



.NET Rocks!
The Internet Audio Talk Show
for .NET Developers
With Carl Franklin 
and Richard Campbell
<http://www.dotnetrocks.com>

<http://www.dotnetrocks.com>



Carl Franklin and Richard Campbell interview experts to bring you insights into .NET technology and the state of software development. More than just a dry interview show, we have fun! Original Music! Prizes! Check out what you've been missing!



Text Transcript of Show #351
(Transcription services provided by [PWOP Productions](#))



Rick Strahl on AJAX and jQuery
June 17, 2008
Our Sponsors



<http://www.devexpress.com>



<http://www.code-magazine.com>



<http://www.telerik.com/>



Geoff Maciolek: The opinions and viewpoints expressed in .NET Rocks! are not necessarily those of its sponsors, or of Microsoft Corporation, its partners, or employees. .NET Rocks! is a production of Franklins.Net, which is solely responsible for its content. Franklins.Net - Training Developers to Work Smarter.

[Music]

Lawrence Ryan: Hey, Rock heads! Quit waxing your board and listen up! It's time for another stellar episode of .NET Rocks! the Internet audio talk show for .NET developers, with Carl Franklin and Richard Campbell. This is Lawrence Ryan announcing show #351, with guest Rick Strahl, recorded live, Sunday, May 25, 2008. .NET Rocks! is brought to you by Franklins.Net - Training Developers to Work Smarter and now offering SharePoint 2007 video training with Sahil Malik on DVD, dnrTV style, order your copy now at www.franklins.net. Support is also provided by Telerik, combining the best in Windows Forms and ASP.NET controls with first class customer service, online at www.telerik.com, and by CoDe Magazine, the leading independent magazine for .NET developers, online at www.code-magazine.com. And now, the man who's trying to get a script for post TechEd stress disorder, Carl Franklin.

Carl Franklin: Thank you very much. Welcome back to .NET Rocks! Carl Franklin and Richard Campbell live at PWOP studios.

Richard Campbell: I got to love it, sitting face to face.

Carl Franklin: This is how radio should be.

Richard Campbell: Absolutely.

Carl Franklin: Usually in Vancouver, I can't see him when we're talking.

Richard Campbell: No, just a disembodied voice.

Carl Franklin: Right, but this is more fun. We've never even done Mondays like this when you've been here.

Richard Campbell: No, no because whenever we do Mondays, I'm in the booth across the hall.

Carl Franklin: That's right, that's right.

Richard Campbell: It's a l w a y s Mark and Karen.

Carl Franklin: Because they are funnier when they can see each other.

Richard Campbell: Well, she keeps flashing him.

Carl Franklin: Can we say that in .NET Rocks?

Richard Campbell: I'm just stating facts here.

Carl Franklin: Let's just say if you are a Mondays listener, and if you're not, Richard and I are the voices of reason on that show.

Richard Campbell: Oh yes, we are the same ones, no two ways about it.

Carl Franklin: Yes. Well, let's just get right into Better-Know-A-Framework.

[Music]

Richard Campbell: All right sir, what have you got for me?

Carl Franklin: So, I want to talk about String.Format.

Richard Campbell: Oh really?

Carl Franklin: Specifically about the feature of the .NET Framework called composite formatting.

Richard Campbell: Okay.

Carl Franklin: You might have done it but never named it composite formatting, but composite formatting is when you use a format string that has the braces and the end braces...

Richard Campbell: All the symbols in it, yeah, yeah.

Carl Franklin: The symbols, yeah, exactly. I started in the docs with String.Format and of course that brings you to the docs on composite formatting and here's what it has to say about that. The .NET framework composite formatting feature takes a list of objects and a composite format string is input. The .NET Framework composite formatting feature takes a list of objects and a composite format string as input. A composite format string consists of fixed text intermixed with indexed placeholders, called format items, that correspond to the objects in the list. The formatting operation yields a result string that consists of the original fixed text intermixed with the string representation of the objects in the list. So basically, the classic example is you have a telephone number that's just numbers.

Richard Campbell: Right.

Carl Franklin: And you want to format with all the things, or you have a number that you want to use, dollar signs and put commas in and you format as currency so there is a C for currency. So there's just a whole bunch of these little rules and I think if you're searching through strings and you want to identify text, you are going to use regular expressions, but if you're producing strings, that need to be formatted in a certain way.

Richard Campbell: But, you know, in the case of stuff like dates and currency and stuff, there are a whole internationalization of strings featured separate from this.

Carl Franklin: That's right.

Richard Campbell: They're probably the recommended way to go about it. It's when we get into custom formatting around our park numbers and those kinds of things, you need to have products that this format is so useful.

Carl Franklin: Well, actually, what happens is the composite formatting uses the culture of Windows whatever that's on.

Richard Campbell: Okay.

Carl Franklin: And they all have their own formatting rules because you can see it in the docs where there's an example where you have an N to 100 and then you use a format for -- and you can just do this with WriteLine too, you don't need a String.Format.

Richard Campbell: Sure.

Carl Franklin: But Console.WriteLine, if you pass in "{0:C}" as the format string, it displays \$100.00 to the console on computers that have US English as the current culture.

Richard Campbell: And that's the way you want to go about this so that if the user is actually configured for a different culture, you just say, "I want you to format this in the currencies appropriate."

Carl Franklin: Exactly and that's why you're not doing specific formatting in the code. So, this is a good idea, composite formatting. Look it up, know it, love it, use it, it's Better-Know-A-Framework, folks.

Richard Campbell: Nice.

Carl Franklin: Only on .NET Rocks! Yeah, I should have gone into car sales.

Richard Campbell: There you go, maybe.

Carl Franklin: Or game shows. Oh, wait a minute, we do that.

Richard Campbell: We're in game shows.

Carl Franklin: Yeah, I'm sorry. Richard, you got some email for us.

Richard Campbell: I got a quick one for you here.

Carl Franklin: Why is there a dollar bill between us? That's what I...

Richard Campbell: I don't know. It's kind of odd.

Carl Franklin: Somebody stuffed the dollar bill in the window between the two booths and I don't know who it was and you can't get it out. It's double-paned glass. First of all, I don't know how it got in there or why it got in there but I'm going to have to take the staff aside and find out who did that.

Richard Campbell: Nice. It's there to torture you. All right, I got a quick email for you. The subject line is ".NET Rocks is a Gateway drug."

Carl Franklin: Uh-oh.

Richard Campbell: "Richard and Carl, I just wanted to let you know that since my first hit of .NET Rocks, I've been addicted to PWOP podcasts."

Carl Franklin: Awesome.

Richard Campbell: "This has affected my podcast listening to the point that I had to remove two other subscriptions to fit Hanselminutes and RunAs Radio. Not only has my listening to .NET Rocks caused changes to my iPod, but also to my whole life. My wife gets sick of me having earphones on or reading books about .NET, SQL Server, or other great technologies. The only good she sees from all this is that when I talk to her about it, I can put her to sleep quicker."

Carl Franklin: Nice.

Richard Campbell: "Keep up the great work. In several recent shows, there have been discussions about going back to basics, I cannot agree more with this idea. I am telling everyone I know about the show and several of the developers and managers could use information about the basics more than the details of the latest technologies."

Carl Franklin: Yes.



Richard Campbell: "With all the recent additions to the Microsoft technology, I think many people are forgetting about the basics like good architecture."

Carl Franklin: Like good architecture, exactly.

Richard Campbell: That email is Gifford Haynes. Gifford, we're with you, man. Sorry about the addiction.

Carl Franklin: Basics like composite formatting. This is what I'm talking about.

Richard Campbell: Yeah, you know, I really think running through some core patterns and start thinking about architecture again, these are all good things and I'm sorry I don't have that show for you today. It's a bit of a machine here. We got to drain the shows that we've got out.

Carl Franklin: That's right. We recorded a lot of shows at TechEd, so we got to burn through those first.

Richard Campbell: Right but I will start sliding this, I guarantee you.

Carl Franklin: For example, Jeff Richter is going to be on the talk about memory management under the hood of Windows.

Richard Campbell: Absolutely.

Carl Franklin: Which is going to be so fun.

Richard Campbell: Yeah, some good things coming up. I'm listening. We're doing what you ask.

Carl Franklin: Richard, as you know, our friend Greg Brill down at Infusion has hired about 14 of our listeners...

Richard Campbell: Wow.

Carl Franklin: Since we started talking about the New York City Tour.

Richard Campbell: Way back when. It's been a while.

Carl Franklin: It has been a while. They want another round of tourists to come and work for them. It's a great company right in Manhattan. They're going to move you. They're going to put you up in an apartment in Manhattan and pay your rent for a year.

Richard Campbell: They also have offices in London and in Dubai.

Carl Franklin: Yes. Dubai, if you want to go to Dubai, they got offices there.

Richard Campbell: Sure. So, I know that New York is always...

Carl Franklin: Canada, Toronto also.

Richard Campbell: They have a Toronto office but New York is always the big driver, but just be aware, if you enjoy this experience, there are more places to go, more opportunities to be had.

Carl Franklin: That's right and I don't think they are offering to put you up in an apartment for a year anywhere else but New York.

Richard Campbell: Right.

Carl Franklin: I think it's a special deal for New York City because, you know, cost of living, etc.

Richard Campbell: Absolutely.

Carl Franklin: Can you imagine just hanging out in New York for a year in your own apartment, like in the city?

Richard Campbell: Yeah, have the experience.

Carl Franklin: Have the experience, work with some really creative guys. Anyway, if you want to read more about that, go to shrinkster.com/kh6 and have fun while you're there.

Richard Campbell: You will.

Carl Franklin: Eat some pretzels. And with that Richard, let's introduce Rick. Rick Strahl is the big Kahuna and janitor at West Wind Technologies on Maui, Hawaii. The company specializes in Web and distributed application development, develops commercial and free tools, and provides training and mentoring with a focus on .NET, IIS and Visual Studio. Rick's a C# Microsoft MVP, a frequent contributor to magazines and books, and a frequent speaker at developer conferences and user groups all over the world. He is also co-publisher of CoDe Magazine. For more information, you can visit www.west-wind.com/weblog or contact Rick at rstrahl@west-wind.com, that's R-S-T-R-A-H-L. Hi Rick, welcome to the show.

Rick Strahl: I'm glad to be here, first time for me.

Carl Franklin: Yeah. We run into each other at conferences all the time and we have been -- I don't know, this sounds like an apology almost like, you know, we've been meaning to have you on the show.



Richard Campbell: Come on Rick, you know perfectly well I've been pursuing you for a while. It's been you who's been the coy one.

Rick Strahl: Yeah, well, I didn't want to mention. Yeah, Richard has been trying to track me down. He had me cornered at the MVP conference and I finally had to give in.

Richard Campbell: Yeah, surrounded by Phil Haack and Rob Connery and it's like, "What do I got to do, man? "What do I got to do to get you on the show?"

Rick Strahl: Yeah, it was a little hard to turn that one down.

Richard Campbell: The listeners have been asking about you. You write a great blog.

Rick Strahl: Thank you.

Carl Franklin: Yup and you always have great sessions. When people are coming out of your sessions, I see like the heads spinning and they say, "Oh my God. That was awesome."

Rick Strahl: That's cool. Yeah, I try to get as much content in it as I can and sometimes that has that effect of head spinning.

Carl Franklin: Right. You've always been a sort of a JavaScript AJAXy kind of guy and web development is your forte. jQuery is something you've been into lately.

Rick Strahl: Yeah, jQuery is one of the tools that I have been using quite a bit lately and it's one of those things that have gotten me excited. So, when you guys asked, "What's the latest thing that you're into?" I think jQuery would fit the bill quite nicely.

Carl Franklin: So, for the absolute novice, can you give us the elevator speech?

Rick Strahl: Well, I guess we should back up a little bit and just talk about where jQuery really fits which is JavaScript in client side web programming, and you just mentioned a second ago that I've always been an AJAXy, JavaScript kind of guy and actually that's absolutely not true like most of the world, I've pretty much hated the JavaScript for the longest time and it really wasn't until AJAX came around that I even started thinking about JavaScript seriously. I mean everybody has probably done some JavaScript development over the years in the course of web development but really not seriously. You know what I mean? We're usually talking about doing a bit of validation here or opening a new window or

something like that where a serious development with JavaScript on the client side and actually doing something useful in terms of UI composition and serious processing on the client is something that's rather new and I was no different. It wasn't until about two years ago I would say that I started even then on using JavaScript at all.

Richard Campbell: I've always looked at JavaScript as a necessary evil.

Carl Franklin: Yeah, me too.

Rick Strahl: Yeah and it kind of is because if you really think about it today anyway, I would say JavaScript is probably one of the most popular languages in terms of the actual usage.

Richard Campbell: Or lines written.

Rick Strahl: With the amount of people that actually are using it, I would say it's probably more used than Java or C# or VB or anything like that. Everybody has at least used a little bit of JavaScript and everybody knows it and it used in a heck of a lot of web pages actually.

Carl Franklin: We used to have to get our hands into it, I mean before, seriously it was before ASP.NET really and even when ASP.NET was just first out, we were still tweaking JavaScript, I was, but then, you know, the tools got really good and I just sort of forgot about it.

Richard Campbell: I wonder just how much, what the percentage is between generated JavaScript and handcrafted JavaScript.

Carl Franklin: Yeah.

Rick Strahl: That's exactly where the things are changing I think. So in the past, it's been we put a few little snippets here and there and now it's becoming more of something that's actually part of the development work that happens on the client side. Some of it is generated so if you're using ASP.NET AJAX, of course a lot of it is generated for you. If you are doing more client-centric, purely client-centric development where you're talking to services that return data and then you update the DOM from that, then you're writing a lot more JavaScript than your own and even on that space, I think a lot of things are changing quite drastically and that there are now a lot of libraries available that you can choose from that you don't have to write everything from scratch. So in the past, you probably have to write some sort of framework to reuse some functionality on your own and now there is a ton of different frameworks that are relatively mature that you can use to help you with that sort of thing. So some of the things that you



typically need to do and you need wrapping up is things like the AJAX functionality on the client so the amount of code that goes into making an AJAX callback if you do it with raw XML, HTTP calls, is fairly severe. If you want to do it right, do error handling and do everything that goes with the actual callbacks like routing it to the appropriate handler and whatnot. It actually ends up being a fair amount of code and when you do that, you don't want to rewrite that code every time. So, all of that sort of stuff needs to be wrapped up in libraries.

Carl Franklin: Okay.

Richard Campbell: I think most folks are using AJAX exactly that way whether it's the, I still call it the Atlas library but I suppose that's wrong but you know what I mean.

Carl Franklin: Just outdated, that's all.

Rick Strahl: Yeah.

Richard Campbell: Yeah, why do we need -- the only reason I could see that we would want to touch that code by hand is that those libraries get a little fat.

Rick Strahl: Yeah. Well, there are actually a couple of things. So when we're talking about ASP.NET AJAX, or Codename Atlas as it was in the beginning, the problem with that particular library for the client side stuff, it doesn't really offer you a whole lot of functionality in terms of what you can do with it on the client side if you are doing client-centric development. So Microsoft AJAX provides you functionality of server-side programming and it generates a bunch of JavaScript for you. So update panels, for example, there is a boatload of JavaScript that gets generated into the page and in the libraries that come into play actually to make all of that work. That's a very service-centric approach but if you want to actually write pure JavaScript code on the client side which means you actually write your own JavaScript code to update page elements and pull data down from the server using maybe the web services, then all of a sudden you have to write your own code and then the library is actually not that useful for a client side development.

Richard Campbell: Right. You know I think we talked about this, Carl, with Dino Esposito and he was pointing out that's a very interesting fact that there are different ways to implement AJAX or at least asynchronous updating of web pages.

Carl Franklin: Right.

Richard Campbell: And some of them including ASP.NET AJAX still put all the work on the

server where what we want to do is harness the client.

Carl Franklin: Well, you know, client machine's power has changed a lot and bandwidth has changed and a lot of things have changed since the early days. Where does jQuery fit into all of this?

Rick Strahl: Okay, so jQuery is basically a pure client script library. It provides functionality that is otherwise not available at least for ASP.NET developers in the ASP.NET AJAX library. So what it's biggest strength is essentially that it provides you the ability to manipulate the DOM in a very easy to use manner which is of course, in my opinion anyway, the biggest pain point of doing JavaScript. It's not so much JavaScript by itself but it's manipulating the browser.

Carl Franklin: Now, the DOM of course is the internal Document Object Model that the accesses all different, the ways that you can access the different elements in the browser on the page and in Windows and all of that kind of things. Is that what you're talking about?

Rick Strahl: Correct.

Carl Franklin: So, I guess it simplifies the amount of code that you have to write. You can sort of stuff a whole bunch of what would be lines of hand coded JavaScript into one jQuery line, is that the whole idea?

Rick Strahl: It's not just one line but yeah, it greatly reduces the amount of code that you have to write inside of JavaScript to manipulate the DOM. The mechanism that jQuery uses for making that happen is, well, in fact there are a number of things that come into play. One of the things is selectors which allow you basically to go in and select elements in the DOM using CSS-like syntax. When you are working in an HTML document and you're using client-script to manipulate that document, one thing that comes up over and over again is how do I actually select an element that I want to update or that I want to make a change to and often times that happens to be a single element or sometimes it's a whole bunch of elements together that you want to manipulate all in batch. What jQuery does is that it has the concept of selectors where you can call the jQuery function or dollar sign which is a shortcut for the jQuery function that allows you to select elements in the document and so you can go in and say I want to select all the rows on the table, for example, and get back a jQuery object which is essentially an array-like object that gives you all those elements selected from that one CSS selector syntax feed.

Carl Franklin: Sounds a little like LINQ.



Rick Strahl: It is in a way, yes, you can call it -- in fact, there are a number of similarities with LINQ. One of the things that you can do once you have a selection of objects, DOM elements, that you've actually pulled out of the document, you can then perform actions on those documents. You can either iterate over it manually if you choose to, or jQuery provides a few, I don't know how many functions there are actually, but quite a few functions that you can operate on that matched set. So you can say in batch, for example, all those elements that have been selected apply a certain CSS class against it, or go in and change this particular style against all the elements that have been selected.

Carl Franklin: It's an interesting thought about LINQ and jQuery because there is a project on CodePlex called LINQ to JavaScript.

Rick Strahl: Okay.

Carl Franklin: It's an implementation of LINQ to objects implemented in JavaScript. I wonder what the difference is. I don't know much about this but I just happened to see it.

Rick Strahl: I haven't heard of it so I can't really say but I don't know.

Carl Franklin: It's built using a set of extension methods built on top of the JavaScript array object, but I'm not sure if it's for a LINQ to the DOM or if it's just built using JavaScript to access data in JavaScript, I'm not sure.

Richard Campbell: Yeah, it sounds like a way to create syntax to go through a JavaScript array.

Carl Franklin: Yeah, that's what it looks like. We'll get the skinny on that and we'll look into it.

Rick Strahl: Okay, to some degree that sounds a little bit similar to what jQuery actually does. So the effect is essentially that when you do this selection, you get an array, or it's actually not an array, but it's an array-like object that has a length and that has actual elements inside of it and then you can apply all these functionality. So there are a couple of cool things that you can do. The first thing is there are a ton of methods that you can apply against it. So there are things like styling so you can go in and use for example this CSS method to apply individual styles against all the matched elements...

Carl Franklin: Nice.

Rick Strahl: You can go and add class names, so add classes actually a method that you can apply to it which is very, very useful. For example, when

you might want to use add classes, you can select every other row in a table and provide a kind of alternating row-type display and then apply a class like, say, a grid alternate CSS class against it to give it a different color. So it's one line of code, you basically do that without having to write any or even service-side code to render, say, a different display mechanism for that particular row, for those alternating rows.

Richard Campbell: That's interesting because you could write JavaScript for that. You can definitely walk through that grid and apply a different style alternating between the rows.

Rick Strahl: Absolutely but it's a heck of a lot easier to do it with one line of code, right?

Carl Franklin: Right.

Rick Strahl: So, the selection actually gives you all those elements that you want and then you can apply functions against these matched elements. So there are tons of stuff you can do. Some of the things that I've just mentioned are just CSS styling but of course you can also do, if you want to, with pure CSS styling and because essentially the syntax that you use for selectors is CSS 2.1.

Richard Campbell: Okay.

Rick Strahl: CSS selectors are part of the actual specifications for CSS, so a lot of the syntax that you use with jQuery to select the actual element is based on pure CSS syntax. So, if you're an HTML wonk and you know how CSS selectors work, jQuery will be like second nature to you already. If you don't and that's what's really kind of my thing, I wasn't really all that great with selectors to start with, once I started using jQuery I got a whole lot better at using CSS selectors all of a sudden because I was using it with jQuery and realized "Hey, I can actually do this with CSS selectors a lot," so you can use that same syntax in your CSS style sheets.

Richard Campbell: Maybe we need to go over exactly what a CSS selector is and how it works, why it's cool.

Carl Franklin: That's a good idea.

Rick Strahl: CSS is of course Cascading Style Sheets and you can use a style sheet for styling in HTML documents. Styles are used as typically if you use a CSS style sheet is you typically create CSS classes within that style sheet and then you can apply those classes to individual elements.

Richard Campbell: Right. So, these classes are collections of style elements?

Rick Strahl: Correct and they are applied against any of the elements that have a class equals in the HTML markup.

Richard Campbell: Right.

Rick Strahl: So, if you have a DOM element and it has `class="gridalternate"`, then it will look up in the CSS style sheet, find that particular class element and then apply all the attributes that you actually chose, all those styles that you actually apply to it.

Richard Campbell: It might be that it has a different background color, that it has a different hover over color, that it's in a different font, I mean there are a lot of things you can put into a class like that.

Rick Strahl: Correct. So, that is a class and that's what most people are probably familiar with including myself sometime ago and then of course you can also do things like use the pound sign to select an individual element so you can say `#gdEntries` and point it maybe at a data grid about `gdEntries`. It's actually selector syntax where you're saying point at this particular element and then apply whatever elements you have defined on it. It's defined the same way as you define a CSS class except instead of using the dot class, you use pounds element name. You can use a comma maybe say, "I want another element that also has that same class applied to it," so I can say `#gdEntries#` whatever, error display, and have it be another DOM element and then it applies to that.

Carl Franklin: Hey, this is Carl. I just need to take a minute to tell you about the latest offerings from our friends at Telerik. As you probably know, they have recently released their huge pack of web controls built on top of ASP.NET AJAX that will help you build impossibly fast and interactive applications in no time, but you've just got to check out their Windows Forms stuff. It looks just like WPF. How about a Carousel component? In Windows Forms? How about a super powerful Grid View control? And 32 other desktop components with dazzling WPF-like features. In reporting, Telerik has this new design surface that simulates graph paper. It's got so many advanced page layout capabilities it looks more like graphic design software. So, visit www.telerik.com and download a free trial and make sure you thank them for supporting .NET Rocks!

Now, this has been quite a technical discussion and a little bit dry but I just want to point out this jQuery UI website which we're linking to is ui.jquery.com with some extremely beautiful demos of things that...

Richard Campbell: It's very WPFish, isn't it?

Carl Franklin: That's exactly what I was thinking, Richard. It's very Silverlight-like, especially that first one, that slider with a semi-transparent slider, oh man. Oh, the second one I guess, the slider gallery, just unbelievable.

Rick Strahl: Yeah, and the cool thing about this is that a lot of these plug-ins that you're looking at actually are done with very little code. So, these plug-ins are actually designed in such a way that you can get them up and running with almost no code at all. Basically, there are extenders that extend existing controls, so by simply providing a few simple markup tags on some of these elements, it's very, very easy to JavaScript to attach these behaviors onto these controls. There's a number of really, really cool controls. For example, there is a sortable control that allows you for example to have a static list, a UI list, unordered list in HTML, and make it sortable and when you actually sort it, it has nice UI effects associated with it so when you drag an item into the next one, it kind of slides into place instead of just kind of going to the next thing and being there and then showing up.

Carl Franklin: There's a beautiful image cropper. Did you see that, Richard?

Richard Campbell: Yeah, I did. I'm still trying to think about all the coding that go behind this. It looks like it would be a barrier to code.

Rick Strahl: Well, that's the beauty of it. First of all, if you're using these plug-ins, you're writing very, very little code. Most of these plug-ins are designed with, literally you write like two or three lines of code. For example, the sortable that I mentioned which you can actually look at in the jQuery Photo Album I linked here and which is at shrinkster.com/y67, if you go to the admin page there, you can actually see that sortable and what you can do there in order to make that happen, actually there is literally one line of code that points at the list, you select actually the list, UI list, and then you say `.sortable` and then you apply one or two attributes to that to actually limit it to certain elements that you want to sort and handle and that is all it takes. So it's literally one line of code with a few options that you have to apply in an actual object selector. It's very, very easy to make that happen.

Richard Campbell: There's obviously more code underlying it but you just wrapped it up in such a way that you were able to do this relatively painlessly.

Rick Strahl: Well, actually I didn't do it. This is actually part of the ui.jquery.com, yeah?

Richard Campbell: Right.



Carl Franklin: I'm looking at the code for the cropper and it's nothing.

Rick Strahl: Yeah, there is nothing there from an implementation standpoint of the end user. Now, of course, there are a lot more code on the inside of these controls, of these plug-ins.

Richard Campbell: Right and you see the script references there, jquery.jas and ui.jas, like that must be the meat.

Rick Strahl: Right. There are typically two other libraries that you use. You use jQuery and there is a kind of a UI-based which is some core functionality and then the actual implementation of the actual plug-in that you usually use. But if you actually take a look underneath the covers of some of these controls where there are plug-ins, you find that there is not a horrendous amount of code involved with them. I mean that will a hundred lines of JavaScript code which considering the functionality that some of these things provide, it's not all that much and that kind of a test to the power of what jQuery can accomplish for you. It just produces the amount of JavaScript code that you write drastically.

Carl Franklin: Yeah.

Richard Campbell: The functionality is lovely. I feel like suddenly we have real JavaScript controls.

Rick Strahl: Yes. In a way, it feels like this is what JavaScript should have been.

Richard Campbell: Right.

Rick Strahl: You know what I mean? It just makes it so much easier to actually manipulate the content of the DOM by moving things around, by being able to update content very easily, and the most important thing maybe that I haven't mentioned yet is the fact that jQuery isolates you from all the differences between different browsers for the most part. So, when you apply certain styles, for example, like let's say you want to apply an opacity to something which is a specific CSS style that's not supported directly by all browsers, Internet Explorer doesn't have an opacity style, for example. When you apply that, jQuery is actually smart enough to know that Internet Explorer does it a certain other way which is using a filter and uses that instead to apply opacity.

Carl Franklin: Wow.

Rick Strahl: And the same thing can be applied when you're using, for example, retrieving a width of an element or setting the text of an element which is different in various browsers.

Carl Franklin: I see that you on your blog have an example where you're using jQuery with WCF.

Rick Strahl: Yes.

Carl Franklin: So, WCF has stuff that you can use from JavaScript?

Rick Strahl: WCF actually has, well, let's back up here a second. So, in version 3.5 of WCF which is the version that just shipped with .NET 3.5, there is a new mechanism for REST services and so basically it provides an official API for accessing web services through WCF.

Carl Franklin: Cool.

Rick Strahl: So, in a nutshell, it's essentially what we have with ASMX services in ASP.NET AJAX and providing it now out to WCF as well. There's actually a whole lot more there. You can provide pure REST services, whereas, the stuff that I'm showing in that blog post is specific to the AJAX style callbacks.

Carl Franklin: Okay.

Rick Strahl: So, what it allows you to do is you can essentially cut out ASP.NET AJAX on the client side and just use jQuery directly to talk to the WCF back-end.

Carl Franklin: Okay. Wow, that's cool. So, what is the demo actually?

Rick Strahl: The demo that I'm using there is a Stock application, Stock Portfolio Manager, and so it shows off some of the effects features that shows off how to actually get the data and pull it in and then also to how to interact with some of the REST features that are provided by WCF 3.5 which is actually serving images, for example, from the web service and directly displaying it inside of the page. It provides a nice cross-section of some of the functionality that jQuery actually provides.

Carl Franklin: Now, you also wrote a photo album?

Rick Strahl: Yeah.

Carl Franklin: This is pretty nice. Nice fade-ins and dragging windows around semi-transparently, beautiful.

Rick Strahl: Right. All of that is done using jQuery and, again, most of these effects and some of these things like draggables and droppables are part of jQuery UI and they are very easily done. Again, in a lot of cases, it's literally one line of code that you

attach to a selector to make an element draggable and to be able to hide and show elements and provides effects on that actual display mechanism.

Richard Campbell: Man, are we ever blurring the line between the whole SmartClient like this whole drag-and-drop into different frames and stuff, tell me where the SmartClient ends and the web client begins.

Carl Franklin: Yeah. I love these kind of web apps that real nice windowy kind of things.

Rick Strahl: Yeah, but ultimately building these kinds of web application is still difficult even if you use something like jQuery. I mean it is much, much easier than you would expect it to be but it's still a lot of work because, remember, when you're building web clients like this, you're still in a disconnected environment.

Carl Franklin: Right.

Richard Campbell: Right.

Rick Strahl: So, you're not writing a fat client application where all the data is always available and quickly to you. Everything has to be done asynchronously. You have to download the data and then update the document after the callback returns. So there is always a little bit more work involved in building web UIs like this.

Carl Franklin: Some nice photos, too.

Rick Strahl: The nice thing of course is when you do it with the web, it's readily available for anybody to see without any installation.

Carl Franklin: I'm just looking at the pictures. Sorry.

Richard Campbell: That's what happens when you live in Hawaii.

Carl Franklin: Oh, some nice photos there. Did you take these?

Rick Strahl: Yeah, all of those photos in that photo album, I took, yeah.

Carl Franklin: Very cool.

Rick Strahl: There are tons of other photo albums that use that same mechanism, so if you want to check that out, that's cool.

Carl Franklin: How does this interact with the Visual Studio or .NET? I mean it's just standard

JavaScript when it comes down to it so the JavaScript editor is your friend, I guess.

Rick Strahl: That's true. So when you write code with jQuery, of course you are going to write some JavaScript, but all things considered, the amount of JavaScript that you actually write is not that much. So the Photo Album, I actually checked it out, is I think 120 lines of code actually behind that, or something in that order. So it might have changed a little bit since I've actually worked at it last, but there is not that much code actually there. So you're working within Visual Studio and Visual Studio is your friend and it does help having the latest version of Visual Studio to write of course. If you haven't done so, you definitely want to install the latest hot fixes for Visual Studio...

Richard Campbell: For 2008?

Rick Strahl: For 2008, yes. It's not the service pack that's coming in sometime, but there is actually a hot fix that you can get for it and what it does, one of the features that they've added there is better support for third party libraries including jQuery.

Carl Franklin: Okay.

Rick Strahl: So, you can actually get some IntelliSense support for jQuery there.

Richard Campbell: Nice.

Rick Strahl: Which natively wasn't available previously. There are a few things that you can do to improve that on top of that which is basically by annotating the JavaScript libraries with some of the ASP.NET AJAX syntax that they use for IntelliSense. So there are a few tags that can be added into the documents like a reference and parameter types and whatnot that make it much more interactive yet than what you got natively.

Carl Franklin: Okay.

Rick Strahl: I think from what I've heard is SP1, and I have not installed that yet since I'm on the road, is supposed to make all of that functionality natively available so that you don't have to patch anything manually which is kind of nice.

Carl Franklin: Yeah, that's kind of nice.

Rick Strahl: I'm looking forward to that.

Carl Franklin: So, you also have links to a couple of reference things here. There's a jQuery reference print sheets for printing them out. Is there that much to memorize that you need a print sheet?



Rick Strahl: Yeah, there's a fair amount of stuff to memorize. Again, I don't have an actual number of functions here, but I'm looking at a sheet that's printed in about 8-point and it's full.

Carl Franklin: Wow.

Rick Strahl: So, there's a lot of functionality that you can do. Just to give you some idea, we've looked at some of the CSS functions. There is a function for traversing the document. There is DOM manipulation so you can apply HTML, you can apply text, you can retrieve a value very easily. There are things like being able to append HTML DOM content to other elements or appending to the current element that you are on and actually appending other elements into it. There is functionality for effects. There is event handling, which is a big one actually.

Carl Franklin: Event, I was just going to say that. That's huge.

Rick Strahl: Yeah. Event handling is pretty important and jQuery makes that a snap too. It basically provides you with event handler functions that you can apply to a selector. For example, you can say selector, whatever the selector is and then simply create a function on that and that function will actually be called on any click event that occurs, so it's an event handler that gets hooked up natively. What is nice is jQuery provides a couple of things. First of all, it always calls you in the same context. So it always calls you in the context of the actual element that was clicked or hovered over or the key was pressed on or the mouse moved over or whatever. So this, in the context of that event, is always going to be that DOM element. In addition, it also passes you an event object so jQuery basically provides you a normalized event object that gives you the capability to look at any browser and get the same event parameters passed to you. Different browsers return event information differently so if you look at the source object in Internet Explorer, it will be something different than the source object that you see in a Mozilla-type browser.

Carl Franklin: Right and that is where you have to have code that forks to say if my browser is this, do this; if my browser is that, do that, knowing really what's around that code.

Rick Strahl: jQuery essentially normalizes all of that and gives you an event object that you can talk to and it will always give you consistent values back.

Carl Franklin: Awesome.

Richard Campbell: That's amazing, really. Now, we talk about the sort of core things that this is

doing for us. Giving us a common object model is a big deal.

Rick Strahl: Yes, absolutely. So, jQuery is really not the first library that has come up with this idea of normalizing the DOM. There have been several other libraries that did this beforehand, the most popular probably being a prototype which is another JavaScript library that's also quite popular. jQuery has come a little bit later but it has gotten quite popular very, very quickly because it's got a simpler model for working with the DOM in general and the whole way that the community has grown around jQuery is really quite exciting and I think that's a lot of what has contributed for jQuery success and becoming so visible so very quickly.

Richard Campbell: Yeah, what's interesting to me of course is that as soon as I hear framework around JavaScript, I think that makes JavaScript bigger, and the problem I already have was that JavaScript was too big.

Carl Franklin: Or the code that you're actually executing is too big. I mean that's what it really comes down to because you're JavaScript itself, when it gets loaded, that is what can take a long time and especially if you're doing a lot of UI stuff.

Rick Strahl: Yeah. So, jQuery is trying to be a very small library so one of the goals of the library is to stay very compact. So the core jQuery library is actually very, very small. No, not very, very small but fairly small. So the compressed library, when you zipped it and packed down is about 19K.

Carl Franklin: Wow, that's great.

Rick Strahl: Then of course if you want to do some of the more interesting things like the sortable or the drag and drop, then you have to add additional libraries to it but most of those libraries tend to be very, very small because the core of jQuery engine makes the code that you write actually much more compact than you normally would.

Richard Campbell: Well, I like the ability to be granular so you can take that function now if you don't need it.

Rick Strahl: Right, exactly.

Carl Franklin: Now, I'm looking at the PNG file that has a reference card on it which is a very nice diagram by the way. I don't know if you did this but it is beautiful.

Rick Strahl: Yeah, I didn't do this so I just link to it. These guys have done a really nice job. They also have a cool iPhone version of this thing which is kind



of interesting. You can actually, if you have an iPhone, you can navigate to it and it's actually quite nicely done so that you can click and it looks like a native iPhone application even though it runs over the web.

Carl Franklin: Well, the thing is this really brings home what you can do with it just by looking at this reference chart. I know you mentioned this but it's worth talking about again, it's that this can work with AJAX and it has sort of higher level wrappers around the AJAX things. Now, is it working with ASP.NET AJAX or does it work just as it has its own AJAX implementation? What implementations is it using? Is it its own?

Rick Strahl: Yeah. So jQuery includes its own AJAX functionality and as far as interoperability goes, if you want to use it in combination with ASP.NET AJAX, you certainly can. There is no conflict but you can also use it by itself and if you want to do AJAX to WCF as I mentioned earlier, you can basically just use jQuery and not use any of the Microsoft AJAX libraries if you choose to do so.

Richard Campbell: This gets back to sort of the bare essentials of what AJAX was about which is I have a web service out there that can retrieve data for me and the client is calling it, fetching and rendering it.

Rick Strahl: That's pretty much it, yes, and this is kind of what I call the raw AJAX programming where you're just dealing with the server as a data service as opposed to an HTML generated.

Richard Campbell: Right, HTML generated on the client and not on the server.

Carl Franklin: Yeah.

Rick Strahl: Absolutely and there are a lot of different ways that you can actually accomplish that and jQuery actually makes that process and that this is really would excel and make that process a whole lot easier.

Richard Campbell: So, I got to think that where this is hard is that it's really not a visual design tool.

Rick Strahl: No, it's not, yes.

Richard Campbell: This is back to building UI with code.

Rick Strahl: Yeah, but I mean when we're looking at web development in general this days, as complex as web development is getting in terms of

layout, the visual tools, I mean think of the Visual Studio editor, how visual is that really.

Carl Franklin: They haven't really caught up.

Rick Strahl: Yeah and if you really think about it, I mean if you're building complex sites, the truth is that Visual Studio editor gets in the way more a lot of times, the visual editor anyway. For designers especially, a lot of people are actually using just the HTML markup window to do most of their layout, so this is really no different and you just have to run it in order to see what it's actually going to look like.

Carl Franklin: Oh man, there's some nice stuff here. I'm looking at some of your links to the plug-ins, the jQuery cycle plug-in, which cycles through photographs by fading one into another.

Rick Strahl: Yeah, the cycle plug-in is one of the coolest things to check out and it's something that you can plug into your own applications again with very, very little code.

Carl Franklin: Very cool.

Rick Strahl: Yeah, it does all sorts of very, very cool transitions from one image to another and there is about 20 or 30 different effects you can apply and some of them are just really especially for web applications.

Richard Campbell: I'm not thinking about photographs. I'm thinking about almost the iconification of elements of a business app.

Carl Franklin: Sure.

Rick Strahl: Yup.

Richard Campbell: There are just a lot different directions you can go with this. I just feel like we've got a whole development model here that here are essentially controls in the form of plug-in going against a fairly lightweight framework that just makes the DOM less of a bare.

Rick Strahl: Yeah.

Carl Franklin: Check out Farbtastic, the color picker.

Richard Campbell: Yeah, I saw that. It's beautiful.

Carl Franklin: Did you see that? It's a color picker so you have like a wheel where you have your primary colors all around in the wheel and then you have it mix with white and black, the standard, and of course then you end up with a color code, very cool.

Richard Campbell: All results to a color code. What is it about the DOM that has made things so difficult? I feel like we got everything we -- this is only wrapping up features we've always had, it's just we didn't know about it.

Rick Strahl: Yeah, if you really think about it, the biggest problem in client side development has always been the DOM. It's just that it is different in different browsers, and so if you look in the Internet Explorer model and you work also with Firefox and maybe Opera and Safari, all of a sudden you realized the vast majority of coding that you do to make things work is not to actually make things work but to make them work in all the different browsers and that is just a model that needs to go away and unfortunately the browser vendors aren't always stepping up to make that happen, right.

Richard Campbell: Well, and it isn't getting any better either. You just had a blog post where you're talking about how hover behavior changes in IE 7.

Rick Strahl: Yes, that's correct.

Richard Campbell: I'm just staggered, IE 7!

Rick Strahl: Yeah, it's crazy when we think about it. You know, as each new version come out, more and more differentiation occurs and even a tool like jQuery actually has to figure that out and what they can abstract. Eventually the code bits get so complex to actually try and figure out, well, for this particular operation, we need to do this and this and this and reams of code and execution cycles are wasted just trying to figure out what's right, but yeah, if you think about it, there's really no easy way to back out of it now. Well, we're going to be stuck with older browsers for some time to come and I don't know what the solution is really. The best solution that people have come up with is these libraries that can actually wrap these differences and give you a common object model which is a bit way to go, I guess.

Richard Campbell: It strikes me that the jQuery folks just happen to step back at the right time and come up with a good abstraction across all of those browsers so that we have a simple library for doing this.

Rick Strahl: That's absolutely it and then the other bonus is that it's really, really easy to build this plug-ins. So the model that they came up with is basically a simple function that you can implicate on a specific object to plug-in to that jQuery wrap set. So in order to create a plug-in, all you do is create a function actually and that function actually has been

add to access to that matched wrap set that allows you to get access to all those matched up elements and it's a super simple model so if you want to build a simple plug-in, it literally can be done with like two or three lines of code and you can of course take that a whole lot further like some of these really cool plug-ins like Farbtastic or the cycle plug-in to do really sophisticated stuff against all those matched object. It's really, really easy to extend and work with and that is also why there are literally almost a thousand plug-ins available now and they are growing on a daily basis. If there is anything that you need in terms of DOM development, chances are that somebody has built a plug-in for it already.

Richard Campbell: That's amazing. How do you find plug-ins?

Rick Strahl: If you go to the jQuery site, there is actually a plug-in section that you can look at and that's how you find most of the plug-ins that are available.

Richard Campbell: So, jquery.com is sort of the hub for all of these plug-ins?

Rick Strahl: Yeah. What's also very cool is the jQuery team is if you look at the documentation sometime, it has strict requirements on how plug-ins can be registered there. So you need to have documentation, you need to have samples, and they need to follow certain pattern. So all the samples in the jQuery documentation, actually all the reference material is laid out in exactly the same way, everything has a little sample with it, you can click on it, you can run it, you can view the source immediately and the same thing, the same requirements are made for the plug-ins and most of the plug-ins that you see actually follow those guidelines. So right off the top you have something that you can run on a website, take a look at and then even copy the code out and use it right away. It just follows kind of the rules that everything should be simple and easily discoverable and that is what I personally like about jQuery.

Richard Campbell: How in the world are these jQuery guys making a living?

Rick Strahl: I don't know.

Richard Campbell: Because the codes are all freely downloadable. There is a donate button.

Rick Strahl: Yeah. I think the jQuery guys are actually supported by a commercial company at this point.

Richard Campbell: Okay.

Rick Strahl: Somebody is sponsoring them.

Carl Franklin: That makes sense.

Rick Strahl: There's a donate button and I think they are actually getting a fund out of the donations from that as well.

Richard Campbell: I also imagine if they are selling, it's almost like the CSLA model too. There is a book and I got to imagine as soon as you get interested in this, you're going to buy the book.

Rick Strahl: Yeah. There are a number of cool books actually out there and you have them on your list there. The one that I really like is using jQuery and that's a great introduction to just how jQuery works even if you use jQuery before and you are fairly familiar with it, it's a great one to take a look at just because there are always new ways you can look at jQuery and what it does. There are so many different ways that you can apply it so looking at other people's jQuery code is one of the greatest way to learn of course.

Carl Franklin: You know, View Source has been our friend forever with JavaScript, hasn't it?

Rick Strahl: That is true too, yeah, but it's actually nice because for me personally, I still like looking in the book and actually have an explanation of what it does.

Richard Campbell: Yeah, why we did this. It's almost like a challenge to see how few lines you can write to get the results.

Rick Strahl: Actually, that's one of the things that maybe gets overdone a little bit so when you look at some of the jQuery samples, you have like one factor with 20 methods applied to it. It is a little crazy at times. So I guess you can break a code up if you choose to do that but ultimately, it's just code, it's just another line of code whether you apply a selector or a function to a command or whether you actually write it out explicitly. So one of the things you might want to think about is, well, I'm already using ASP.NET AJAX, do I need jQuery? First of all, as we mentioned earlier, you can certainly use jQuery in combination with ASP.NET AJAX but also if you choose to, you'll find that there is a ton of plug-ins available and some of these plug-ins might make a good ASP.NET controls. So for example, sometime ago, I actually took a date picker control that exists, the jQuery date picker from Marc Grabanski, and I decided this is a useful tool and I actually like to wrap that up into a control that I can reuse and simply drop onto a form and it turns out it's not terribly difficult to do that even though there is no explicit model to do that in ASP.NET given that jQuery is just a client library. So, it's relatively easy to do that and the advantages of

doing so is that you can actually wrap some of the related resources like spell sheets and images. You can also wrap the jQuery library itself into a resource so that when you actually use this component like the date picker, you can simply drag and drop it into a form, set up a few properties and you're done. So that's kind of a cool use of jQuery in combination with ASP.NET and have it behave almost like a native control even though you're using something that is not native to ASP.NET.

Richard Campbell: It's actually totally client side. I mean I know you're telling me this will work with ASP.NET AJAX and that makes sense to me in an existing application that it wouldn't break my existing ASP.NET AJAX implementation, but if I was starting from scratch, isn't it a sensible thing to just pick one or the other? Why would I do some processing on the server, some processing on the client?

Rick Strahl: Well, I think it's still easier to generate initial content on the server, so if you are building an end-user interface, typically not everything tends to be AJAX-based.

Richard Campbell: Right.

Rick Strahl: Your base content that you want to display and then you want to manipulate that content and that usually happens in AJAX. So for me personally, I very rarely build a page that's entirely client-driven but if rendering something on to the page and then additional content comes in, that is AJAX-driven. So you can certainly build applications that are pure AJAX but I find that it is often easier to do some service-side rendering to get the initial content on the web page. The other consideration is, of course, that if you deal with clients that have JavaScript disabled all together, you still have something that shows that might be useful enough for the users to do what they need to do and use AJAX as just the cream of the crop type of scenario.

Richard Campbell: Does anybody do that anymore? Actually turning JavaScript off?

Rick Strahl: Yeah, you'd be surprised actually. A lot of large companies still have policies that don't allow JavaScript.

Richard Campbell: That's craziness.

Carl Franklin: That's crazy.

Richard Campbell: You know, the other angle on this, I mean we're talking about the AJAX side of jQuery but that is not the only thing in here. It's only one sort of corner of everything that the jQuery does. I wonder what this combined with MVC would be like.



It's the one thing I see that's the challenge with MVC is making great clients because you've given up the web control.

Rick Strahl: Absolutely. jQuery is actually a great option because first of all ASP.NET AJAX really doesn't work all that well directly with MVC at this point until Microsoft's retools it a little bit because basically ASP.NET AJAX relies on the page model.

Carl Franklin: On the what model?

Rick Strahl: On the page model, the ASP.NET page model.

Carl Franklin: Oh.

Rick Strahl: So, you have some dependency there that if you want to do client side programming, there is no real easy way to callback. You can't put a script manager on an MVC page, for example. That just doesn't work because it is a page control. So jQuery is actually a great alternative if you're using a tool like MVC or you're using even some other development environment. The nice thing is that if you use jQuery, you can of course use it with any other tool if you decide to write a few PHP pages or whatever. It works with anything. It's completely tool-agnostic or survey-agnostic at least.

Richard Campbell: That is kind of amazing. You could use it as a bridge to cross multiple environments.

Rick Strahl: Absolutely, yeah.

Richard Campbell: I'm just boggled now. I'm thinking of all the possible directions I can take on an existing app or an alternative app.

Rick Strahl: Choice is always kind of mind-boggling and that it gives you sometimes too much choice and too many things that you can do all of a sudden so you're really never sure what's good for you or what's too much, kind of scary thing to think about.

Richard Campbell: Yeah, there are worse problems in this world.

Carl Franklin: True.

Rick Strahl: Yeah, that's true. Too much choice, that seems to be one of those things that keep popping up these days. You know, what do I do? Do I build a Silverlight application? Do I build a web client application with jQuery? Do I just use ASP.NET AJAX? Do I skip over AJAX altogether and just do server-side? You know, there are just a million different options that you have at this point.

Carl Franklin: Yeah.

Richard Campbell: Thinking architecturally, we've generally done this and ASP.NET is the culprit. ASP.NET made us a very server-centric way of doing development. It was figuring out what browser you were using back on the server and then sending out code that was specific to that browser.

Rick Strahl: Yeah.

Richard Campbell: Now, we're sort of pushing back to saying we're wasting a lot of horsepower on the browser's side but ASP.NET AJAX really didn't solve that problem.

Rick Strahl: No it didn't, but it's also, I think it is somewhat wrong-headed to just put the blame squarely on ASP.NET because if you really want to, you can absolutely build these applications with ASP.NET and even with the ASP.NET page model. I mean it is just a matter of how much do you want to rely on what the server-centric model provides but the capabilities to do so, to build pure client-centric applications is absolutely in the capability of the ASP.NET page model as it exists today. I mean I know a lot of people these days are really harking on the page model as guiding you to the long path...

Richard Campbell: The problem.

Carl Franklin: Yeah.

Rick Strahl: To which there is some truth but the truth also is that you can build really lightweight applications even with the page model if you choose to do so.

Richard Campbell: Yeah, this is mostly about control/abuse.

Rick Strahl: Absolutely.

Carl Franklin: You know, it's always been that especially with Microsoft tools, it's been the ability to do things the wrong way.

Richard Campbell: I do think that Microsoft sold us a bill of goods to some degree. That demos we got of ASP.NET forever where the draggy, dropy, pung, pung, here's your controls, you're done, thanks for playing.

Carl Franklin: Yeah, system.draggydropy.

Richard Campbell: I almost see MVC as sort of the strike against that, they've gone this far, the other way is possible, junking so much of that model

and then putting a lot more obligation on the developer to figure out the rest.

Rick Strahl: There are a lot of good things about that but there are also some very strong things that you lose and the biggest thing is that you just have no control over the page that gets rendered with MVC. So if you want to build the controls for example, there is really no good way to do that because there is no way to manipulate the page as a whole. This is especially true if you're talking about AJAX technology because there is no way to actually take that data or that strip code and inject it in various part of the page. If anything, it has only one spot in the page that you can inject anything into. So having a page model is very nice and the client script objects especially if you're dealing with JavaScript of any kind.

Richard Campbell: Yeah, I guess that's an interesting point. We didn't just sacrifice post backs when we go to the MVC; we give up the entire page context.

Rick Strahl: Yeah.

Richard Campbell: And that is actually a big cost actually for all of this.

Rick Strahl: I'm kind of curious to see how that works out for control developers because it seems to me it would be very difficult to build any sort of reusable model that's not just inserting at one spot in the page and that of course would control the developer's need. They need multiple entry points to be able to hook into and MVC, at the moment at least, doesn't do any of that.

Richard Campbell: Yeah. To be fair, MVC is still in beta. We may eventually see that reintroduced because it is recognized as too serious a loss.

Rick Strahl: Yeah, yeah and there is certainly something to be learned and if you want to build really client-centric applications that take advantage of a tool like jQuery where you're just calling back for the data on the server, then MVC makes a lot of sense actually because it is much easier to create this data-centric view that we turn data back to a client like jQuery via JSON callbacks and server calls. So it makes that portion of it much, much easier for sure.

Richard Campbell: I feel like we're starting to finally get a version of Studio that can develop a truly client-centric web client and it doesn't suck.

Rick Strahl: Yeah, I hope so.

Richard Campbell: I just find it very painful. As soon as you step outside of the boundaries of the sort of expected development model of ASP.NET in

even Studio 2005, you're struggling constantly, you might as well be using notepad.

Carl Franklin: Well, you know, it's hard to put down that RAD tool. I mean it is so nice and wraps everything up for you.

Rick Strahl: But you also got to remember that doing development when you're doing server-side development and client side development, just by its very nature, it's going to be a little bit harder than just doing server-side development. They are adding one more level to your application now and you're flipping just to one more document now to actually make the changes to the application logic. So there is one more level that you need to debug, that you need to test, that you need to actually step through and work through and it will take some time to get all of those tools right and I think Microsoft is slowly coming around to that. I think they have known it for awhile but its taking them sometime to get the tools in place. So we'll see where that all leads but my guess is that by the next version of Visual Studio, they will have that right.

Richard Campbell: It's nice to see them appreciating the model and seeing some great results like this, jQuery is just very compelling. I think they really figured something out here that is going to make our lives better.

Rick Strahl: I agree.

Carl Franklin: Rick, it's been great having you on the show. This is good stuff. I had not seen jQuery before we talked about it so I'm very excited to get my hands on it.

Rick Strahl: Okay, that sounds great. Well, that's what it's all about, right? Getting more people interested in this stuff.

Carl Franklin: Absolutely.

Rick Strahl: And turning a few more people into JavaScript believers. Remember, like a couple of weeks ago when we're doing that DevConnections thing, that you were asking people to raise their hands as to who wants to do JavaScript and like nobody raised their hands, I'm going to turn you guys into the guy that raises their hands.

Carl Franklin: Excellent. Do me a favor, hang ten, dude.

Rick Strahl: Yeah, not for a while, man. I'm stuck here in Switzerland. There ain't no waves here.

Carl Franklin: All right. We'll see you later. Thank you.



Rick Strahl: Okay, thanks guys.

Carl Franklin: All right, and thank you. We'll see you next time on .NET Rocks!

[Music]

Carl Franklin: .NET Rocks! is recorded and produced by PWOP Productions, providing professional audio, audio mastering, video, post production, and podcasting services, online at www.pwop.com. .NET Rocks! is a production of Franklins.Net, training developers to work smarter and offering custom onsite classes in Microsoft development technology with expert developers, online at www.franklins.net. For more .NET Rocks! episodes and to subscribe to the podcast feeds, go to our website at www.dotnetrocks.com.