



[HTTP://www.dotnetrocks.com](http://www.dotnetrocks.com)



Carl Franklin

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!



Richard Campbell

Text Transcript of Show #479

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



Hall and McWherter Test ASP.NET!

September 8, 2009

Our Sponsors



[HTTP://www.devexpress.com](http://www.devexpress.com)



[HTTP://www.code-magazine.com](http://www.code-magazine.com)



[HTTP://www.telerik.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! Stop straightening your bananas 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 #479, with guests Ben Hall and Jeff McWherter, recorded live Monday, August 24, 2009. *.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, *dnrTVstyle*. 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 Red Gate Software – essential tools for SQL Server, .NET, and Exchange. Support is also provided by CoDe Magazine, the leading independent magazine for .NET developers – online www.code-magazine.com. And now, the man who can check out anytime he likes, and frequently does, Carl Franklin.

Carl Franklin: Thank you very much and welcome back to *.NET Rocks!* It's Carl and Richard again.

Richard Campbell: Yes, we are.

Carl Franklin: We just won't go away.

Richard Campbell: [Laughing]

Carl Franklin: Oh, my God, is it hot out here.

Richard Campbell: Must be summertime.

Carl Franklin: Well, you know, it's been a mild summer but man, lately, [whistling]. The air conditioner at the studio and at my apartment decided to quit on the same day.

Richard Campbell: Nice.

Carl Franklin: It just went [screaming].

Richard Campbell: Always the hottest day.

Carl Franklin: Too hot. That's enough chitchat. Let's get into the show here. Better Know a Framework, coming at you.

[Music]

Richard Campbell: Awesome.

Carl Franklin: So Better Know a Framework, of course, where I shine a little light on a dark and dingy corner of the .NET framework and hopefully you'll learn a few things, or at least know where to go to learn a few things. Today I'm talking about System.IO.UnmanagedMemoryStream.

Richard Campbell: Oh.

Carl Franklin: Yeah. So this is access to an unmanaged block of memory from managed code. So typically, what you wanna do, like, if you have a resource file, you can load that up and map it into memory...

Richard Campbell: Right.

Carl Franklin: ...and then use the UnmanagedMemoryStream to go ahead and pull that data out without having, you know, if you've got a big file or something, without having to allocate and then de-allocate or garbage collect. So it avoids garbage collection, which is a nice little way to keep your profile up.

Richard Campbell: Especially if you're dealing with a really big file.

Carl Franklin: Exactly.

Richard Campbell: Okay.

Carl Franklin: So, you know, that's how you can do it. You avoid copying data into the garbage-collected heap, and you reduce allocations and improve your applications working set.

Richard Campbell: Nice.

Carl Franklin: So if you've got any big files and you wanna load it up, check out the UnmanagedMemoryStream. That's all I've got. Richard, you got an e-mail?

Richard Campbell: I do indeed, and it's about 472. That's the show we did with Steve Evans on IT for developers.

Carl Franklin: Oh yeah, great show.

Richard Campbell: He says, "Hi, guys. You made me laugh, cry, and worry..."

Carl Franklin: [Laughing]

Richard Campbell: "...while listening to your show 472. I'm currently the Web developer manager/senior developer for a new media agency here in the U.K. I was hired into this role, but my others are even worse than developer/network support, and it extends more and more because I was standing closest to the server when the last guy left."

Carl Franklin: Isn't that funny?

Richard Campbell: Funny how that works, yes.

Carl Franklin: Mmm.

Richard Campbell: "So even though I started out as a graphic designer, now I've also been a developer, IT support, network support, infrastructure manager, DBA, graphic designer, and more."

Carl Franklin: Yep. That's how it happens.

Richard Campbell: Yeah. "On another note, you talked about password security. For the development servers, there are strong passwords for both RDP and for the databases. They all use strong passwords, not passphrases."

Carl Franklin: Hmm.

Richard Campbell: "Do you consider this bad practice? All the staging and live servers have strong passwords, normally ten characters or more. We also use the same admin passwords for most things from Web servers, SQL servers, and mail servers because it saves the dev team/DBA team from having to remember multiple passwords. I'm worried that this is wrong too. Keep up the great shows. You always make me reevaluate my processes, which in the end keeps me and my team up-to-date. I intend to convert everyone to .NET Rocks! Cheers." From James Stuttgart.

Carl Franklin: P a s s w o r d s v e r s u s passphrases. Your verdict, sir?

Richard Campbell: I'm a passphrase guy, but that's mostly 'cause I think in phrases anyway and they're easier to remember. You know, they grab on. The problem with like, ten-, fifteen-character passwords is you invariably have to write them down.

Carl Franklin: Yeah.

Richard Campbell: But the other big whammy here that James mentioned is this idea that all these machines have the same password, so given one crack...

Carl Franklin: Right.

Richard Campbell: Or, you know, if the guy needs access to a Web server and suddenly finds out, "Hey, I can log in to all the databases too with this password," so...

Carl Franklin: Right, right, right.

Richard Campbell: It's definitely... There are issues there. I'm not the manic password rotator guy, like they... You know, there are folks out there that every thirty days you need to enter a new password and it can't be the same as the last fifteen passwords you've used.

Carl Franklin: Mm-hmm.

Richard Campbell: I just think that's a little too evil, so...

Carl Franklin: So you like the idea of finding multiple cryptic passphrases and using those just because you can remember them?

Richard Campbell: Yeah, but the phrases...you know the thing about...the thing that's great about phrases is that they're hard to hack, but they're easy for humans to remember.

Carl Franklin: Yeah.

Richard Campbell: So they don't have to be that cryptic.

Carl Franklin: Oh, okay.

Richard Campbell: It doesn't...yeah, you know, but they need to be about...

Carl Franklin: They can be a line from a song or something.

Richard Campbell: Yeah, or, you know, some phrase that matters to you, right? "There once was a moose named George" is a bear to hack. James, thanks so much for your e-mail, and if you've got questions, concerns, ideas from shows, criticisms, you name it, send us an e-mail -- dotnetrocks@franklins.net.

Carl Franklin: And with that, let's just introduce our guests. It's probably gonna be a good show. There's a lot of stuff to talk about so we'll get right into it. Our guests are Ben Hall and Jeff McWherter. Ben is a U.K.-based C# developer/tester with a strong passion for software development and loves writing code. Ben enjoys exploring different ways of testing software, including both manual and automated testing, focusing on the best ways to test different types of applications. He also loves developing Web apps using ASP.NET and Ruby on

Rails. He is a C# MVP and maintains a blog at blog.benhall.me.uk. Jeff McWherter is the Director of Simplicity at Web Ascender, a Web consulting firm based in Okemos, Michigan. His lifelong interest in programming began with a home computing magazine in 1983, which included an article about writing a game called Boa Alley in BASIC. Jeff currently lives in a farming community near Lansing, Michigan. When he is not in front of a computer, he enjoys rock- and ice-climbing, and road trips with his wife and two dogs. Jeff is the Program Director and founding member of the Greater Lansing Users for .NET (GLUG). That's a kind of a Swedish drink, isn't it? Christmas drink – glug. He is also an ASP.NET MVP and ASP Insider. Welcome guys.

Ben Hall: Hello

Jeff McWherter: Thanks for having us.

Carl Franklin: It's all about testing the Web. So who wrote the book? Is it Ben's book or is it Jeff's book, or did you both write it?

Jeff McWherter: We both wrote it.

Ben Hall: Yeah, it's a 50-50 split and we both had...both put in different parts of the book, wrote different sections of it.

Carl Franklin: And what's the name of the book?

Ben Hall: It is called *Testing ASP.NET Web Applications*, and it's published by Wrox.

Carl Franklin: Awesome. Well, where do we start? There's so many different kinds of testing when you're talking about, well, any software. How does it break down?

Jeff McWherter: Well, why don't we start like this? What do you think about when you think about Web testing, Carl?

Carl Franklin: Well I think about testing, I think about unit testing which has to happen no matter what kind of application you're writing. I think about functional testing. I think about usability testing. Of course I think about performance testing, I think about UI testing.

Jeff McWherter: Well, that's perfect because we've got chapters about all that different things, all those things are inside the book. And our first chapter that we start with is unit testing. So I'll turn that one over to Ben. Ben's the guy who wrote that chapter, and let's hear from Ben about unit testing.

Carl Franklin: Okay. Is that where you start? Unit testing?

Ben Hall: Well, in many different ways you can start testing an application. Some people like to go from kind of the exceptions testing, but... kind of requirements gathering and actually defining quite high-level exceptions tests beforehand. They... Most of them know where the end goal, of what they're aiming for. Some people like to do the fast ones, they like to kind of dive right in with the unit tests and start writing code and afterwards write higher-level tests. There are two different ways you can approach the problem.

Carl Franklin: Sure.

Richard Campbell: So when we're talking about unit-level tests, is this really the Web side of things or is this more about exercising the classes that Web pages call?

Ben Hall: So one of the advantages of using a framework and using the MVC pattern is that you can actually get quite into the core of the application and unit test the actual core functionality. And this means that you can cut out some of what actually having to load up the Web browser, having to load up some of the ASP.NET core runtime. And instead just focus on the actual core, and measure what your code is actually doing.

Carl Franklin: Now we know that ASP.NET MVC is really great for unit testing, but what about just regular ASP.NET applications? If you separate your concerns the way you're supposed to...

Ben Hall: Mm-hmm.

Carl Franklin: ..is there an issue? What's the issue?

Ben Hall: If you separate your concerns, kind of, if you follow a pattern similar to the model-view-presenter, the MVP pattern...

Carl Franklin: Mm-hmm.

Ben Hall: ...then you can definitely have good coverage of actual classic, kind of ASP.NET WebForms application in a similar way you would with MVP. The advantage which MVP brings, it...

Carl Franklin: MVC you mean.

Ben Hall: ...you get much more control.

Carl Franklin: MVC I think is what you mean, right?

Ben Hall: So there's two different types of patterns. There's the MVP pattern, basically you have your controller and you inject a few objects into the controller and then the controller can interact directly with the view. You can then interface to separate out the...program against the abstract.

Carl Franklin: Mm-hmm.

Ben Hall: With MVP... With MVC, the framework did more of this kind of legwork for you. So the controller and the view are more separated.

Carl Franklin: Right. So it comes down to MVC sort of forces you to separate your concerns, whereas if you're doing straight ASP.NET development with, you know, where all your goo is in a class somewhere outside of the UI, then it's up to you to do the separation and follow the rules.

Ben Hall: Yeah, definitely. And that's probably one of the main problems with ASP.NET how it was, in that it didn't really encourage you to separate your concerns.

Carl Franklin: Right.

Ben Hall: It's quite happy to sit there and allow you to put huge amounts of logic in your...

Carl Franklin: Button clicks.

Ben Hall: ...in your click event.

Carl Franklin: Yeah.

Ben Hall: And afterwards then, writing any sort of unit test around that, become increasingly difficult.

Carl Franklin: Right.

Ben Hall: With the MVC framework, they're kind of...they're not forcing you but they're definitely encouraging you to separate your concerns straight out the box and kind of encouraging best practices, and so you can actually unit test your code more effectively.

Carl Franklin: So tools for unit testing that you like to use?

Ben Hall: So in terms of frameworks, I definitely like to use the xUnit framework, which is on CodePlex, and I find it's got a nice syntax, it's really quick, it's really lightweight. NUnit – classic framework – does everything you need, does an excellent job of it. Rhino Mocks I use for marking, and then ReSharper and TestDriven.Net are helping you to write your code and helps you run your unit tests.

Carl Franklin: Okay. Is there more of the story about unit testing ASP.NET applications? Or if you're already using these tools to, let's say, you know, test non- ASP.NET apps, is there anything in particular?

Jeff McWherter: Well with the ASP.NET apps, one of the things in the book that we stress a lot about is you don't have to be using MVC to write good tests and be able to test your Web applications. Going back to the classic WebForms, the separation of concerns, we covered that a little bit, but just abstracting your logic and getting it out of the Button.Click event is just really important. And being a consultant, I see so many people's code, just different codebases constantly, and people are still writing code inside of button click events. If you guys take anything away from the show, take away that – don't write your logic in the Button.Click event – abstract it out. And separation of concerns is very important and will make your WebForms testable, and you can do things like test driven development with WebForms. It's a little bit more difficult than using MVC framework, but you still can use patterns like that for testing

Carl Franklin: This portion of *.NET Rocks!* is brought to you by our good friends at Telerik, without whom this show would not exist. No doubt you bump into testing tasks now and then in your work, and we can bet writing functional tests is not your favorite thing. It's difficult. It takes ages and the results could be dubious. Well, get ready to start liking it, thanks to Telerik. With the just-launched WebAii testing framework, building web automation tests is a breeze. Enjoy codebase automation of advance ASP.NET AJAX and Silverlight apps. Write a single test and have it executed against multiple browsers at once. Benefit from rich API LINQ support, integration with Visual Studio unit testing, NUnit, xUnit, and MbUnit – not to mention the free wrappers for a Telerik RadControls for ASP.NET AJAX and Silverlight as shipped with Telerik's new testing tool. Surely one of its best features, WebAii testing framework, which is developed by ArtOfTest, is absolutely free. If you're already hooked on WebAii testing framework, you can start using it right away. Go to www.telerik.com for more info. And hey, make sure you thank them for supporting *.NET Rocks!*

Carl Franklin: I'm remembering back to the days when I used to teach ASP.NET classes, and I think by the second half of day one, we were already out of the button click and writing classes – writing least one other class project, and then we would separate tiers from that, but...

Jeff McWherter: Right. And another thing about unit testing that a lot of developers don't realize that they do, they do it constantly. If you're opening up a

WebForms...or a Windows Forms application like all of us have done -- you open up a Windows form, you throw a Button.Click event on there, you put some test logic in there and make sure something works, that, you know, in a way is a unit test. So moving into...moving that into a framework like NUnit or something like that, making it reusable and being able to automate that, you know, you might as well keep that stuff around. It's definitely important.

Carl Franklin: All right. So from unit testing we go to...?

Jeff McWherter: I think we go...move into the automated user interface testing, more like functional testing.

Carl Franklin: Okay.

Jeff McWherter: There's a couple...couple camps on how to do that, and I'll talk about that camp first, that's the part that I wrote about, and that's using a framework to drive the Web browser. So using a framework, we talk about the Selenium and we also talk about WatiN. WatiN is framework derive from the Watir Framework which run on the Ruby side, and what it does is it integrates with a testing framework like NUnit and you can call the WatiN framework and drive a Web browser. You new-up a browser object, like an IE object, you throw some...you call some methods on it to navigate through a Web page, and then you use your testing framework to assert that certain things happened on the website, and then you're able to drive the Web browser and start making sure things show up on the screen. Selenium is another framework that you can do that sort of testing with. Selenium supports driving the browsers in Safari, Internet Explorer, Firefox, multiple browser support, and there's a whole stack of applications in there. The core of Selenium is written in Java. It turns away some .NET developers because they like to have everything written on the .NET framework for some reason, but it's just another one of those very powerful tools. One of the nice things about Selenium is the Selenium Recorder, which is a plug-in for Firefox and it'll record a browser session so it makes writing those tests a lot easier just having to, you know, you just record your browser session and it generates some code for you, and you can run, copy that over into your unit test and reproduce it.

Carl Franklin: Have you guys seen Telerik's testing tools, like the Web UI Test Studio or Testing Framework?

Jeff McWherter: I have not.

Ben Hall: I have come across them, and I haven't used them, but I have come across their tools.

Carl Franklin: I guess they're really good for AJAX and Silverlight, things like that. You do have the recording capabilities and simulating user interactions and things.

Jeff McWherter: Right, and Silverlight is an interesting one. Testing Silverlight and testing RIA apps is pretty difficult, testing not only for functional testing, testing for accessibility. That's a problem that we're facing in the community right now, is developing those tools, and it's great to see that Telerik is focusing on that and being able to provide testing capabilities for that.

Carl Franklin: I Shrinksterized that link if you want to check it out, listeners. It's shrinkster.com/18xd. So these other tools that you mentioned, are these free tools and...or not, and where we can we get those?

Jeff McWherter: Yeah. One of the great things about Ben and I, we both come from the open source world so a lot of the tools, most of the tools we cover in the book, are open source tools or free tools to download. You know, maybe we can include those links in the show notes and make sure that the listeners have that. But...

Carl Franklin: Absolutely.

Jeff McWherter: ...both Selenium and WatiN are free frameworks for downloading.

Carl Franklin: So that's cool. So it's good to know that if you buy the book, it's not requiring you to purchase anything. Do you...is there testing that you can do by yourself without... Can you write automated testing tools yourself? Is it, you know...I guess that's the question.

Ben Hall: You definitely can. Most frameworks, however, they've got huge amounts of effort and they help you switch, kind of fantastic fashion there. Writing your own is complex and takes a lot of time when the frameworks out there are quite mature now, especially on the .NET platform. There's...got a lot of support, lots of community around them. And actually it's quite easy to kinda just to jump in and get going with the existing tools that are out there.

Richard Campbell: And WatiN is just a... is literally just a wrapper over the top of browser object model, so it's pretty painless. You mean...I've done this myself by hand against IE, and it's lots of fuss depending on the version of the browser you're using.

Jeff McWherter: And writing your own framework, this is...living in Michigan with the auto industry and the Big Three, this is something that's

very common for me, and talking with Ben, it's not common for him, but there are a lot of companies that don't allow open source or just even open source tools like NUnit to be introduced in the company. So MSTest, and inside of Microsoft's Visual Studio Team Test Edition, there's the Web testing tools inside of there that provide functional testing for Web, driving the browser, and stuff like that. If your organization does not allow for open source solutions, there are solutions inside of Microsoft Visual Studio to perform these tasks.

Carl Franklin: Ah.

Richard Campbell: But...and the point here is you're actually writing code to drive the form. So you're actually automating the process of filling in a form and hitting "Submit," that kind of thing?

Jeff McWherter: Yeah, um...

Ben Hall: Yeah, pretty much. It's automating the user's kind of...how the user interacts with your website.

Carl Franklin: Right.

Ben Hall: That you have full control over the browser. You can do things like refresh the page, you can fill in the text boxes, hit the "Submit" button, go backwards, go forwards, just basically however the user would interact with your Web page, you can now automate – and by automated tests, round out, verify the Web page has actually worked as you expect.

Jeff McWherter: There are some problems around these sort of tests though, such as what happens if your user interface changes, names of things changes. In the book we offer some patterns on how to help you avoid those things, but they're considered fragile tests. If you change anything, your test could break and you're gonna have to go back and maintain the task. One of the things about testing is testing...it is an investment and you have to put money into that and hopefully your management sees value in that, but there definitely is an investment to keep the tests running correctly.

Carl Franklin: If I could jump back to unit testing for a second, do you guys recommend to developers that with...that they do test-first development, and is that recommended only when using MVC or can you do test-first –test-driven development without MVC framework?

Ben Hall: You can definitely do test-driven without the MVC framework and it's definitely something I would recommend to all developers no matter what kind of code they write in, whether it be ASP.NET or WinForms, or even WPF. I think test-

driven .NET just allows you to...it kind of changes your mindset about how you write code. Instead of just kind of bashing out code, kinda going straight into implementation, TDD – writing a test often –you actually stop, you think about the code, you think about what you're trying to achieve, and then you have your test to guide you and actually help you implement the actual code. And then once you're done, you have to test to make sure that it's passing and working as you actually expect.

Carl Franklin: And is it easier, obviously, using MVC? I think the answer is "yes."

Ben Hall: Yeah. MVC definitely gives you a lot more control...

Carl Franklin: Mm-hmm.

Ben Hall: ...over what you can test, and what you can actually test is a lot more easier to access. The object model is a lot simpler. It's also abstracted away from some of the core ASP.NET and the infrastructure such as the HTTP request. That's now been more abstracted away, making your test more isolated and actually more testable in a unit testing fashion. It's possible with WinForms and kind of the old style of doing it, but you just have to think a bit more, you have to kinda think about how you're structuring your application, and a bit more of a...you just have to put a little bit more effort into your actual architecture than you do with MVC.

Jeff McWherter: A good friend of mine, we were in a conversation once and we were talking about test-driven development and different types of testing methods, and he told me the type of testing that I do is "test-eventually development."

Richard Campbell: [Laughing]

Jeff McWherter: Test-driven development, writing your test first, I've definitely done it in the past. I'm working very hard to do that all the time, but honestly test-driven development is hard, and it's very hard for somebody to pick up a book and learn how to do test-driven development and do it correctly. Pairing with someone, doing pair programming, working with a mentor really helps you learn how to do that. Where I'm at in my career, as I've said, I try to do it as much as I can with the WinForms and MVC, especially in MVC I try to do it more, with Win Forms I try to do it, but I do practice test-eventually development. I think there's a support group out there for people like us.

Carl Franklin: [Laughing]

Richard Campbell: I just can't see how you would build a Web GUI in that sort of test-first model. You

really need to have the form laid down before you can construct the test for it.

Jeff McWherter: It definitely is very difficult constantly compiling and getting the errors and it not being in there, but as you just keep on abstracting things out and thinking about things I would like to do. And you know this brings up a good point: Test-driven development is a horrible name for the process of what you're doing. It should be "create a good design development first." Test-driven development is more about design and *how* it works rather than the actual testing. The unit tests that you get off of it, they're definitely a good side effect of it. And it's difficult to learn, but it's about design and learning how to get those objects up, get 'em built and then wire up that form and get 'em to save, and do the things that you...

Carl Franklin: Well, and you just...you just hit on something that made me think we might be giving the wrong impression. Yeah, wire up that form. So I mean so much of Web development is driven by the design of the page, isn't it? I mean, don't you... In a typical situation, you'd go to an artist, a Photoshop person who'd do a mock-up, and, you know, you'd sort of wanna have not necessarily all the UI with the buttons and the Silverlight or everything, but at least some sort of idea as to what the UI is so those forms would be written, and then you're using your test-driven development tools to build the logic layer. Isn't that true?

Jeff McWherter: In an ideal world, we have... In an ideal Web world, we have a designer mocking up a form and giving it to us and then we're able to take those elements and abstract them out into the things that we need. But most developers don't live in that ideal world, and there stuck mocking up those things themselves and doing it, and a lot of times the processes get fudged together, they don't do a mock-up. The mock up is just the .aspx page and it gets done like that and it's very difficult to work on those environments, in my opinion.

Carl Franklin: So you're saying it's difficult to work in an environment where you don't have some sort of design upfront?

Jeff McWherter: Correct. Design meaning not a software design, but more of a...

Carl Franklin: Graphic design.

Jeff McWherter: ...a UI design, a mock-up.

Carl Franklin: Yeah, yeah – a mock-up. I mean, there's just no end to the amount of Photoshop people that can design what a Web page should look like. Is it that difficult to, you know, is it that difficult to get someone like that involved in the process?

Jeff McWherter: In some environments, it absolutely is. I've seen lots of people who want to do all the work internally and I'm not a big fan of that. I say everybody is good at something, and if you're not good at doing the mock-ups, you know, try to find somebody who is, whether it's external, whether it's you know, some...maybe a secretary or somebody else at the company, they might not be a developer but they may have a good eye for design, good eye for color, they definitely can help with those sort of tasks.

Ben Hall: Yeah and I've worked with professional full-time usability people. And the difference they can actually bring to a project is quite immense. They see things in a completely different way. And how they structure that layout is just quite... it's difficult to believe how much impact they can make on a project compared to just trying to do it yourself, kind of in a creative fashion.

Richard Campbell: And it occurs to me that they are not...the mindsets are conflicting to a developer's mindset. You literally can't do both. You think one way or you think the other. I found when I've had designers hand me stuff, I'm looking at them like, "Do you know how hard this is going to be to code?" And then you sort of get that reality check of, "But it can be really easy for the guy to use."

Carl Franklin: This portion of *.NET Rocks!* is brought to you by the ANTS Memory Profiler from Red Gate Software. I'd like you to think for a minute about the project you're working on right now. Is your app showing signs of high memory usage? Do you need to regularly restart it because the performance seems to degrade overtime? If the answer to any of these questions is yes, then it's worth looking into a memory profiler like ANTS Memory Profiler from Red Gate Software. ANTS Memory Profile analyzes the memory usage of your app and provides detailed data, letting you to easily locate memory leaks fast. It only takes a few minutes to run, and it's so much easier to optimize your app when you know exactly where memory is going. ANTS Memory Profiler runs against both ASP.NET and Windows applications, and at \$495, you can't afford not to run it. To get your hands on your 14-day free trial, just go to shrinkster.com/19e0, that's one, nine, echo, zero, and don't forget to thank them for sponsoring *.NET Rocks!*

Jeff McWherter: So I think this leads us into a good segue to talk a little bit about accessibility testing and usability testing. Everybody... Different dev environments work different ways. Different dev shops have their ways of doing things. I've worked in places where I've been the sole designer, not a very good designer but nonetheless a designer. I've worked in a place where I was told to wrap everything around a div tag and the designers will come in and

they'll clean it up, just as you write, you know, write a little bit accessible code. To give you an idea of the current place that I work right now, the very first day I was here, two employees were having an argument about shades of gray. So it's a little bit different place, meaning that they're designers. It's a little bit different in any other place that I've worked, and it has been great to learn about that sort of thing, and I've definitely taken a real interest in learning about Web accessibility and looking at websites out there. There's a lot of websites out there that are not accessible to users.

Richard Campbell: So Jeff, after all this design work, you know, what is the phase of testing now? You've got a design that works, is it, you know, do you go for acceptability [sic]? What does that really mean?

Jeff McWherter: Accessibility is "are all users able to use your website?" So, an accessible website would be somebody with the disability, maybe the disability is mobility disability. They're not able to use a keyboard, or possibly the one that is most common that we always hear about is vision disability, it's someone is either blind or legally blind or can't really see the Web page very well. There's certain things that we can...certain ways to design our websites to make it much easier for these users to use the website, and that's the next type of testing that I definitely encourage users to start looking into. It's really hard. The tools out there, there aren't any real tools out there that you can click a button and say, "My site is accessible to everyone." The problem with accessibility is most developers see these compliance laws – Section 508, WCAG – and they see these things and they treat them as checklists: "I need to do this, I need to do that, I need to do that," when they should be thinking about the user instead. They should be putting resources into learning "how does a blind person use a website?", "how does somebody who is legally blind use a website?" They use a tool called JAWS, which is a screen reader or other screen readers on the market, and even just between someone who is legally blind and somebody who is fully blind, they use these screen readers different.

Carl Franklin: And actually Windows has built into it accessibility software that will read the screen to you and read browsers, you know, read Web pages to you.

Jeff McWherter: Right, definitely. Using these tools, everybody uses them differently and learning how these people use these tools is very important. Someone who is blind will crank up the speed on the screen readers and they'll go lightning fast, whereas somebody who is legally blind in my experience, they'll have it more on a normal pace but they'll bump up the font size and the resolution very high to look at

that and it definitely changes how users view your website.

Richard Campbell: All right, this is a very specialized area here and I don't know, I never think much about websites being accessible like that.

Carl Franklin: Yeah, I don't either.

Richard Campbell: Yeah, but obviously people have to deal with this. I really think that's a show unto itself, don't you?

Carl Franklin: How about this? How about Silverlight content? I mean, isn't Silverlight content, you know, especially text drawn with graphics instead of text? I mean it... There isn't... If you have a little Silverlight tool, unless you're in a text box or something, you just draw some text on a button or on a screen or something, that's not something you can copy or, you know, it's not text, it's graphics. Isn't that true?

Jeff McWherter: Right, that is true and Silverlight definitely poses lots of issues for accessibility. Silverlight 2.0 has added some stuff for accessibility to help make it more accessible. Flash and all those other RIA apps also, but we run into accessibility issues like that. And you guys mentioned that you don't know how it really fits into your application as it is right now. There actually have been some very large lawsuits about discrimination, about websites that were not accessible for people with disabilities...

Carl Franklin: Wow.

Jeff McWherter: ...and they'd have to go back and change it. Target was one of those a couple of years ago that they had to go back and make some modifications to the website. And the things that we're talking about to be accessible aren't really hard things. We as Web developers, you probably heard that you don't wanna use tables for layout. So tables are used for tabular data, and that's very confusing. Still to this day, a lot of people still wanna use tables for layout. A good way to think about this is, "If the data belongs in an Excel spreadsheet, then it's correct to use the table for tabular data." If you're using it for layout, it's definitely...you're gonna wanna find a different method like CSS...

Carl Franklin: Now why is that? Obviously, I mean, that's your opinion. That's not a hard, fast rule of law. I mean, you can put layout in tables and it works just fine.

Jeff McWherter: Well, screen readers definitely have a problem reading it though. It's the way that it reads on a screen, it's very difficult. It would read the

top first and then bump down a little bit and read...and it makes the experience for somebody using the screen reader very difficult to use.

Carl Franklin: Isn't that a function of the screen reader though, the direction in which it reads?

Jeff McWherter: It is the function of the screen reader and it could be blamed on the way that the screen reader's work, but these are the tools that we have to work with, and we always have to work around them and we have to work around these issues for years.

Richard Campbell: Well, it's one of the best cases I've heard for why you wouldn't use tables for layout, is that, you know, you use CSS for layout because there are tools out there that expect tables to be tabular data. All right, let's keep moving along. One of my personal favorites, 'cause it's certainly something that I've done a ton of, is performance testing or load testing. Is this a different kettle of fish from what we've talked about so far? What do you guys do for performance testing?

Jeff McWherter: Again, yeah, this is definitely a different...a whole different testing discipline that requires a whole different set of skillsets to learn about this. Lots of different tools on the market out there that are very expensive. LoadRunner, which I believe HP owns now, I always like to say a lot of these tools you have to sell your kidney to be able to afford to buy. Some of the options that we like to use is the...inside of Visual Studio Team Test edition there is some performance testing tools in there. You can...if you decided to develop your Web application and develop a suite of tests, the functional tests, those Web tests, you can use all those Web tests and do a performance test and a load test on your site. And also another one that's out there that's just a great tool that's been around for years is the WCAT program – the Web Capacity and Analyst Tools – which is included in the IIS resource kit, which is a way to script it, and you can do some neat things like replay IIS log files to get some data on the site about performance testing.

Ben Hall: I'd like to add to that. There's also a tool from Red Gate called ANTS Profiler.

Carl Franklin: Right. It's only like 400 bucks, right?

Ben Hall: Yeah, something like \$400, and that can test...performance-test your ASP.NET Web applications. There's actually the new version 4.0 that's got some really cool UI features that's right for identifying performance spots within your application very, very easy. It's got a nice graph at the top, and it's got some...and it can see the peaks in

kind of the CPU usage. It can just quickly identify where you need to actually dig in deeper and identify those problems.

Richard Campbell: Right. And there... Of course the difference here between the sort of profiling of the site, because if you're actually trying to load-test to find out how fast the site can go, sticking a profiler on a machine really messes with its ability to perform, because profiling is hard work itself, right?

Carl Franklin: Well, at least you can find where the bottlenecks are and then focus on testing those and amping those up.

Ben Hall: Exactly. It's better, or in my experience it is always better to try and identify the bottlenecks first.

Carl Franklin: Right.

Ben Hall: And try to eliminate as many of those as possible, and then afterwards consider performance and how much it can actually...where the actual limit of the application is. If you've got huge bottlenecks already in the application and you're trying the performance-test, then all you can do is hit those existing bottlenecks which you know about. If you can use a tool like ANTS Profiler to eliminate those bottlenecks just on your local dev machine, then when you actually get to your performance-testing stage, then you actually...the results are a lot more useful, like, more realistic about how your application might work.

Carl Franklin: Here's a question, and Richard you might have some insight into this because of your work with Strangeloop. You test your websites internally on your internal network, right? Does that give you a more accurate picture of how it's gonna perform on the Internet because you sort of raise the roof on the network bandwidth limit, that you sort of take that off the table?

Ben Hall: It kind of does. Personally, when I do it, I would perform some tests internally just more like sanity checks to make sure that application works, and then if it's going to be a public-facing website, I always try and upload it to a staging server...

Carl Franklin: Yeah.

Ben Hall: And ideally in the same colo location, and then actually use that to test how it will work for people actually on the Internet. Because you do get some oddities, and it's a very different beast to be running all your tests on a local server.

Richard Campbell: The biggest problem I have with internal testing is that it hides latency issues. So, you know, often as devs we really want to drill into our code and making our code faster, and then we actually get out on the wild you find out that the server processing time of your page represents five percent of the total page load time because you're...the guy is pulling down 60, 70 resource files with a 200 millisecond latency.

Carl Franklin: Right, so you...

Ben Hall: And that overwhelms everything else.

Carl Franklin: So really have to do both. You wanna...you wanna take the bandwidth thing off the table first, right, to test your code and how fast your code is working, then you've really gotta test it out in the wild. Now, are there companies out there that set up, you know, farms of machines on the Internet that you can automate to hit your website, to load-test it like under real conditions from multiple places around the world, you know, some multiple routes to the same site, that kind of thing?

Jeff McWherter: Well, there's definitely some stuff coming out with cloud computing out there. BrowserMob is one of the sites out there, and because setting up a test environment can be very expensive depending on the amount of load that you need to generate, so using a service based out in the cloud which is based off of Amazon EC3 services, BrowserMob will be able to be much cheaper for you to generate your load and maybe just do your testing when you need it.

Carl Franklin: BrowserMob, now that's interesting. I'm just looking at it now.

Richard Campbell: But guys, like LoadRunner, this is one of the services that they offer, right, for...

Jeff McWherter: Yup.

Richard Campbell: Priced accordingly of course, but...

Jeff McWherter: Of course.

Carl Franklin: Is that HP?

Richard Campbell: Yeah.

Jeff McWherter: Yeah, yeah.

Richard Campbell: Hewlett Packard owns LoadRunner.

Jeff McWherter: Another thing about

performance that we definitely missed talking about is the requirements. It's very important to determine what you're testing for before you even start testing. If you're writing an internal app that is only going to support 200 users, it's gonna be...your load test is going to be much different than trying to generate something, generate load such as the traffic that Digg or Slashdot would get. So some things I like to ask is just how many – you know always having the project stakeholders involved – asking them what they expect the traffic to be like, and most of the time they have no idea. Sometimes they do and when they do know is...that's when it's great to be able to say, "They wanna be able to record two million requests at this time, can we do this?" And when you get that answer, "Yes, the application can do that," it's always great to say, "Okay. Well, where does it fail?" So that's when the actual load test comes in, finding out where it does fail. That's important, too, to know that.

Carl Franklin: Hey, getting back to load testing, I found a company out here, LoadStorm, loadstorm.com, which generates tens of thousands of concurrent virtual users to simulate real users on your website or Web application. And it's...if you sign up for some "Breeze" account, I don't know what that is, you can run unlimited load tests with fifty concurrent users...v-users all month, every month at no charge. I'm not sure, but I think there's another level where you can get something like 0.1 cent per virtual user hour or something like that. I'm not sure exactly what it is because I just landed on the page, but there you go, loadstorm.com. Have you guys ever heard of that one?

Jeff McWherter: I have not heard of that one.

Ben Hall: I've not come across that, but there are definitely lots of companies which are taking advantage, so you just need to have a look around and experiment with the different services, see one which kind of suits you and suits the kind of traffic which you're planning to generate. There's lots of companies now that are taking advantage of the EC2, and authorization cloud. So there's definitely stuff there for you to investigate.

Carl Franklin: I found this on a list of tools at shrinkster.com/18xf, that's softwareqatest.com, and it's a website test tools and site management tools list and it's huge. You might wanna check that out.

Richard Campbell: At some point, these third party load-testing tools, you know, what is the difference between these and a denial of service attack?

Carl Franklin: [Laughing] That's true. Can you imagine signing up to hit somebody's website when they least expect it?

Richard Campbell: That's right, point it to somebody else's website.

Carl Franklin: [Laughing] Don't do this, people.

Jeff McWherter: [Laughing]

Richard Campbell: It's not a nice thing to do.

Carl Franklin: We're evil only in thought, not in deed.

Richard Campbell: [Laughing] All right guys, we're getting down towards the end of the show here. Have we missed anything? What about security?

Jeff McWherter: Yeah, absolutely, talking about the denial of service attacks there. Security testing is one of the things that I found in my experience a lot of developers really don't know a lot about and just really don't test for it just because, mainly because it's hard. We do have a chapter in the book about security testing. The problem with security testing is there's hundreds of books out there. There's just so many books about it. It's such a wide topic. You can do show after show after show about the different types of security testing out there. So what we did was we took the approach, there's a project out there, the OWASP project – the Open Web Application Security Project – and they publish every two years a top 10 vulnerability report and these are the top 10 vulnerabilities that web developers should be out there and talking about.

Carl Franklin: Hey, did we miss integration testing? What's that all about?

Ben Hall: So integration testing is basically your unit tests you want to be quite isolated and you don't want them interacting with external parties and external resources. Your unit tests should be isolated, they should be fast, and they should pass every single time. You wanna make sure that they're covering your logic and that you see the green bar to make sure that all your business logic is working perfectly every single time. If you see your unit tests fail, then you know there's something wrong urgently. With integration tests, this covers your...the code which interacts with your external parties such as a database or maybe a Web service, an e-mail server. Now, these have issues interacting with these types of services have lots of issues around them because they need extra configuration, they take a lot longer to run, they might need some additional files, or some additional support in order to make the tests actually pass. And also they might just be down. For example, if you're accessing a database across the network and that network goes down, then all your tests are going to start to fail and this is where integration

testing comes in. We like to...generally it's best to separate the two – separate your unit tests and separate your integration tests – just to keep the boundaries kind of more clean and so you can run independently.

Carl Franklin: Okay. And getting back to your book, can we find the book on Amazon? And give the name of it one more time.

Ben Hall: You can find the book on Amazon. It's available to preorder now and it's called *Testing ASP.NET Web Applications*.

Carl Franklin: Excellent. Ben, Jeff, thank you very much. It's been enlightening.

Jeff McWherter: Thanks guys.

Ben Hall: Thanks guys.

Carl Franklin: All right, keep doing what you do, and you out there, keep listening to *.NET Rocks!* and we'll see you next time.

[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.