



Hansel minutes

Hanselminutes is a weekly audio talk show with noted web developer and technologist Scott Hanselman and hosted by Carl Franklin. Scott discusses utilities and tools, gives practical how-to advice, and discusses ASP.NET or Windows issues and workarounds.

Text transcript of show # 63

May 3, 2007

Scott Guthrie and Jason Zander on Silverlight

Scott sits down with Scott Guthrie and Jason Zander at Mix 07 to talk about Silverlight, his incredible keynote demos, dynamic language support, and all of the rest of the mind-blowing news from Mix!

(Transcription services provided by [PWOP Productions](#))



Our Sponsors

 **telerik**
deliver more than expected

<http://www.telerik.com>

**NET** DEVELOPER'S
JOURNAL

<http://dotnet.con-sys.com>





Scott Guthrie and Jason Zander on Silverlight
May 3, 2007

Lawrence Ryan: From Hanselminutes.com, it's Hanselminutes, a weekly discussion with Web developer and technologist, Scott Hanselman, hosted by Carl Franklin. This is Lawrence Ryan, announcing Show #63, recorded Thursday, May 1st, 2007. Support for Hanselminutes is provided by telerik r.a.d. controls, the most comprehensive suite of components for Windows Forms and ASP.NET Web applications, online at www.telerik.com. Support is also provided by .NET Developer's Journal - The World's leading .NET developer magazine, online at www.sys-con.com. In this episode, Scott talks with Scott Guthrie and Jason Zander at MIX about Silverlight.

Scott Hanselman: Hi, this is Scott Hanselman and this is another episode of Hanselminutes. The podcast doesn't waste your time and we are over here at MIX in Las Vegas and I am fortunate enough to be sitting with Scott Guthrie and Jason Zander and Scott, we know that you control all things, you type USING and you hit control space, you're pretty are everything there. Jason you were at actually a dev in the core CLR team and now have risen to a powerbroker.

Jason Zander: Well, I was one of the original developers in the CLR, starting in 96, small set of us starting working out then. So, I am general manager for the CLR team Compact Framework, ASP.NET and things like that working for Scott.

Scott Hanselman: Oh, awesome. So, everyone is talking about the Silverlight surprise, that's someone actually called it the Silverlight surprise out in hallway, and it's Cross-platform CLR, Mac people are wondering around MIX, running .NET code on their Macs.

Scott Guthrie: This is little bit of a fun surprise to drop on people.

Scott Hanselman: I thought that people's brains were going to explode, it's pretty fantastic.

Scott Guthrie: It is definitely fun, you know it's fun when you are doing a keynote and people are wondering what is going to come next and we had a little bit of that yesterday.

Scott Hanselman: So, there is the Silverlight story, there is Beta, there's 1.0 Beta and there is 1.1 Alpha and all those bits are up at Microsoft.com/silverlight but the Cross-platform thing to make that happen, people were speculating tiny CLR, micro CLR, totally new written CLR but you guys have done something different?

Scott Guthrie: Yeah, what we have done is we are shipping out Silverlight 1.0, this summer and that's basically focused on Media scenarios and it only has a JavaScript programming model. We are going to go ahead and ship 1.1 later but the Alpha 1.1 is available for download now, one with Visual Studio tool support for it and basically in the 1.1 release, we are including a CLR engine and the CLR engine we are using is actually, is based on the same CodeBase for the full Desktop CLR. So, it's not a tiny CLR, it's not a Compact Framework version to CLR, it is actually the same CLR execution engine, that you are using with .NET today.

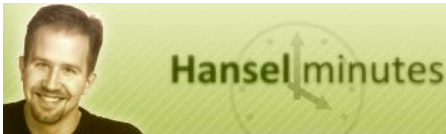
Scott Hanselman: The 2.0 CLR is, it's generics and collections and all that stuff.

Scott Guthrie: Generics as collections as the same Garbage Collector and has the same jitter and the same type system and without one of the benefits that means is you can actually compile an assembly and if you use the same Framework Libraries, that same file format and that same assembly can be used to open Desktop Engine, as well as open Silverlight.

Scott Hanselman: So, are we going to see prescriptive guidance coming about whether it's appropriate to start moving computational work that happened on the server side like validation and data transfer objects and different business objects and moving those over on to all those Desktop Pentiums that are currently hanging out and not doing any work?

Scott Guthrie: Yeah, certainly we are going to for that, you will see much more prescriptive guidance as to, how you use all the different app patterns that we have. I think, one of the important things that people should make sure, and understand is Silverlight in many ways, doesn't replace things, it's really enabling new scenarios that weren't possible today and so there is definitely, we still, -- you absolutely expect people to be building full Desktop Apps using WPF. Apps, you are going to building server side apps with ASP.NET, you know what we are trying to do with Silverlight, is enable you to run .NET in the browser, and go ahead and kind of merge, some of the attributes of both worlds and integrate nicely into an overall solution.

Scott Hanselman: Right, I mean certainly if folks have .NET on their desktop, they can exploit .NET, all they want. But in your demo, you took an XP Box and I think it was an XP Gold Disk it didn't have anything on it, you were running IE6,



Scott Guthrie and Jason Zander on Silverlight
May 3, 2007

it looked like it had just been booted up fresh, no .NET framework on it of any kind 1.0 at all. And you went click, click and suddenly, you were running .NET in the browser.

(00:05:02)

Scott Guthrie: Yeah, one of the things that we have done with Silverlight, and you'll see us take the same install technology to the full .NET framework going forward as well, but with Silverlight, one of the things that we did was, really tried to make the install experience lightening fast and so, in general once you have the bit download, it's about a ten second install experience on a clean machine and that does not require the .NET framework on machine, does not require any Codecs on the machine, it literally is as you take a fresh machine, in ten seconds you're good to go.

Scott Hanselman: It wasn't that whole of 3 minutes, what is MSI doing kind of thing -- it was actually so fast, I ran it three times because I didn't know that it did anything.

Scott Guthrie: Yeah, we heard that from a few people, yeah, somehow make it obvious, that like oh, it worked, because a lot of people keep saying like, okay, I guess it must have crashed because, 7 seconds and it's gone, it works...

Scott Hanselman: It takes longer to remove Shareware than it does to get this thing on your box, I mean it's just butter, it really installed like butter.

Scott Guthrie: And that's certainly kind of an experience that I think you know it's critical for the types of scenarios we are going after with Silverlight are, to really kind of enable, kind of Ubiquitous Web Apps on the web, to take advantage of this and then you have to have that kind of completely seamless install technology in order to make that happen.

Scott Hanselman: Now, one of things that you said a moment ago, was that right now in Silverlight 1.0 the kind of engine, the glue is the JavaScript inside the browser but outside Silverlight, we are going to enabling JavaScript inside Silverlight, inside the browsers, so you're going to have managed JavaScript like JScrip. What other dynamic languages are going to be in there?

Scott Guthrie: Yes, one of the things that we announced yesterday, is this new Framework Library called the Dynamic Language Runtime, and it's an assembly that provides a whole bunch

of infrastructure support for enabling dynamic languages and dramatically simplifies writing new languages on top of .NET and we will support that with Python, we have our IronPython compiler, that we've already shipped in the past, JavaScript, a dynamic version of VB and then one of the things we also announced yesterday was that we are supporting Ruby, through a new compiler that we are building called IronRuby.

Scott Hanselman: So, this means Ruby in the browser but also the DLR because it's not actually even though it's called a DLR at runtime, it doesn't modify or patch the CLR, like you said, it's just an assembly, I can reference it on the server side to start on Ruby, mix and match?

Scott Guthrie: Yeah, you can start using, Ruby on the server side, you can use Ruby inside WPF, and obviously you can use it inside Silverlight. So, from the language perspective, if you love Python, you love Ruby, you love C#, you love VB well, the .NET's the place to be.

Jason Zander: Yeah, one of the big things in that, we should say that's also available on CodePlex right now, you can download the SourceCode, but the big thing we are trying to do, with the Dynamic Language Runtime, was take some of the elements that came out, of the IronPython work that Jim Huginin had done and said, how can we, just like we did with the common language subset in the CLR originally, make it really easy for dynamic language author to go build new languages that can share and take advantage of that open functionality. And so, as soon as you sit on top of this, you get the Dynamic Dispatch and type interaction and things like that, which you otherwise have to go write yourself. And you might wind up with kind of a walled garden approach, where that dynamic language didn't understand this one, this kind of bridges that gap and then because it's all written with the same CLR underneath, it's the same file format, same IO, you can use the debuggers, the breakpoints, all of that kind of stuff just comes to you for free.

Scott Hanselman: So, this is going to be a playground for the guys that wrote Boo, particularly because I've looked at that -- if look at the -- if you take Boo from Codehaus, if you look at their stuff, which is Python, kind of wrist friendly, CLI language and if you look at their stuff in Reflector, they are doing a lot of compiler generation, and they are doing some kind of -- there are doing some funky stuff in there, they are going to be able to kind of throw all that away, because a lot of those services, run



Scott Guthrie and Jason Zander on Silverlight
May 3, 2007

Dynamic Dispatch going to be available to them in the DLR.

Scott Guthrie: Yeah, Oren's already...

Scott Hanselman: Oh he's...

Scott Guthrie: Oren takes a little bit and he's looking forward to playing with the DLR lot more. Yeah, certainly, one of the goals that we're going to have with the Dynamic Language Runtime is, to be able to do a talk at some point, future PDC or future lang.net conference, where you can actually start at the very beginning of the talk and sort of define a language on stage. And then you can actually code it and actually have it up and running, working inside...

Scott Hanselman: Silverlight?

Scott Guthrie: Having to work inside ASP.NET, and having to work inside WPF, or in Java...

Scott Hanselman: Well, I wonder how long it's going to take for somebody to hack together a PowerShell ASP.NET Dynamic Language thing.

Jason Zander: I, probably won't take too long actually ...

Scott Hanselman: Yeah, I suspect you probably won't take very long at all.

Jason Zander: But the key thing for us, again he thing we are trying to enable, why use the CLR? It's like well, you get to use RTC, RTC Time Compiler, it's world class, and now you get to essentially use Jim Hogan and his code and John Lamb and all the code he is contributing and just bring them on top of yours. So, it makes it really, really, simple.

Scott Hanselman: So, in the past when I have tried to do kind of a poor man's code generation, or poor man's dynamism in within the CLR, I do code dump stuff and I usually end up leaking assemblies, I can't get assemblies kicked out and stuff and then with .NET 2.0, there is introduction to the dynamic method.

Jason Zander: Yeah, it's Lightweight Code Gen, LCG for short.

Scott Hanselman: So, you are basically kind of creating programmer intent, rapping it around this handle and calling it a dynamic method and then get Garbage Collected just like any other object.

(00:10:06)

Jason Zander: That's right, it's kind like I have this piece of IL and I would like to trade that for a function pointer police and which is really nice and a kind of malleable dynamic environment, where you're defining, redefining methods etcetera, that was actually a feature that was put into version 2.0 with the CLR way back when, in order to support dynamic languages and so then Jim was able to come along, with Python 1.0, you know, you leverage that in a much better code and that's essentially, what you get because, if you just try and do this work yourself, it can be hard to figure what the tricks are, how to make it perform and things like that and that's what the DLR does for you, just does a lot of heavy lifting, you just look up to it.

Scott Hanselman: Yeah, Oren's going to love that and the Goo guys. So, when you are doing a lot dynamic code generation like that, how do you balance that with security and safety, I know there's probably a new security model within Silverlight but also just the regular CLR and you have got IL being jitted left and right here.

Jason Zander: Yeah, but the important thing there is you are still using IL and you are still using the type-safe subsystem inside of the CLR, so we are still going to run the verifier, when appropriate and based on unusual security settings and things like that, and so you get all those same kind of benefits, right that is the advantage of having the Managed Code System, there for you, and as you mentioned in Silverlight, there is actually a new pretty simple straight forward Sandbox Security Model, we will bringing that forward on to the Desktop CLR as well. So that your apps, again when you write it for Silverlight and you decide at some point, wow I'd really love to add 3D and full WPF, you can quite easily just take that code and move it forward. So there has been a lot of innovation around security in this release, it's something we haven't even talked about too much yet but probably be doing follow ups and blogs, and stuff over the next few weeks.

Scott Hanselman: Very cool. Do you see that there are lot of people, who've done a lot of crazy stuff in JavaScript, when I say crazy stuff, I don't mean that it's necessarily bad or pejorative but they have taken JavaScript and just squeezed it so much, that they've built entire engines, I am not talking specifically about AJAX but whole object models and realizations in the browser. How compliant is the ECMAScript support that's going to get jitted v/s the stuff on the outside, such that someone could take good JavaScript and an engine on the client side and put that in and suddenly get a thousand X speed up.



Scott Guthrie and Jason Zander on Silverlight
May 3, 2007

Jason Zander: A couple of different things, you will be able to go ahead and use our JavaScript compiler, right. I think, the big key thing for developers, we are going to have to sort through is, how compatible can we make that code, and we often do it. Obviously, the one nice thing about JavaScript, it's been in IE, Firefox etcetera, for a long time, it's kind of standardized, etcetera and anytime you come along with the new code system, the language, that's usually the trickiest part to get that there. So, we will have the performance as you mentioned with the compiler code on top of it, but we are going to work through the guidance and we won't recommend it saying, hey just go drop this thing in, replace everything, we are going to have a more gradual kind of set of guidance...

Scott Hanselman: As a programmer, yeah.

Jason Zander: Yeah, yeah, exactly.

Scott Hanselman: Now we know that the JScript.dll and it's history, it's been around for a very long time and there is a lot of people that feel that JScript because of the way it was architected and some of the COM background around the scripting languages and the Plugins and how it's reused with classic ASP, it's a little slow, particularly in its interactions with the DOM. How are Silverlight -- and Silverlight is now, going to be allowed to reach out of itself and into the DOM as well as JavaScript code, in the browser outside of Silverlight reaching back in. So, there is this kind of seamless bridge between two type systems. How does that work and how does that perform it?

Scott Guthrie: Now, we basically, basically there's two features that call out - one is we do have a managed DOM API now the chips for Silverlight, and so you can go ahead and use any .NET language inside Silverlight, you can use C# you can use VB and use Python, Ruby and JavaScript and from within .NET code, you can go ahead and just say, HtmlPage, HTMLDocument and you have the reference to the HTML document that the Silverlight control has been hosted on, and so you can create elements and you can get access to elements, its DOM API and to get element by ID and get a reference to it. You can also -- and this is kind of really cool, if you for example, reach out and grab a button, a reference to a button, you can also attach a managed event to that button and so you can reach any HTML element and wire up any event that typically you do in JavaScript and wire up a .NET delegate to it. And so it you want

to, you really have the complete capability to interact with HTML in a rich way.

Jason Zander: That's hot.

Scott Guthrie: Yeah it's pretty cool, and it works in cross-platform, so it's just not a naïve thing that works in Firefox, it works in Safari.

Scott Hanselman: So then, there is a dispatch mechanism within some of the agHost JavaScript, then it must be watching.

Scott Guthrie: It's not an agHost but it's...

Scott Hanselman: Event bubbling or something?

Scott Guthrie: It's in the Plugin.

Scott Hanselman: The Plugin is watching, what's going on?

Scott Guthrie: Yeah.

Scott Hanselman: Oh, I didn't realize they had that kind of power. Plugins can do a lot.

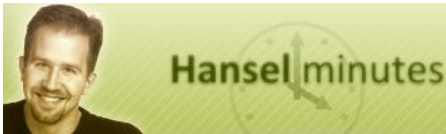
Scott Guthrie: Yeah Plugins can do a lot.

Scott Hanselman: How can they go full screen, I thought that was kind of creepy?

(00:14:48)

Scott Guthrie: Yeah, it was all -- basically, Plugins do have the ability to go full screen, so one of the demos that we showed, a couple of demos we showed, was the ability for people to effectively go completely full screen and do some pretty neat interactions as a result. And one of the nice things about the full screen is, you can go ahead and not only run video in snippets but you can actually run the video over the entire thing. So, you can actually do full screen video, with Silverlight and even cooler in some of the things that the Major League baseball folks showed on Monday, is you can actually overlay text as well as controls and rich UI, any UI you want, go WPF on top of the video, while you're running full screen. And that just really, lets you build some mind-boggling UI experiences.

Scott Hanselman: I noticed when they did that, they would go to full screen and they'd look at the video and they'd go back in the -- it wouldn't actually restart like with Windows Media Player, where it's like, oh I have gone full screen, and we are going to start all over again, and going to go seek.



Scott Guthrie and Jason Zander on Silverlight
May 3, 2007

Scott Guthrie: Yeah, so we don't have to restart and you can even go ahead and actually resize the video dynamically. So, it's not just full screen or minimized you can actually say, like I am going to expand the video by 5% every second, and the video stream does not stop, it does not pause, it does not restart, you get a completely smooth experience, and that's so we can cross all platforms and all browsers.

Scott Hanselman: And back to the JavaScript kind of CLR bridge and this seamlessness that's going on, could that -- I know that every technology, every new technology gives someone potentially more rope to hang themselves with but people are doing a lot of threadings and more and more in the client and now you've got two different threading models, the JavaScript outside and the JavaScript on the inside. They could both potentially listen to events, be poking at buttons, one could probably get themselves into a bit of a problem.

Scott Guthrie: Well, from a threading perspective, you are actually handling all the events on the same UI Thread. And so the way browser works, is if you have two buttons with two event handlers, you can't run into a situation where you click both of them real quickly and both events fire, you are actually executing on the same threads.

Scott Hanselman: So, there aren't two UI Threads?

Scott Guthrie: No, yeah there is only one UI thread to browsers, now what Silverlight will enable you to do is to create background threads, and so if you want to be able to use something computationally intensive in the background, there will be a model for you to go ahead and do that within Silverlight and then, you can basically post an event that the UI thread, can then go ahead and handle and so once you've finished with that you can go ahead and update the UI.

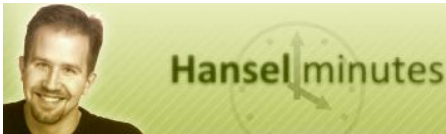
Scott Hanselman: Can poorly, but I'm a little confused, can poorly behaved JavaScript in the browser outside Silverlight affect the UI thread enough, that the Silverlight will have trouble doing its video. There is the browser WinFrame, there is the client area that's rendered and then there is kind of the execution thread of the browser page itself.

Scott Guthrie: Theory, if you have like an influent loop in your JavaScript that could affect obviously...

Scott Hanselman: It had nothing to do with Silverlight, I am just thinking more about just generally because the potential not only for using the client for more serious computations now, becomes kind of pretty significant, but also the ability for concurrency, to be doing one thing, while doing another, I mean, we were watching the Top Banana demo during the keynote and he had about nine videos going simultaneously, while moving a scrollbar over on the left, and I am trying to think about so how much concurrency and status, you're having to manage, and how much of this is just being handled by the intrinsic media functionality built in Silverlight. It appears that playing video, it's trivial, it's just a handle, you just say, hey play it. You don't need to worry about anything. It manages state.

Scott Guthrie: Yeah, built into WPF and then also the WPF on Silverlight, as a media element control and basically with a media element control, you can go ahead and point to kind of any type of media file including a WMV Video File and there is an object model for playing, pausing, stopping and seeking for that and there is Top Banana App, which you can actually go to www.silverlight.net and I believe we have a video up and just that a nice high rise video if you want to watch it, but it's pretty stunning and the nice thing is all of the actual graphics performances uncaps later inside that Media Element Control. And so, yeah, all he did is app and say, play, play, play on the nine of them and Silverlight handles the rest.

Jason Zander: I think, from using the concurrency on the client, you have the nice thing with System.Threading so it's the same API, you are already familiar with. I think, the very common usages are going to be, if you are going to do intense numerical crunching, big algorithm analytics, you know take your pick, and you'll pick off a thread to do that. Another interesting one will be things like, hey, I'd like to go and make a web service call and it may have a lot latency associated with it it may take a long time to stream in. So, this gives me an opportunity to kick off that background thread, pulling in results, processing them and kind of queuing them up for the UI piece. That's the sort of thing, I think you will wind up using in. People do try and do that today, with JavaScript with the wacky combination of timers and the sort of things. I think, that gets to, it's hard to write that kind of code, it's hard to debug it and get it right. So, having that powerful just threading model with .NET, there for you, works great and then the media controls encapsulates all the other hard work, so that too is very powerful.



(00:19:45)

Scott Hanselman: And you bring up a good point, that you are saying that there is the threading classes that we are used to, it may be not be the BCL System. Threading, that exact same code but it's behaving the same, it's the same public interface. It may have Silverlight specific things but just as I -- there is System.IO.IsolatedStorage, is that the same Isolated Storage, it's probably no, it's quite Silverlight specific for the security model.

Jason Zander: In general, what we did is we went through again looked at the scenarios, came from the core up, decided what the main bar was. In a lot of cases we actually just factored the existing code, that's the case, for the Core CLR itself and it's the case for the bunch of the framework libraries as well. In other cases we said what's unique about being in the browser? I have XML HTTP, I have other kind of services for the browser and so we took some of the same API contracts, gives you the same behavior like you mention. We may have plumbed some of those in through the environment that we live in. So, there is a little mixture of both and in every case, it was what's the right thing for what the developer wants to accomplish in this environment and that's how we did it.

Scott Hanselman: So, I don't know if this is an appropriate analogy but tell me if my thinking is right, for a while I would write code on 1.0, 1.1 and then make sure they compiled and worked on both and then there was the time when I'd write code on 1.1 and 2.0 and maybe mono and you want to just basically pick the intersection of the base class libraries, that same thinking that we could do and it's totally appropriate, even encouraged to write assemblies and take that exact same DLL and use it on the server side, use it on the client side, using that subset of intersected Base Class Library, functionality.

Jason Zander: Yeah, I mean what we try to do in Silverlight is, I think if you look at the number of classes we have, I haven't counted them but it's probably 1200 or something. We have some pretty rich libraries including for example LINQ. So, if you have some business logic and you have some collections of objects that are generic collections and you've got computation or you got some validation logic, or anything like that, you know you can express that using LINQ, it's some of the functional programming constructs that it provides, you do all the querying, you can do all the data modeling and yeah, that assembly will work in any .NET environment.

Scott Hanselman: So many people out in the hallway now that mostly from my point of view, the developers, that are thinking about, is this going to be, or is this, whatever they are thinking about, the way to do something like one, guy said well, we'll take our HTML forms and we'll squirrel them off into a Silverlight, that would be like a one pixel by one pixel Silverlight, it won't have any UI, will load up the object and then send it across via web services. And this is how they are going to exploit their web services stuff.

Scott Guthrie: I mean, certainly you could do that...

Scott Hanselman: You can do a lot of things.

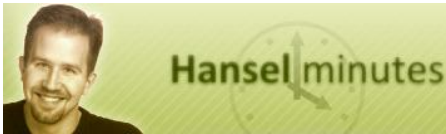
Scott Guthrie: You can do a lot of things. So, you can absolutely have a HTML page that has no WPF UI on it at all and use both the HTML and DOM API inside Silverlight, the other thing that's kind of cool though is, Silverlight allows you to expose public static methods, on your root container class, and you can then go ahead and decorate those methods and say, they are script Serializable and that means that you can actually from JavaScript in the browser, so again using the native JavaScript that shifts with the browser, you can actually construct JavaScript objects and call those public static methods that might be written in C# and will actually Serialize the JavaScript to and from those methods.

Scott Hanselman: So, this is like a Method Missing for JavaScript?

Scott Guthrie: Yeah, you know JavaScript is a little advanced.

Scott Hanselman: But it's got it's whole link bound thing I keep putting things in Ruby terms but I just decided, I want a function be called foo, I am going to call it from JavaScript and when I say FOO, it's happening in managed code.

Scott Guthrie: Correct and if you pass in for example couple of strings or an imager or an array of objects, we can actually serialize back and forth and we were actually reusing the same serialization technology that we used whether it's been AJAX on the server or it's part of our Jason Marshaling Code but the cool thing is we are actually Marshaling not -- we are Marshaling to and from native .NET types. It provides another easy way, if you want to go ahead and you have some computational intensive in the browser, you can actually -- yeah if you want to, you can actually host a Silverlight controller that has no UI



Scott Guthrie and Jason Zander on Silverlight
May 3, 2007

and just call into it and use the CLR engine to basically process it.

Scott Hanselman: It seems like, you can scale up your use of Silverlight, I mean if using the example you just gave is a really simple one, maybe I am a scientist and I got something really hardcore going on and I want to just add a computational engine and do some distributive computing, suddenly Silverlight without UI is doing distributive computing, then I decide to put some visualizations in and maybe I go to a 100%, by 100%, Silverlight fills the screen and I am kind of living the whole rich media life style and took all the way, up and down that whole.

Scott Guthrie: I mean, the WPF UI Framework that will ship in Silverlight will be a rich framework. Today, in the alphabets, it is very much focused on visualization and some of the core primitives that we need in order to build higher level controls. We don't have many higher level controls yet. You will see those controls start to appear and now we have the core bases there and you will see constructs like the WPF layout managers, Data Binding, eventually DataGrid and those types of controls appear. Although they are not all there yet, but it does provide a really good foundation today as with today's build as well as a Control Encapsulation Model Subset, you can build those types of components.

(00:25:05)

You know, I definitely reckon people do take a look at WPF as because it is a very exciting technology. One thing, going back to your threading comments, in terms of how the Top Banana Video App was able to display things as were stunning and as in real time as it looked, is that WPF is what we called retained mode graphics engine, which means, that when you actually paint something on the screen, you don't have to manually write any code to repaint it. WPF, and the whole model does that for you automatic, which is different than say, Win32 or using TDI, which made it much more harder to do that. The other thing that we've done with our WPF implementation in Silverlight is, it is very much optimized to take advantage of as many cores or processors you have on the system. And so, if have two cores, going forward pretty much every machine will have two cores or more cores, you automatically get a really rich UI framework, that is designed to maximize and take advantage of that. And that is also some of the reasons why you can build such stunning graphics using Silverlight.

Scott Hanselman: Is that usage of multiple cores, is that a WPF aspect of Silverlight or that's a new thing in the core CLR?

Scott Guthrie: Well, the CLR does take advantage of multiple cores with things like Garbage Collection and things like that, but from the graphics perspective and from a UI framework perspective, that is certainly something we are doing in the Silverlight and the full WPF going forward. And there is a number of, -- that's a good blogpost I should do -- but there is a number of things that we are doing across .NET, with the UI layer but also with things like LINQ that by going to a mode, where you are not actually telling us what to do, you are actually describing what you've like to have done. So, XAML you don't say, paint this pixel, you say, I want a rectangle here in this dimension. And with LINQ, you don't say don't do a four loop, I say, I want to have something that meets this criteria. By doing more declarative intent, the CLR and .NET and higher level frameworks will be able to go ahead and optimize appropriately across multiple cores.

Scott Hanselman: Does Silverlight act any different on Vista with the Desktop Compositing Engine, because it's painting, it's constrained to what the browser can do, right? So I think, I'd ask this question to you casually in the hall, the idea that, if you use the accessibility magnifier and you hover over a WPF app like the New York Times Reader, you get beautiful anti-alias stuff but if the WPF engine in Silverlight is different and living within the browser, if you take the Vista magnifier over that, you probably get rasters, you probably get bitmaps.

Scott Guthrie: Yeah, I have not had a chance to try it out, so I am not sure.

Scott Hanselman: Yeah, I have got a Vista Box lying around, you know what I am asking about though, don't you Jason?

Jason Zander: Yeah, I understand exactly what you are asking about.

Scott Hanselman: I mean, it's not particularly interesting, but it's not a big deal, but it sure is an interesting question because it tells you how, it's an experiment right, because I don't have the code, so as a scientist, I have to use a scientific method to glean what I think, is going on inside.

Jason Zander: Yeah, but the key thing with WPF on the desktop especially with Vista, Vista introduced a new driver model and it made it much easier to do this kind of cooperative



Scott Guthrie and Jason Zander on Silverlight
May 3, 2007

graphics processing on the machine and so you get full hardware acceleration and other things like that. And so, that's -- one of the events that you are going to get with the desktop version of WPF. In this case, Silverlight is optimized for cross plat, giving you reach, etcetera, but it shares a lot of the same graphics techniques underneath of that.

Scott Hanselman: So, that's very, very good information because, you are starting to give me and hopefully other folks enough information to decide, what tool is right for the job, like I have always wondered about the XBAP, the XAML Browser Based Applications and I have always thought it was kind of this redheaded stepchild but you just right then, made me realize that, if I was doing a browser based 3D accelerated application that did protein folding or something and I wanted it to be and I didn't want it to be a one touch deploy jump out of the browser, I wanted a Silverlight like experience and a very minimal installation then an XBAP would be totally appropriate for that, and we would get all the hardware acceleration and all of the full WPF desktop experience inside the browser.

Jason Zander: Certainly, yeah the way we have been thinking about Silverlight, this is I think, an important point for people to understand is, that all the API's and the features in Silverlight are absolutely designed to be cross-platform and we won't be adding in features to Silverlight that we can't consistently expose on all platforms. So, developers can know hey, if I write to the subset, it will work in all browsers and all platforms. There will always be richer features that we can use that are platform specific like 3D hardware acceleration, like richer database support and things like that and those, obviously the .NET framework and the Windows API and things like XBAP's and full WPF Applications, the client will be able to take advantage of and certainly, we see that there are different types of apps that people built, sometimes they want it to be as ubiquitous as possible, Silverlight is the way to go, for plain apps.

(00:29:58)

And sometimes, you want to take as much maximum reach of the operating system you are running on, and the system whether it is office integration, or it's 3D, or it's offline support, and there is whole bunch of other good reasons there and there are obviously the full .NET framework with WPF, either running in XBAP mode or outside the browser as is absolutely the right thing to do.

Scott Hanselman: Cool, and people are stoked, peoples' brains are exploding I mean, one guy blogged he's like, well I have learning to do. He is okay, he's been kind of sitting around just coding C# for a few years, like, oh that's the biggest thing since .NET 1.0 as far as I am concerned.

Scott Guthrie: We are pretty excited about it, it's been, we have been -- Jason and I have been working on this for a year now, and it's just great to finally be able to ship the bits and be able to talk about to talk about it and see people's reaction because people were just stunned yesterday and it's always fun to have that reaction.

Scott Hanselman: Yeah the future is bright. I really appreciate you guys taking the time out of your schedule. Thanks a lot for talking to us here on Hanselminutes.

Scott Guthrie: Thanks for having us.

Jason Zander: Thank you.