charting is a process that is tightly coupled to the data extraction process, we needed to leverage the server architecture available to ASP.NET developers. You see, a charting tool must be very verbose in terms of properties and methods because developers always want to be able to tweak chart settings and visuals based on the data being plotted. A Silverlight client control with that many objects and properties would've unnecessarily complicated your Silverlight application where most of the client code would govern how the objects are laid out or how do they animate, or interact with each other; not how the chart parts or elements are set up.

Finally, as of today most developers are still using Visual Studio 2005 and ASP.NET as their main IDE and deployment platform, respectively. So, in the end, we decided to pro-vision Chart FX 7 with a XAML renderer that could be used natively in a Silverlight application. With this approach, the Chart FX 7 ASP.NET server controls enable you to harness the power of servers and leverage your existent ASP.NET infrastructure while providing your users with a Silverlight experience beyond what can be done with AJAX and DHTML alone.

The Chart FX Silverlight Add-On foundation is described in the following diagram:

# Building a Silverlight app with a Chart FX chart

The Chart FX Silverlight Add-On Renderer combines the most complete and robust charting functionality provided by Chart FX 7 through a client/server architecture with the visual richness of Microsoft Silverlight.

Before beginning, please be sure you have installed the Chart FX Silverlight Add-On Renderer. It was included with Chart FX 7 version 7.0.2845.*. A standalone installer is also available for customers with a previous release at http://www.softwarefx.com/Silverlight.

## Generating the chart on the server

The first step to using the Chart FX Silverlight Add-On Renderer is to create a new website using Visual Studio 2005, which you will then add a Chart FX 7 to create the server-side chart. Due to a security restriction in Silverlight, the server and client pieces must reside on the same domain, so you should select IIS as your web server (location must be HTTP):
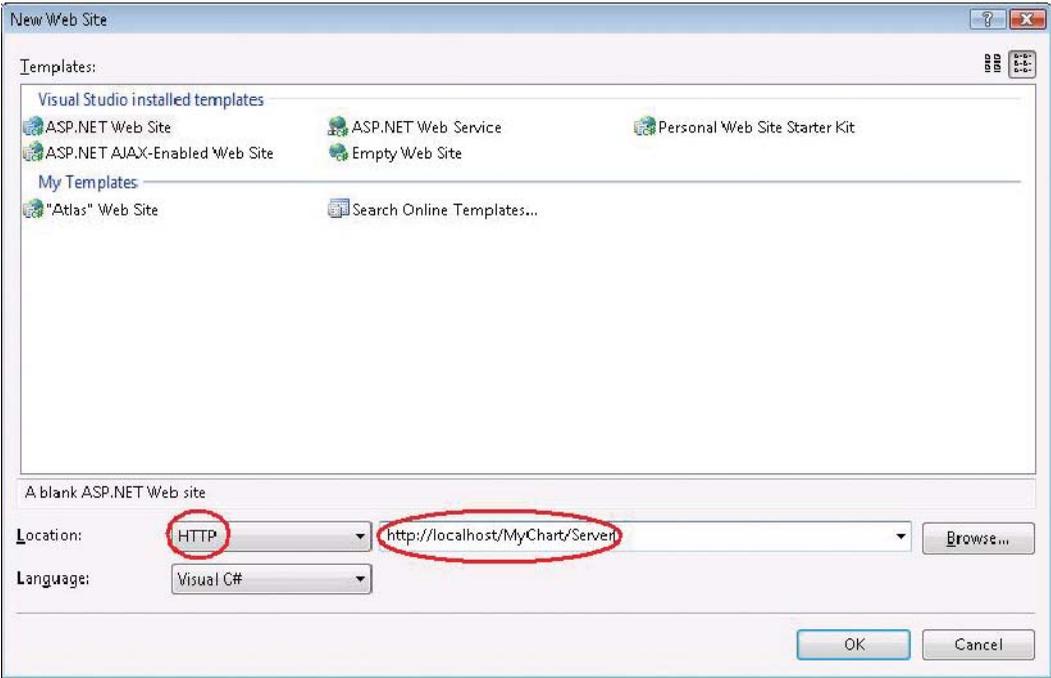
You can add as many .aspx pages as you want to the project. For simplicity, we will limit this part of the tutorial to just a single page with one chart. Begin by dropping a chart from the toolbox onto the web form and setting its desired size. Using the wizard, property grid and code, you can connect the chart to data and customize the gallery type, axes, titles, legends, etc. One crucial part is the selection of Silverlight as the chart renderer; this can be done through the smart tag wizard by selecting Silverlight from the list of available renderers, as shown below:
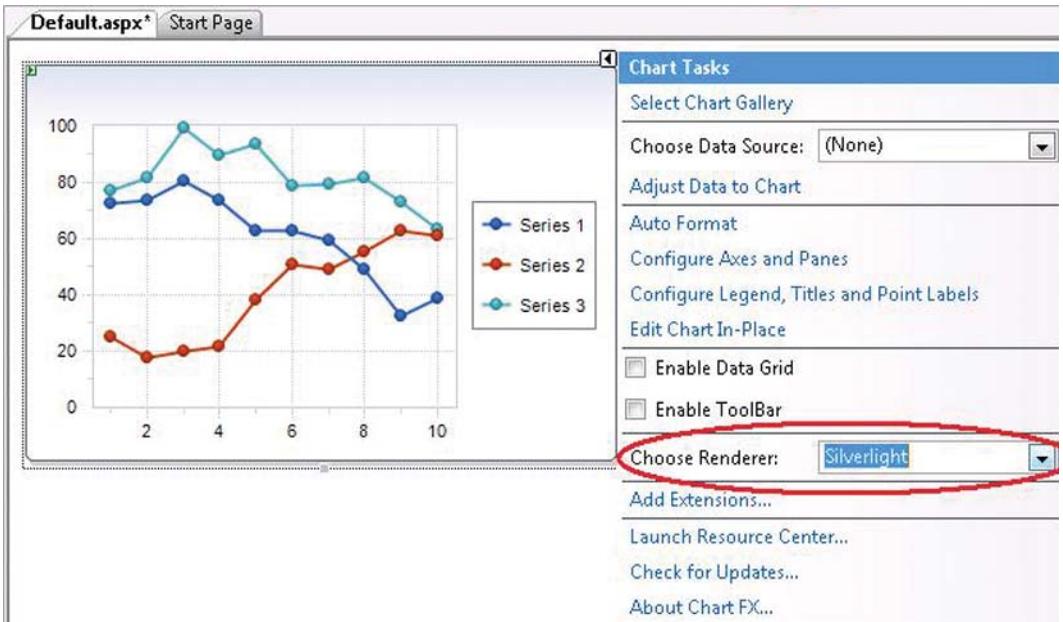


Although we recommend selecting the Silverlight renderer at design time for ease of use, you can also do it through code. To do so, you must first add a reference to the Chart FX Silverlight Add-On Writer to your project (ChartFX.WebForms.Writer.Silverlight.dll, available in the bin folder of your Chart FX installation directory). Then, you will need the following code:

```
Chart1.OutputWriter =
new ChartFX.WebForms.Writer.Silverlight.SilverlightWriter();
```

*Note: Please be aware that your page must contain only chart controls using the Silverlight renderer.*

If everything is setup correctly, running your page will show nothing on the browser, but if you inspect the underlying html code that is generated you will notice a portion of the page containing the chart in XAML code, including a container ‹Canvas› among many other Silverlight objects. At this time you are ready to setup the client portion of the solution.

## Including Chart FX in your Silverlight application

Chart FX Silverlight Add-On also includes a native Silverlight client component that you can use directly in your Silverlight applications. At the time this tutorial was created, Microsoft Silverlight 1.1 was in its alpha stage, and the design time support for Silverlight applications in Visual Studio 2008 and Microsoft Expression Blend 2 September Preview was minimal. In either case, you will need to do the following to include Chart FX Silverlight Add-On functionality in your application:

1) Add the ChartFX.Silverlight.Client.dll assembly to your project's references. This file is located in the bin folder of your Chart FX 7 installation directory.

2) Add the following entry to your main Canvas header:

```
xmlns:cfx="clr-namespace:ChartFX.Silverlight.Client;
assembly=ClientBin/ChartFX.Silverlight.Client.dll"
```

3) To add a chart anywhere in your XAML page you must add a line as follows:

```
<cfx:Chart x:Name="chart1"
Source="http://localhost/MyChart/Server/Default.aspx"
Width="375" Height="250" />
```

The Source property is the way you link the chart you have created using Chart FX 7 and the Silverlight client component. It must point to the URL you created in your web form in the first step. There are some advanced scenarios to minimize the impact on the server when you have multiple charts; for details, please refer to the API appendix.

Although the final size of the chart is determined by the size specified in the server page, we recommend setting the same width and height to the client component. While the control obtains the chart from the server, an animation indicating that the chart is being loaded appears; setting the same width and height on the client and server will ensure a smooth transition from the animation to the rendered chart.

*Note: For a complete list of members supported by the Silverlight client component refer to the API Appendix.*

The Chart FX Silverlight Add-On client component supports all of the capabilities of a standard Silverlight control, so you can apply any transformation you would apply to other controls, allowing for eye catching effects.

Important: When you run your Silverlight application, the browser will open the TestPage.html using the file system. Due to the security restriction mentioned earlier, the Silverlight application and chart source must reside on the same domain. For that reason, you will need to create an application in your local IIS pointing to the project folder and access it with your browser via http://localhost
(e.g. http://localhost/MySilverlightChart/TestPage.html).

## Integrating Chart FX Extensions in your Silverlight application

This approach will immediately make a wealth of Chart FX resources available to your Silverlight applications. For example, if you are in need of integrating maps to your Silverlight applications you can use the Chart FX Extensions Pack and generate not only XAML versions of all of the maps it provides but also write code behind them.

## Conclusion

Silverlight's greatest advantage may very well be the vast amount of synergistic third party products and Microsoft technologies that will allow developers to blend media with logic for advanced, enterprise-level, web-based applications. This will ultimately be the driving force behind Silverlight's adoption in the business world and Chart FX epitomizes what could be done with Silverlight in a real-world Silverlight application.

By providing integration into Visual Studio 2005 and generating XAML code from a fully functional and enterprise ASP.NET technology, Chart FX's underlying technical approach is fundamentally superior. It immediately leverages the vast data visualization features available in the Chart FX product line while still allowing Silverlight to display and manipulate charts natively.

Despite the current Silverlight and Visual Studio 2008 limitations and restrictions, the roadmap is clear and Silverlight is no longer a moving target. This means that it's only a matter of time before you see a wealth of Silverlight tools and controls coming from third party vendors which will ultimately help you save development time (and money) while building your mission critical Silverlight applications.

If you are considering developing Silverlight applications, consider the many data visualization opportunities Chart FX will open for you today! If you are already a Chart FX registered user you can simply download our latest service pack or if you are just starting with Silverlight and Chart FX, you can download a trial version at http://www.software-fx.com/Silverlight.