



Hanselminutes

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 #178

September 3, 2009

Optimizing Your Website with Jeff Atwood and Stackoverflow

Mark Miller thinks and talks fast. Fortunately he codes fast also. He works on CodeRush for DevExpress, a very intense Visual Studio plugin that helps you visualize and refactor code. How is it built? How does it break the rules? Scott digs in.

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



Our Sponsors

 **telerik**
deliver more than expected
<http://www.telerik.com>

 **nsoftware**
<http://www.nsoftware.com>

NET 
DEVELOPER'S JOURNAL
<http://dotnet.sys-con.com>





Lawrence Ryan: From hanselminutes.com, it's Hanselminutes, a weekly discussion with web developer and technologist, Scott Hanselman. This is Lawrence Ryan, announcing show #178, recorded live Thursday, September 3, 2009. Support for Hanselminutes is provided by Telerik RadControls, the most comprehensive suite of components for Windows Forms and ASP.NET web applications, online at www.telerik.com. In this episode, Scott talks about the Visual Studio plugin CodeRush with Mark Miller.

Scott Hanselman: Hi, this is Scott Hanselman and this is another episode of Hanselminutes and I'm sitting down today with Mark Miller from Developer Express, the famous architect of CodeRush and Refactor! How's it going Mark?

Mark Miller: It's going well Scott. How are you doing?

Scott Hanselman: I'm not doing too bad.

Mark Miller: Good.

Scott Hanselman: I wanted to talk to you because I got an interesting email from a guy who's trying to decide whether he would use CodeRush or a tool called ReSharper and both of these tools plug into Visual Studio and change it in some fundamental way. Could you maybe start by explaining what your tool does and your competitor ReSharper is a similar thing and the reason I bring up ReSharper is that CodeRush is kind of my favorite and kind of my favorite little tool like this but I'm interested in how you write something like this. So let's talk about, first, what it does and then I want to get into some technical details about what exactly you had to do to make it happen.

Mark Miller: Okay. Well, the essence behind CodeRush is to help you write code, see the code and navigate through the code faster. So it's essentially visualization, code creation and navigation are the main areas that it helps you with and anything that kind of falls into those categories including kind of, I guess what I would call outside of the box ideas in terms of how we can make people write code faster are certainly considered and if they are useful and functional and efficient they're put into the product. So the product kind of encapsulates those things. To give you some more specific examples, part of the product is refactoring tools. So CodeRush includes Refactor Pro which has about 185 refactorings in it that cover a lot of, all the languages we support, C++, C#, Visual Basic, Java Script, ASP.NET. So you've got refactorings in all those areas and more XAML, too I should mention XAML as well and then in addition to the refactorings you've got code declaration tools. So if you need to, if you're doing like TDD development or consume first development where you write the consumption code first then

based on clues at the consumption code, we can then make the declarations for you. A simple example is, if you have like `myinstance.mynewproperty = false`, you can declare `mynewproperty` as a Boolean type there, it's a simple example but of course can get very, very complex and sophisticated where the inference in terms of the type based on the assignment is evaluated while you're working and writing the code.

Scott Hanselman: Okay.

Mark Miller: So...

Scott Hanselman: Let me interrupt your advertisement because I don't want to talk about, that's a lot of stuff, so your tool does everything, it's like a Swiss army knife but it's, to be clear to people who maybe having used tools like this before, it's living inside Visual Studio?

Mark Miller: Right.

Scott Hanselman: This whole thing is in process, there's no little CodeRush.exe that pops out right?

Mark Miller: Right and to make it, make things more intelligent like for example if I have an assignment to something I need to actually evaluate that assignment at design time essentially, while you're writing the code not at runtime and so we have to have a sophisticated ability to do that and it's essentially the same as what IntelliSense uses inside of Visual Studio to be able to infer types and things like that. So there's a lot going on in the background while you're working and yes, it is inside of Visual Studio, it's, go ahead...

Scott Hanselman: Where do you hook in? I mean this has been working since 2003 or 2005?

Mark Miller: Actually since 2002, we've actually...

Scott Hanselman: 2002?

Mark Miller: We shifted in 2003 but it was working for 2002 when we shipped it in 2003.

Scott Hanselman: Okay.

Mark Miller: So since the very first version of Visual Studio for .NET and it hooks in through the Add-In technology and the VSIP package technology and...

Scott Hanselman: VSIP is Visual Studio Integration Partner, Integration something?

Mark Miller: Yeah, I think that's right, it's Integration Program or Partners or something like that, I think that's what it is.



Scott Hanselman: Okay.

Mark Miller: Yeah and basically it's a sophisticated way to hook into Visual Studio and extend it. So that's the first entry point and then there's a number of other points where we connect in the versions of Visual Studio up to 2008 because we do some things that VSIP does not allow like painting on the editor, for example. So when you, for example, do a refactoring like Inline Temp where it takes a temporary variable and inlines it, you'll see us paint on the editor, we'll cross out the declaration for the temporary variable...

Scott Hanselman: Okay.

Mark Miller: And give you a little indication that that's going to be moving down.

Scott Hanselman: Okay, so I want to dig into this, forgive me a little bit as I steer you around because I knew you've got, if you haven't met Mark Miller, he has a massive head and a huge amount of information inside said head and it can be very tricky because the man has a huge amount of information that wants to get out very quickly.

Mark Miller: Yeah...

Scott Hanselman: We know each other well enough, I think he'll forgive me if I steer you a little.

Mark Miller: No, I usually jump in the middle anyway and yeah.

Scott Hanselman: So I want to understand, I've written little Add-Ins before and by little Add-Ins I mean, I've added a menu, I've added a button and things like that and this is all using the older COM-based stuff, I think it was called IExtensibility2 or something like that, is that right?

Mark Miller: EnvDTE is I think what I'm familiar with, the Environment DTE...

Scott Hanselman: Oh, Environment DTE.

Mark Miller: Yeah, that's the object you work with.

Scott Hanselman: EnvDTE.

Mark Miller: Right, yes, yup.

Scott Hanselman: Wow, DTE, that's really old.

Mark Miller: Yeah. I think it's Design Time Environment, maybe I'm not sure even what that...

Scott Hanselman: Yeah.

Mark Miller: I don't remember what it stands for.

Scott Hanselman: So you've got, you get plugin at a very, very low level and you kind of get your hooks into Visual Studio not at the level that I was doing where it's like add a toolbar but it's something more fundamental about, you're really start messing around with a lot of stuff. You have to, like you said, paint to the...

Mark Miller: Right.

Scott Hanselman: Do they want you painting on text editor or...

Mark Miller: Yeah.

Scott Hanselman: Is there an official way to do that or you're just doing it?

Mark Miller: Well, there's not an official way to do it but they do want us to do it because we add value to Visual Studio. Well, I say they want us to, I think maybe unofficially they want us to do that...

Scott Hanselman: Okay.

Mark Miller: Because it adds value, it adds excitement, people are like, "Hey, can I do that too?" and it actually, I think, significantly influenced the direction of Visual Studio 2010 based on what we could do on 2008, what were able to do. So yes, we hook in using the officially sanctioned mechanisms. You use the term mess around and I would replace that word with carefully constructing a safety net, that's would what I would replace that word with...

Scott Hanselman: Well, that's a good point because you're doing things that are really out there in the sense of animations, painting on a scrolling background that you really have no business painting on because they don't give you a fight, you said no official hooks to say...

Mark Miller: Right.

Scott Hanselman: Paint the pretty arrow over this XML comment.

Mark Miller: Right. So to make all this work, there are a lot of things going on in the background that compose the safety net that I'm talking about.

Scott Hanselman: Yes.

Mark Miller: So for example in, up to 2008 we've got subclassing going on. So we subclass the window, we see when Windows messages are coming in and based on what messages come



through, that actually goes in and influences the safety net in terms of what we're showing you.

Scott Hanselman: Yeah.

Mark Miller: So the other things is just figuring out the positions of all the characters on the screen, for example in that example I mentioned earlier where I talked about Inline Temp and we do a strike through, we cross out the declaration. Well, we have to know exactly where that declaration is on the screen in order to get that, nail that correctly and there are subtleties, there's actually a lot of subtleties in terms of getting that calculation right because the font can change and other factors can change as well that change line height and character position.

Scott Hanselman: Right.

Mark Miller: So yeah, there was a lot of challenging stuff that we had to write. I remember, these were some of the first stuff we did Scott, as I recall, it took us like a good three weeks of solid development time before we got to a point where we felt like we accurately knew where all the characters were on screen.

Scott Hanselman: So let me interrupt for a second because this is a significant thing, if you're not familiar with CodeRush or tools like it as you can probably figure out by now, it really integrates with the IDE and does a lot of stuff such that it's kind of fundamental to the experience. One of the things that I like about CodeRush and I say this as a, just so the listeners understand, I've been a fanboy and a friend of yours for a million years and I've been, I tell people about CodeRush but you don't have any, CodeRush has no relationship with the podcast, you guys don't give me a sack of money, actually you have never given me anything ever. You've actually taken things from me Mark, so we have no relationship, I'm not trying to write the word...

Mark Miller: Right. I let you write a segue once and that was about it, that was about it.

Scott Hanselman: That was about it, yeah.

Mark Miller: Yeah, so...

Scott Hanselman: I think you got me a Diet Coke one time.

Mark Miller: Did I? That was at a conference from a free vending...

Scott Hanselman: It's a free diet Coke.

Mark Miller: Oh, yeah, the free one, yeah.

Scott Hanselman: The point I'm making is that I'm a fan of CodeRush no more, I just want to make sure

that you understand that I'm talking about this because I think it's interesting but what's interesting about CodeRush is that there's a lot of what people will call eye candy and I call it good user interface and user experience design. One of the things that CodeRush, one of the examples that I want to understand and maybe you can walk me through this is that, you've got a lot of these refactorings...

Mark Miller: Right.

Scott Hanselman: One of them is reorder parameters.

Mark Miller: Sure.

Scott Hanselman: So when I want to blow somebody's mind with CodeRush, I go over to a function and I've got a `intA stringY dateZ...`

Mark Miller: As the parameters, right?

Scott Hanselman: As parameters to the method...

Mark Miller: Yeah.

Scott Hanselman: And I put the cursor over and I hit, what I call the refactor button...

Mark Miller: Yup.

Scott Hanselman: I think it's `tilde+A` on my machine because it's a button I never push.

Mark Miller: Right.

Scott Hanselman: Then, suddenly, those three parameters get highlighted and I can use the left arrow and the right arrow to move them, to reorder them...

Mark Miller: Right.

Scott Hanselman: So that itself is interesting...

Mark Miller: Actually only one of those parameters is highlighted...

Scott Hanselman: Oh...

Mark Miller: But you get arrows showing you which way you can move it...

Scott Hanselman: Right, they go.

Mark Miller: And then the arrow keys and they move, right.

Scott Hanselman: So then I can move left and right and those parameters will move so that I can go `X-Y-Z, Z-Y-X`?



Mark Miller: Right.

Scott Hanselman: Yay! Well, we're on an audio podcast, so you can't see this but literally if I hit the right arrow, the intX will jump up and make a little froggy hop over to where stringY is and then stringY will rush away, while A is simultaneously jumping over, so it's this completely unexpected experience...

Mark Miller: Right. This animation moves them, you see them actually move, the text actually moves smoothly as they exchange positions.

Scott Hanselman: Right and for old-timers on the show, it always made me think of the CD ROMS of the early, early 90's where it was like a missed or these kinds of things where it was like you'd have a plain background, which in this case is the text editor, and then you'd have a video maybe inside of a picture frame on the wall or whatever and they were using this tricks where the video was seamlessly integrated...

Mark Miller: Oh, I see.

Scott Hanselman: Or at least as best they could.

Mark Miller: Right.

Scott Hanselman: You know what I'm talking?

Mark Miller: Yeah.

Scott Hanselman: It was like Quicktime pops out in a 320 x 240 whole and they try to make it look like...

Mark Miller: Sure.

Scott Hanselman: The whole screen is active, but it's not, it's just the tiny area.

Mark Miller: Yeah but it's only that little area, to some degree it's similar trick that what were doing there. We're painting directly on the window right on that GDI canvass and we are doing that at, we're animating so we're doing it, updating it pretty frequently but we're also taking care of a lot of other things. For example, if you have word wrapping turned on, we need to know where that's going to go right, and that might be an option like for example in VB, having word wrapping on is not that uncommon of an option to have so we have to support word wrapping and we also have to support what happens if you're changing positions with another parameter that's off screen right, because if it's off screen we need to actually scroll horizontally so you can now see the new position. So it's a combination Scott of that kind of painting trick but also being aware of where everything is and scrolling if necessary to make that happen and I want to throw in why we do this right, because it's like...

Scott Hanselman: Right, right.

Mark Miller: It's this kind of, whenever I see eye candy, my first thought is I consider it as possibilities that really, do we do just really make eye candy or is there some functional purpose behind it and usually I get past the first question pretty quickly with, "Oh no, there is a functional purpose" and in this case there is. If we take away that animation and we move instantly, it's hard to, see what's just happened.

Scott Hanselman: Right, right.

Mark Miller: And unless you totally are expecting it, right? If you say, "Oh, I know it's going to happen and I hit the key" and instantly it jumps then you're maybe okay and comfortable with that but if we show you the animation, then it totally confirms visually what just happened. In fact, if you look at new UI's that are coming out, like for example I like looking at the iPhone UI a lot and they've got animation all over the place there. If delete for example a picture, I actually see the picture kind of get sucked down into the garbage can right, for example on the iPhone. So it's this, you know somebody worked on this animation and the reason they did it is because it serves a purpose, it's not simply eye candy, you get a confirmation that the piece was deleted and you feel good about it because you actually saw it...

Scott Hanselman: Right.

Mark Miller: Go down into the trash can just like it happens in real life when you throw stuff away.

Scott Hanselman: Yeah, that's a very interesting idea. What would happen if when I was throwing trash away in real life, I picked up my arm and I threw the trash and then the paper disappeared from my hand and into the trash, it might take me a second to go, "Wait a second, where did the paper, oh, it's in the trash."

Mark Miller: Right. In fact if you and here's the thing, now we do actually support, like see for example you have many parameters, you want to take the first one and move it to the end, let's say you hit that arrow key 1,2,3,4,5 really fast like that, we actually do jump it as fast as you hit it but if you would just hit it once and you don't hit the key again...

Scott Hanselman: Oh, I didn't know you did that, you're saying that if you detect that I'm hitting the key really fast, you'll go, "Oh, he's in a hurry, I'll turn off the animation for now."

Mark Miller: Yeah. It took us a while to figure this out and to get it to a point where it felt perfect for everybody...



Scott Hanselman: Oh, that's cool.

Mark Miller: And this was one of these last, kind of the last things we added is when we realized that some people knew in advance they wanted to move it like three or four over, they hit it and they were waiting and so we took that away so they didn't have to wait and getting that balance right so that there was still animation it made sense and no animation at all, if it didn't make sense was a little bit tricky but I think we've nailed it on there. Speaking of UI in the code, my favorite is for rename and it's a real simple refactoring but what's cool about it is you get this inline experience right, you said I want to rename this and you can select a portion of the identifier and hit rename and then that piece remains selected and you just type in the new replacement for it and it renames all the corresponding references to that identifier everywhere throughout the code and it's very fast as well. So even if you've got an identifier that's showing up in tens of thousands of places inside of your source code, you still have this instantaneous experience where you are able to edit and rename right they're live in the code, the commit is also something we had to work out, when do we commit to this? How do we know you've committed because normally when you bring up a dialogue we know you commit by hitting OK.

Scott Hanselman: Hey everybody, this is Scott coming at you from another place in time. No doubt you've probably bumped into testing tasks now and then in your work and you know writing functional tests is probably not your favorite thing, it's kind of difficult, it takes time and the results can be dubious. Well, get ready to start liking tests, thanks to Telerik with the new WebAii Testing Framework, building web automation tests is a breeze. We've got code automation with advanced ASP.NET Ajax and Silverlight applications. You can write a single test, have it executed against multiple browsers at once. You benefit from a rich API, there's LINQ support, integration with Visual Studio Unit Testing also NUnit, xUnit and MbUnit, not to mention the free wrappers for Telerik RADControls for ASP.NET Ajax and Silverlight all shipping with Telerik's new testing tool. One of the best features, the WebAii Testing Framework, which was developed by ArtofTest is it's absolutely free. Before they got hooked on WebAii testing framework, start using it right away, go to telerik.com for more info, thanks a lot.

Right, let me interrupt because this thing the very thing that's significant about not just CodeRush as a product but your style and I think it's a different style than typical is that if one was going to, if I was going to sit down today and Scott Hanselman is going to write a refactoring tool, I'd hook up some hot key, I'd push a button to do a rename and I'd probably pop a dialogue box...

Mark Miller: Right.

Scott Hanselman: And I would tell them the name of the text underneath the cursor and I would and they would spend this time in this modal dialogue...

Mark Miller: Right.

Scott Hanselman: By modal, that means that they're stuck, it's basically forcing their attention.

Mark Miller: And it changes the view, right? Instead of looking at the code, you're looking at a picture, a representation of the thing that you want to change.

Scott Hanselman: Right.

Mark Miller: Same thing with the reorder parameters and then most people bring up a dialogue with a list of parameters and buttons that you have to press up and down right. So you're reaching for the mouse to click on those buttons because they usually don't have hot keys on them...

Scott Hanselman: Right.

Mark Miller: And it's a total, from my perspective, it's a total interruption in the flow and you're right, my style is like no modal dialogues, I want to get all the UI flat down into the source code...

Scott Hanselman: Right.

Mark Miller: So if there's opportunity, if there's an interactive point, if a refactoring requires interactivity, we do everything we can to make that interactivity at the level of the code that you're looking at so you're right there working in it and...

Scott Hanselman: So you never leave the editor?

Mark Miller: That's our goal right, yeah, to...

Scott Hanselman: But you are like, that's insane. Well, I mean not that you're, well, you're insane. I mean that's insane, the amount of work that you must have done to get...

Mark Miller: Right?

Scott Hanselman: Painting on the edit, all those things that you were describing that you needed to know about word wrap and...

Mark Miller: Yeah.

Scott Hanselman: What font they're using, if you did that wrong and I'm sure at some point it didn't work while you're in the process of doing the coding...

Mark Miller: Right.



Scott Hanselman: You must have been off by a few pixels or you have visual glitches, it must have been very irritating, right?

Mark Miller: I think a lot of people on the team thought that I was insane when I was telling them how we were going to do rename and there was a fair amount of push back but because I was leading the team, it was ineffective. They couldn't stop it from happening, I said, "No, we're going to do it." And then afterwards when they started seeing the reaction and they started kind of understanding what was going on, then it's like there's no more push back on anything right, but I think it's unusual even for the team to have that kind of push back right, but that that was definitely, the case with reorder parameters because I was saying, "No we have to animate, we have to move it, we needed to show people that it's moving." But it is a kind of a unique style but the payoff is there from I think a customer satisfaction stand point, if you use this and try it and then you go try it the old way, the modal way or something without animation, you definitely realize something is missing and when you realize that, you're thinking about, you're not thinking about coding when you're using that editor interface, you're thinking about why it's not as good as it could be.

Scott Hanselman: Right, right.

Mark Miller: So yeah, it is insane and it does take a lot more effort but the one thing though that we've done Scott is for example, like to do those strikethroughs, our strikethroughs, I usually take a lot of pride in the strikethroughs. If you look closely, it actually looks like somebody took a pen and drew across the code and actually was lifting up the pen, a felt tip pen, as they get to the end, it actually gets thinner and slightly translucent and it curves.

Scott Hanselman: It looks like someone actually took a pen on your screen and then struck out like an editor.

Mark Miller: Right.

Scott Hanselman: Like a red pen that someone would use when you wrote up a paper for English class.

Mark Miller: Right but strikethrough...

Scott Hanselman: It doesn't just look like a line...

Mark Miller: Yeah.

Scott Hanselman: It looks like a felt pen.

Mark Miller: Yeah, exactly and the thing is though strikethrough is a very common kind of preview mark up thing that we do and so we've got,

consequently we've got methods that come in our eventargs so we've got like, when we write a refactoring one of our events is prepare preview and the preview is what shows, comes up when you're hovering over a menu item or you've got it selected with the arrow keys, you're about to hit enter on that menu item for the refactoring, for example rename or reorder parameters or something like that and what we'll do is in that event handler for refactoring that prepare preview, we have an opportunity to actually paint on the editor. Well, the eventargs, has got something that says "Addstrikethrough" and we simply just give it a range of code and we come in and give it multiple lines of code and that's all we need to do, and it'll draw the strikethrough in there intelligently. It won't, it will only draw over code, it will ignore whitespace and so it's a simple one call kind of thing and it's very fast and easy to do. So if you want to write your own refactoring, you handle this event and you make that, call right in the eventargs, it's eventargs.addstrikethrough and that's it, you're done. So even though, yes there's been a lot of work and kind of insane amount of preparation almost to make this so that we can do these beautiful things on the editor, this eye candy or ease of use, whatever you want to call it, we've made it so that our developers and people who extend it can access that easily. So now we're at a point where we easily, it's just we have nothing but a lot to gain from all that work.

Scott Hanselman: Is this because, is this a matter of like, what is it they say? I think this is an old joke that any technology, any sufficiently advanced technology is indistinguishable from magic?

Mark Miller: Right, yeah.

Scott Hanselman: That's one thing.

Mark Miller: Sure.

Scott Hanselman: The other one was that any problem in computer science can be solved by one additional layer of indirection, therefore any sufficiently advanced technology required several layers of interaction. I mean you've really layered this thing such that this COM interface and the low level GDI stuff is...

Richard Campbell: Yes.

Scott Hanselman: Is a distant memory.

Mark Miller: It's far away, it really is. If you were have to ask me to sit me down and have me do a quiz on this, I would probably fail it spectacularly.

Scott Hanselman: So you wrote that engine, like we were talking about where letters are, where things wrap that was figured out with magic and buried long ago?



Mark Miller: Yeah, I think so, to some degree it is although that code has been resurrected now as we move to 2010 and because we're no longer in GDI world, we're in WPF world and...

Scott Hanselman: Well let's just take a moment, I just want to make sure that when you use a three-letter acronym or TLA that people know what it means. So GDI is that Graphics Device Interface, that's the way that we've been painting in Windows since time began...

Mark Miller: Right.

Scott Hanselman: And then WPF is the Windows Presentation Foundation and the thing that I think you're alluding to is that the editor and most of Visual Studio changed from GDI, the 25 year old technology to WPF, a newer technology...

Mark Miller: Right.

Scott Hanselman: And the entire editor swapped out from underneath you.

Mark Miller: Yeah, it did. Well kind of, I mean they have some of the old editor is still there but we kind of consider that dangerous to work with in 2010. They've provided shims and stuff and things like that but our goal is to go all with the new interfaces for 2010 but we also have the goal of making it so that our plugins simply needs to be recompiled and without any changes to them and for the most part I think all of our refactoring plugins will meet that criteria, we'll simply just need to rebuild them and they'll work.

Scott Hanselman: So they will need to be rebuilt?

Mark Miller: Yes but what's cool about that is those plugins are making calls, right, to like add strikethrough, they're not directly painting on the editor, they don't really have a canvass that they're working with or a graphics object, I'm sorry they don't have a graphics object that they're working with, they simply make that call through the eventargs right, to add strikethrough or to add insert point or draw arrow, that sort of thing, that's all done in the refactoring and so we've already kind of, even in the refactorings, we kind of decoupled away GDI to some degree right already in there and then our goal is to do the same with everything else. There are some plugins that actually do work with a graphics object in GDI and so those plugins will have to be changed a bit but what we're doing actually is we are, actually something kind of interesting is we're making it, we're creating an abstraction layer over both WPF and GDI so that all of our plugins can ultimately be in one source code base and then depending on what engine is installed it will either work in 2010 or 2008 and below and that's very cool from a plugin writer standpoint because you write

it once and then it works on all the others. It also has the possibility, this is not any kind of promise but it's certainly technically possible that if we can move CodeRush to a completely different platform that had a different kind of drawing mechanism on it that wasn't GDI, that wasn't WPF and it would still work.

Scott Hanselman: So you're saying that really, when a plugin writer, CodeRush plugin writer writes something, the surface area they sit on is entirely CodeRush, you don't let Visual Studio leak out at all?

Mark Miller: Well they can get to it. Like they ask for something that we don't support, that is supported through like EnvDTE or VSIP, we say go, go around us but for the most part they really don't need it. In many many cases we're better too like for example one if the things that just came to mind is the way we represent source code to plugin writers is a really neat way, we give them a tree and they can modify that tree, that parse tree and then give the tree back and then we can generate code with that. Whereas with Visual Studio it's two different mechanisms, if you want to generate code you use one mechanism, if you want to understand code it's a different mechanism plus there's no, just to date, there's not been agreement from the Visual Studio, the different language teams at Visual Studio, there's not been agreement over how to universally represent source code and so what happens is you get to a point where everybody's represented the same way, you have like classes and then you have methods and your parameters and then after that we stop. There's no inside of methods that's available if you're just going to use the extensibility mechanisms that come out of Microsoft and that's one of the cool things about the DX core which both CodeRush and Refactor are built on and the DX core is this free extensibility engine that we also, the extensibility engine we use and we make free for other developers to download and use, as well. The DX core gives you this really sweet universal way of modifying, working with code and understanding it and maybe modifying it as well and the cool thing about this Scott is that all of our refactorings are written without any checks to see what language we're writing, we're working in. So there's no check to see, are we in Visual Basic, are we in C#? We just simply take the tree, we manipulate it and hand it back and then generate the code. We might make checks with regard to the active language and say, "Hey does the active language support a width block?" For example and if it was Java Script it would say yes and if it was Visual Basic it would say yes and if it was C# and C++ it would say no but we ask the active language questions about what it's support and that may influence how we generate the code. We might take advantage of the feature for example of the language feature if it's there but there's nothing in our code that checks to see what language is out there. So as a result, when you write a refactoring or if you write some code declaration piece or something that's



maybe calculating metrics, you don't really have to worry about what language you're working with and it works with all the languages we support after you compile it. So that's another cool thing about this, that's actually the way it should be in Visual Studio as well but we're just not at point yet, we're not at the point yet where we can universally represent all languages in one way and that's something DX core does too.

Scott Hanselman: How much have you had to do that you shouldn't have to do? Like how much, are you doing a little C# compiler when you're off compiling C# and of you're off compiling VB in memory to think as long as you have to parse...

Mark Miller: Yeah, kind of. Yeah, we do have parsers for all the languages and we also have code generation aspects of it as well. So we have something similar to language services that Visual Studio has language services so like there's a C# language service and a VB language service and those things are responsible for compiling and parsing and IntelliSense, things like that. We have a similar structure as well, to our framework where we have language services essentially and ours are responsible for parsing, code generation and also for the rules of resolving types, finding out where types are located. For example, in one language we might find types by going through the imports or using statements from top to bottom and in another language it might go from like bottom to top. It might say the most recently imported one reference is going to be the first one I check, for example...

Scott Hanselman: Ah...

Mark Miller: So those rules sometimes differ from language to language and so we'll put in some code there that helps out in terms of resolving types when we create a new language service but those are the three components. Oh, and actually there's a fourth one, the fourth one is essentially the description of the language which is essentially a whole bunch of properties that describe like, for example does this language support width blocks, for example.

Scott Hanselman: Okay.

Mark Miller: And so we'll fill all those out like yes or no, we'll answer those questions all the way down the board for that language and so when that language is active and we're modifying it, we can go out and find out what it supports..

Scott Hanselman: Hmm, what do you think the future of this? I mean how far can you take this whole thing?

Mark Miller: Well, actually, no, I just realized I didn't answer your last question, let me answer that question.

Scott Hanselman: Okay.

Mark Miller: And the question was, how much did we write that we shouldn't have had to write? My answer to that is pretty much zero, Scott and the reason why is because it's hard to do what we're doing that gives us a competitive advantage...

Scott Hanselman: Okay.

Mark Miller: It's hard to paint on the editor that gives us the competitive advantage and so we'll take any competitive advantage we can get, right? If we can, whatever it is, there's something that's widely considered impossible and we figure out a way to do it, we'll take that. So I'm all over those kinds of challenges, those are actually, so my answer to question is zero. Your next question was what do I see for the future of this? Well, I think that the UI still got a long ways to go for most tools that are out there including Visual Studio. I think that this flattened experience where I'm renaming right inside the editor, it's a really slick good feeling, I feel like after I renamed I feel good, right, because I've done it in the fewest key strokes as possible, I haven't distracted myself from my focus which is writing code and I haven't slowed down my momentum at all, that's kind of the goal in this. So I think UI's got some place, definitely room to grow in the UI area and I also I think in the area of finding out what's wrong with your code in the background, finding things that are maybe hard for humans to find but are not so hard for computers to find. As an example, you're working on some code and you get a little message nearby that says, "Hey you know what, this code is really similar to code that I'm seeing in a few other places in the solution. We could take all of these different methods and bring them into one in an ancestor class as an example, with maybe only a few minor changes." And that I think we'll definitely see things like that in the future where you see more assistants, more pointing things out that are not obvious, that are hard to see because right now when we write code Scott, we're looking at a small, almost a postage stamped sized window relative to the entire book, the stamp book of code, right?

Scott Hanselman: Right.

Mark Miller: We're looking at a single screen and one file that can be one class out of thousands and being able to go out and detect potential problems and alert and show those to you in a way that doesn't distract you from what you're doing, I think is also something that you'll see in the future. That's the long term.



Scott Hanselman: One of the things that and I've kind of went to the competition right?

Mark Miller: Sure.

Scott Hanselman: Because I don't want to be an unbiased journalist but Resharper is pretty clear to people that it's a refactoring tool and it's a test and development tool and they're kind of carving out their niche. Over the years as I've used CodeRush I'm trying to like when you talk to a friend and you go, "Hey, CodeRush is cool, you should try it!" and then they'll say, "Oh, great I'll check it out, what is it?" I'm always like, "It's kind of this thing and it's kind of." The elevator speech is hard because it does so much. Do you think that that's difficult? That it is kind of everything in the kitchen sink?

Mark Miller: Yeah. I usually say, let me show you a demo is what I usually do. If I can get a demo going...

Scott Hanselman: Yeah.

Mark Miller: That usually, it holds people's interest for a long time and I can start showing them the different things that are there.

Scott Hanselman: Because it lets you read code fast because it has visualizers...

Mark Miller: Yeah.

Scott Hanselman: It lets you write code because it got all sorts of expansions and things and then its got the ability to refactor code.

Mark Miller: The things is though Scott, is if you tell somebody these things who has never used these things, they're first thought is probably like, "Well okay, how do I know I need those? I probably don't need those things. Those aren't things I've ever really realized I've needed." right? so that's one of the challenge, that's why a demo is usually better.

Scott Hanselman: Okay. I'm just thinking about the idea that one, well, of course, Visual Studio itself, is kind of always going to be nipping at your heels trying to do similar things when it comes to like refactorings and stuff.

Mark Miller: Right.

Scott Hanselman: Although certainly slower and it doesn't appear to be any danger for either CodeRush or Resharper not being useful in the sense of you guys are so far ahead doing stuff that's so out there. I don't see Visual Studio eclipsing you guys in any time soon. I'm just trying to understand that whenever, at this point basically in last couple of years when people say, "Well, should I use CodeRush or should I use a Resharper?" So try them both, watch the

demos, watch the screencast but it seems that amongst kind of alpha geeks, the folks that really demand performance from their system, everyone's using one of these tools. I think it's become clear that the Visual Studio editor by itself isn't enough to really do crazy stuff with huge code bases.

Mark Miller: Yeah, I think that's true. I think that there's, I mean if you think about just a number of refactorings that are shipping with Visual Studio, it's like, I think seven or maybe eight with C# and one which is renamed with Visual Basic. The refactorings that we have are extremely useful and once you get a sense of what's there, you realize that there are things that you do all the time and or things that would be really nice that if you could do them with the equivalent energy investment of a whim, right? For example, sometimes I'm writing code and I'm working on it and I'm thinking, "Oh you know what, I want to take this and I want to move this over here or I want to extract this or I want to, let's take a look at this." And without a tool that makes it easy, easy, easy to do these things to fix the code, what happens is this design debt, this technical debt accumulates and you end up having code that's not as clean as and doesn't give you as good of a feeling when you're working with it and so you really want subtle changes like refactorings to be effortless, effortless, effortless to apply and to use and once you start seeing these refactorings and you see where they are and you realize that, yes, they are things you do all the time, you just didn't realize that there was a tool out there for doing those kinds of things. So yes, even though the Alpha guys use it, it turns out that almost all developers could benefit from the functionality in tools like this, they can certainly work faster. There is a learning curve that is there but the learning curve is I think both Resharper and I know we certainly work really hard to flatten that out and to make it so that even programmers who've never seen a demo of CodeRush can go in and get something from it within a small amount of time of small investment, so start getting that return on investment very quickly.

Scott Hanselman: Yeah, that's cool. Well, thanks a lot, I really appreciate it. Where can people learn more about the architecture and what you're doing? Like if people want the technical details, can they check your blog or they check the DevExpress blog because I find this to be, kind of the most complicated and most interesting Visual Studio plugin that I've even seen.

Mark Miller: Well, you can find plugin tutorials actually on my blog, so it's at devexpress.com/blogs/markmiller I think is what it is, maybe Marky...

Scott Hanselman: I'll put that in the show notes.

Mark Miller: Yeah, well put that in the show notes so whatever that link is and then also if you only



search for DXcore, you can find out where to download that if you're interested in writing plugins and learning more about that and we also have a forum\newsgroup where people can ask questions about building plugins and there's a lot of plugin authors out there that are more than happy to help out new plugin authors in terms of what they want to do.

Scott Hanselman: Cool.

Mark Miller: So that's where they will learn more about that and then also, oh one other thing, Scott we should plug CodeRush Xpress because it's free and it's...

Scott Hanselman: All right, it is true, it is free.

Mark Miller: It's free and it's got I think something like 60 refactorings in it. It's got, something like 20 consume-first declaration features on it, it's got some cool navigation features, duplicate lines and some other neat features inside of there. So to look for, you can Google CodeRush Xpress and Xpress is spelled without the first E, it's just the capital X and then press...

Scott Hanselman: Or you can Google it with Bing.

Mark Miller: Or you can Google it with Bing, yes that's right.

Scott Hanselman: Thank you for saying that, now I have actually have that on tape, I have you saying Google it with Bing.

Mark Miller: Now is that going to help you for your review period Scott this year?

Scott Hanselman: Actually somebody got, someone was mad at me on Twitter, a couple of, they were not mad but they were like, "Yeah you've been really pimping Bing too much." Someone was complaining because basically a month and a half ago or two months ago, I promised that I would, I mean I use Google apps, I've been a Google user for a long time but I said, "I'm going to use Bing for a month and I'm going to, I didn't say I was going to blindly use it as a member of the big blue monster and of the collective but I said I will try Bing for one month."

Mark Miller: Yeah.

Scott Hanselman: And you know, darn it, it's working pretty well and they've had a number of releases since then and then the index is getting updated and I find myself for searches, specifically using Google maybe 2 out of 10 times. So Bing's getting it 80%+ of the time for me so, plus I like the pictures and I like the news and in the images search, so I'm using Bing for my home page in a number of my machines.

Mark Miller: Interesting. Yeah, so I like the pictures too. I've only tried it a few times but I was, in my mind while you're seeing this, I'm thinking, "Well you know, if you Google for great UI, you get my blog post is like number one and two on Google."

Scott Hanselman: Oh, really? Are you serious? Great UI.

Mark Miller: Great UI, but if you Bing it, I'm like number eight down there and so I'm like...

Scott Hanselman: Yeah, you are.

Mark Miller: I'm a little bit like, "Come on Bing guys, give me, pop me up there, somebody." I haven't quite figured out what the secret is.

Scott Hanselman: It's like Google bombing when you have somebody linked to it, you need to get your friends and then you need to Bing bomb yourself. You need to have a lot of people linked to you with that word.

Mark Miller: There's a Bing bomb out there huh? That's the secret.

Scott Hanselman: Well, have you ever done the Google bomb?

Mark Miller: You know, I've never done it but I think you showed me one a while back...

Scott Hanselman: Oh yeah, the best Google bomb is you go to Google and you type in French Military Victories...

Mark Miller: Right, yes.

Scott Hanselman: And then you hit I'm feeling lucky and then Google says, "Did you mean French Military Defeats?"

Mark Miller: Right.

Scott Hanselman: And it's all really true, it's all a lie when someone got enough people to point to this other site.

Mark Miller: Yes.

Scott Hanselman: So yeah, Bing bombing or Google bombing is when you have a lot of people linked to a term so that the, you basically fool Google into thinking that the internet thinks something about a certain topic.

Mark Miller: It's relevant, right.

Scott Hanselman: And apparently great UI Bing thinks you're eighth.



Mark Miller: Yeah.

Scott Hanselman: Which is very sad.

Mark Miller: So anyway...

Scott Hanselman: Well, thanks for taking the time to chat with us.

Mark Miller: I may switch over. Hey, you're welcome Scott, it's been fun.

Scott Hanselman: This has been another episode of Hanselminutes and we'll see you again next week.