

Entire article, and much more, is available at
www.mknopf.com/topics/code-camp-tampa.html

SILVERLIGHT STRAIGHT TALK

By Michael Knopf

Programmer Analyst (A.K.A. I code)

Software Services for NASA at Kennedy Space Center

www.mknopf.com/topics/code-camp-tampa.html

GETTING STARTED

- Where to get the necessities
- Must Have tools
- Silverlight Toolkit, Contrib. Library, and Free components
- Straight talk about the awesomeness, disadvantages, and difficulties in Silverlight
 - The learning curve
 - XAML, it's a whole new world
 - Blend, it's a whole new tool even if you already know Adobe Photoshop
 - Visual Studio and Blend: it takes two to tango
 - Gotchas that will drive you nuts

GETTING THINGS DONE

➤ Understanding the basics

- UI layout
 - Grids
 - Stack panels
 - Canvases
 - Margins
- Animations
 - Visual State Manager
 - Animation using Blend
 - Behaviors
- Events and Data-Binding
 - Getting a grip on Asynchronous Data Exchange
 - Data-Binding and why it's so awesome in Silverlight
 - Accessing your web services: set-up to easily handle moving from Dev to Test and then into the Production Environment
 - Wiring events to User Controls

GETTING THINGS DONE, CONT.

- Accessing and Using Shared Components
 - Using Blend’s Control Library to access items in referenced libraries
 - Extending existing controls in order to customize them
- Testing and Debugging your application
 - Using Fiddler to intercept web service requests
 - Using Silverlight Spy to identify User Interface elements
 - Unit Testing your Web Services and Business Logic
- Getting Around: Showing/Hiding Silverlight screens
 - Using Delegates to control logic flow between screens
 - Using the new “Silverlight Menu Navigation” template
- Application State and Persistent Storage
 - How and when to use Isolated Storage
 - Silverlight Apps compared to ASP.NET Web Apps
 - Application State in Silverlight
 - Download-all-at-once (Silverlight) vs. Download-when-needed (asp.net)

SECURITY AND DEPLOYMENT

Security

- Forms Authentication
- Integrated Windows Authentication (IWA)
- Restricting Cross-Site Access to your web services
- Securing communication between your application and the web server
- **Guess What:** your ***ENTIRE Silverlight app***, with all its C# source code, is exposed to the world (so yes, anyone can steel that Bank application your building)

Deployment

- Setting up the server to support Silverlight
- Deploying your web services to IIS
- Deployment Scenarios
 - Hosting your services within the web application
 - Hosting your services on a separate DNS location

TOOLS OF THE TRADE

- Where to get the necessities
 - <http://silverlight.net/GetStarted/>
- Must Have tools
 - Visual Studio.NET 2008 or 2010 (available from the link above)
 - Expression Blend 3 (available from the link above)
 - [Silverlight Spy](#)
 - [Fiddler](#) Debugging Tool
 - [Firefox](#) with FireBug (Tools → Add-ons → Get Add-ons)

SILVERLIGHT TOOLKIT, CONTRIB. LIBRARY, AND FREE COMPONENTS

- [Silverlight 3 Toolkit](#) for Visual Studio
- [Silverlight Extensions](#) (view the [demos](#))
- [Vectorlight](#) Free Tools
- [DevExpress Silverlight Controls](#)
- [VisiFire Charting Tools](#)

STRAIGHT TALK

the awesomeness, disadvantages, and difficulties in Silverlight

- You , as a developer, now can and should learn how to developer professional grade User Interfaces that offer end users an engaging and user friendly experience. You will take on both the role of Designer and Developer.
- The things that you wished you could do in C# instead of JavaScript can now be done, this frees you from the limitations of the typical web application.
- The learning curve: its steep (but not mount Everest)
 - **XAML, it's a whole new world.**
 - **Its an unfamiliar landscape** that takes time getting used to (remember the first time you ever worked with HTML and CSS, its just like that)
 - **Its very picky and unforgiving**, one wrong move and you get an very unhelpful BAD PARSER error forcing you to literally dig through all of your code to try and figure out what the problem it.
 - **Blend, it's a whole new tool** (even if you already know Adobe Photoshop)
 - You have to find your way around the tool, its not obvious how to do a lot of things , expect to spend lots of time and energy getting used to it
- **Visual Studio and Blend:** it takes two to tango
 - You really must have Blend in order to develop complex (or even average) Silverlight applications
 - Animation and themes are pretty much done exclusively in Blend.
 - All your C# coding is done in Visual Studio
 - Switching between windows without saving first **WILL CAUSE YOUTO LOSE YOUR WORK**
- **Reliable Error message don't exist**, this increases the time and difficulty in tracking down issues.

STRAIGHT TALK, CONT.

Gotchas that will drive you nuts

- If you don't **explicitly set the Forecolor** (a.k.a the font color) on your TextBlock then your text wont show up (even though it looks fine in Blend)
- **Your project can get out of synch** when working in both Blend and Visual Studio, if your not careful you can end up undoing a lot of your code
- **The Design View in Blend can screw you!** Don't rely on it exclusively, instead always be looking at the Code Window (Split View) when working with the designer
- **No Project Refresh in Blend:** After adding items to your project through visual studio you have to trick Blend into reloading the project in order to work with the additions.
- **Bland fails to apply namespace properly** when using it to add new User Controls
- **Selecting an Image** source through the designer copies the image to the Silverlight Application Root
- **Blend doesn't recognize Web Service References** and will indicate that you have errors when you really don't
- **Error Message are worthless.**

GETTING THINGS DONE

Understanding the basics

➤ UI layout

- **Grids:** they are very much like HTML Tables
- **Stack panels:** this is like an HTML Div where you set the “orientation” of its contents to one of two setting, either Horizontal (default) or Vertical
- **Canvases:** just like it sounds, it’s basically a container where you position its content using `Canvas.Top=“x”` and `Canvas.Left=“x”`. Very similar to using a Top and Left in HTML/CSS
- **Margins:** allow you to add spacing between items (left, top, right, bottom) in order to control its position relative to other objects. Just like Margin/Padding in HTML/CSS

➤ Animations

- **Visual State Manager:** this is the build in “animation” that is applied to object. Is usually developed using Blend because it can be a nasty little bugger to work with by hand.
- **Animation** using Blend: similar to how you animate object in Flash (“Tween” like features as well as Easing ++)
- **Behaviors:** these are “In-Code” animations (basically taking an animation that Blend creates and writing it in C#). Can be reused through your project as well as other projects (by adding references to the DLL its contained within). Not easy to create because currently there isn’t any automation tools to write the C# code (you will be doing by hand, not fun).

➤ Events and Data-Binding

- **Getting a grip on Asynchronous Data Exchange:** timing is everything, your data may take a long time to return (if it returns at all) and you need to let people know what’s happening during this time. Data returned is not guaranteed to be returned on the User Interface Thread, meaning it may be there but not show up in the UI
- **Data-Binding and why it’s so awesome** in Silverlight: Binding objects to data *is INSANELY easy*. Objects look for data that it should bind to first in its own DataContext, then its Parent, then its parents parent, etc... until it finds (or doesn’t) what its looking for. Handles “not found” very graciously.
- **Accessing your web services:** set-up to easily handle moving from Dev to Test and then into the Production Environment
- **Wiring events to User Controls:** very similar to how you define it in ASP.NET web applications.

THANKS FOR YOUR TIME

Get the presentation:

- www.mknopf.com/topics/code-camp-tampa.html

Contact me:

- www.mknopf.com
- mike@mknopf.com
- michael.l.knopf@nasa.gov