



Hansel minutes

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

## Text transcript of show # 53

February 28, 2007



### Hiring and Interviewing Engineers

Scott and Carl talk about the "FizzBuzz" test and try to come up with practical techniques for hiring engineers and technical folks.

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



### Our Sponsors

 <p><b>telerik</b> <i>deliver more than expected</i></p>	 <p><b>NET</b> DEVELOPER'S JOURNAL <a href="http://dotnet.con-sys.com">http://dotnet.con-sys.com</a></p>
---	--





**Lawrence Ryan:** From [hanselminutes.com](http://hanselminutes.com), it's Hanselminutes, a weekly discussion with Web developer and technologist, Scott Hanselman, hosted by Carl Franklin. This is Lawrence Ryan, announcing Show #53, recorded Wednesday, February 28<sup>th</sup>, 2007. Support for Hanselminutes is provided by telerik r.a.d. controls, the most comprehensive suite of components for Windows Forms and ASP.NET Web Applications, online at [www.telerik.com](http://www.telerik.com). Support is also provided by .NET Developer's Journal - The World's Leading .NET Developer Magazine, online at [www.sys-con.com](http://www.sys-con.com). In this episode, Scott and Carl discuss techniques for hiring engineers.

**Carl Franklin:** Hi this is Carl Franklin, you are listening to Hanselminutes from [Hanselminutes.com](http://Hanselminutes.com). Scott and I are here this week to talk about hiring and interviewing engineers.

**Scott Hanselman:** Yeah, this week there was kind of this crazy meme that was running around the Web. I don't know if you heard about this, the FizzBuzz problem?

**Carl Franklin:** No, I didn't hear about this.

**Scott Hanselman:** Oh, you were too busy making podcasts. Basically, a FizzBuzz question is a question that you would ask a programmer you are about to interview, and the question is this -- now, here is an example. Write a program that prints the numbers from 1 to 100, but for multiples of 3 print 'Fizz', and for multiples of 5 print 'Buzz', for numbers that are multiples of both 3 and 5, print 'FizzBuzz'.

**Carl Franklin:** Okay.

**Scott Hanselman:** Sounds pretty straightforward.

**Carl Franklin:** Sure.

**Scott Hanselman:** Apparently, there is a great, great number of people when asked this question, cannot do it.

**Carl Franklin:** Really!

**Scott Hanselman:** They get lost, they get stuck, and there were a number of folks talking about this; you can take a look up at [shrinkster.com/mgy](http://shrinkster.com/mgy), and Jeff Atwood of Coding Horror blogged about this and really kind of -- it is an older article...

**Carl Franklin:** Why Can't Programmers Program?

**Scott Hanselman:** Well, he is calling it 'FizzBuzz: the Programmer's Stairway to Heaven', because it is like walking into a guitar center and saying...

**Carl Franklin:** I have already written it in my head.

**Scott Hanselman:** Exactly. So, here is the issue; when he put this up there, if you take a look at his site right now as of this recording, he has got 350 comments. Here is the really sad part; the vast majority of those are people solving the problem.

**Carl Franklin:** Wow!

**Scott Hanselman:** Now, you couldn't come up with a much simpler problem than that. The point is not the problem, right? The point is, if you can't do it. So, it is weird that people felt the need to spit that out, but whatever, who are programmers; you tell them that they can't do something -- like he says in his post about it being the Programmers Stairway to Heaven, you walk into a Guitar Center and you say, "I don't think most guitarists can play Stairway," suddenly everyone is going to start playing it, which is not the point of the discussion. So, this has kind of turned into a larger question about how do you interview folks? There is the famous Microsoft interview question about why are manhole covers round, which could be a question, or as an example of critical thinking, and could just be really obscure; because with a lot of these questions, once you learn it, you learn it. Now, you are clever because you remember it. How do you interview people without it becoming an issue of trivia? Well, there was an article at Artima Developer by a guy named Bill Venners who says that, "Here are some ideas about how to interview people." First, you want to explore an area of their expertise. You want to find out what is it that this person is good at.

**Carl Franklin:** You mean, this is before the interview? - Do a little research into what they claim to know?

**Scott Hanselman:** No. Well, this could be a phone interview, this could be the in-person interview, it could be any of those things.

**Carl Franklin:** But you say, learn a little bit about their area of expertise; do you mean before the interview obviously?

**Scott Hanselman:** Certainly; I assume you've looked at their resume and you know that they are good at something; my point is, is that rather than having a list of stuff you want them to do -



like, here is the 50 acronyms that I need you to have - EDO or WCF, dah, dah, dah - what is it that they are really awesome at? When you talk to someone, and they think that they are good at something - they feel they have an area of expertise, you may not necessarily be into them for that area, but they are stoked about it, they are good at it; what it is like talking to this person as they explain something that they are really good at?

(00:04:57)

I don't know much about audio; you are an expert in audio though. So, I could find out about your personality by exploring this aspect of your personality. Tell me all about this area of your expertise. By the way, as a small aside, there is a really funny book and a really great audio book by a guy named John Hodgman; he is the PC in the Mac/PC commercials; he is also a comedian, he has got a book called, "*The Areas of My Expertise*"; it is a great book to have.

**Carl Franklin:** Oh, I fancy who that guy is; I heard an interview with him on 'Whad'Ya Know?' with Michael Feldman.

**Scott Hanselman:** Yeah, he is brilliant.

**Carl Franklin:** And of course, he claims to have expertise in everything.

**Scott Hanselman:** Exactly; and that is why the book is so funny.

**Carl Franklin:** That is why it is funny - yeah.

**Scott Hanselman:** So, another idea is to -- and this is all from that Artima Article -- 'Have Them Critique Something', right? Say, what is it that you don't like about C#, or tell me why you think Garbage Collection is just a fad? So then you can find out are they the kind of developer who is overly critical; or they immediately start dogpiling and kind of, crapping on a particular technology, then you can find out, are they kind of a fair-weather programmer? Then of course, there is the standard, 'Let them solve a problem' type stuff, and we can give them all sorts of problems, and I will talk to you about a problem that I recently gave a guy that I interviewed.

**Carl Franklin:** And the problem with giving problems is, you sort of have to give unique problems every time right?

**Scott Hanselman:** Exactly.

**Carl Franklin:** I mean, you can't just open up a book of 50 great problems to ask, because they can just study the book.

**Scott Hanselman:** Right. Well, and the issue with asking someone to solve a problem also is, is this someone who doesn't work well under pressure? Now, some people argue that, "Well, we are under pressure in our jobs, so you should be asked to do something under pressure." Well, other people feel that an interview is a lot like a test, and the test is kind of an artificial thing, like it was in school - quick, perform - tell a joke. So, asking someone to design or write code on the spot, there are many people whose personality types simply will not allow them to do that. One interesting guy named Bruce Eckel on the Artima Articles said, he likes to ask candidates to create an object model of a chicken.

**Carl Franklin:** Now, that is interesting.

**Scott Hanselman:** ... because everyone knows what a chicken is, and it kind of pushes people away from the details of the computer, and let's you know if they can think about the big picture.

**Carl Franklin:** Well, yeah, and if they can translate real world things into objects.

**Scott Hanselman:** Right, exactly. Dave Thomas on that same article says, he looks for people with curiosity, and presents problems, not puzzles - so that was pretty cool. I also ask people about what blogs they read, what books they read, how do they become lifelong learners...

**Carl Franklin:** Yeah, very important.

**Scott Hanselman:** ...because it is not that you want them to be people who are necessarily working on Open Source projects at night, but they should be enthused; they should be stoked about whatever they are working on.

**Carl Franklin:** What do you say, if they say, "I get all my information from reading Scott Hanselman's blog and listening to Hanselminutes?"

**Scott Hanselman:** Yeah; then the interview is over and they are immediately asked to leave.

**Carl Franklin:** Sorry, but you just lost dude; due to excessive suck up points.

**Scott Hanselman:** Yeah, we can't have that. No, I mean, I am just on the same journey anyway right? Oh, actually here is a small tangent, but a kind of a funny programmer thing; this is up at



[shrinkster/mh1](#) - programmer jokes - when they go in heaven, St. Peter is sitting at a desk, he is checking people in. Peter says, "Welcome to Heaven. What is your programming language?" The guy at the front says, "Smalltalk". Peter says, "All right Smalltalk, Room 33. Please be very quiet as you pass Room Six." Process has come repeating itself, "Welcome to Heaven. Programming language?" "VB". "Oh, Room 17, but do be very quiet as you pass Room Six." "What is your programming language?" "C#". "Room 54. Do be very quiet as you pass Room Six." "Why do you keep telling us to be quiet as we pass Room Six?" "Ah, the Ruby on Rails people are in Room Six, and they think they're the only ones here."

**Carl Franklin:** Now, that's an adoption of another just regular old joke.

**Scott Hanselman:** Yeah, it is the baptist joke. But, you want to find out when you are interviewing people, 'Are they religious zealots?', because not every company runs things the same. We have chosen that C# is our language, but even if someone doesn't know C#, if they are a VB.NET person, they come in, are they willing to learn our way of thinking, whether it be programming language, religious zealotry or the way we do testing or the way we do Continuous Integration? It is a real challenge to find someone who will challenge the status quo without necessarily squashing or stepping all over the status quo; because you don't want to incite controversy, but you do want to incite kind of, commentary and discussion. So, is the person's personality one that will integrate with your company? - You don't want them just to fall in line, unless you are explicitly hiring for 'Wanted: programmer who will just fall in line and do what I say'.

(00:09:55)

One of the other things that we found looking through my comments here is that people don't like trivia tests. A while back, I posted that list of what I thought developers should be able to -- .NET developers could know. I think that is at [shrinkster/mh2](#); it is just a brainstormed list of all the different things I thought the average .NET Dev should know. Some of them, like what is a difference between a Thread and a Process, people felt were very basic, while others felt that that wasn't necessary anymore because we were working at a much higher level of abstraction.

**Carl Franklin:** I would beg to differ.

**Scott Hanselman:** Really?

**Carl Franklin:** Sure. I mean, you have to know what a thread is, especially in this day and age, if you are going to write code, because so much of your code that you are going to be writing in the next 20 years is going to be multithreaded.

**Scott Hanselman:** Well, one guy said -- in one of my comments on my -- when I was weighing in about this FizzBuzz thing is that "He knows what Recursion is and he did it in university, but in his experience working for ten years or so, he just hasn't needed to use Recursion to solve a business problem."

**Carl Franklin:** Well, that in particular - that case is true.

**Scott Hanselman:** Some people would find that hard to believe.

**Carl Franklin:** Well, because you are really not writing a lot of code at that level; I mean the things that Recursion is good at, is iterating through lists like lists of directories, and putting in things like schedules into calendars and things like that, and a lot of that stuff just -- that is at the tool level.

**Scott Hanselman:** Oftentimes, iteration can cover up Recursion -- or do Recursion just as well as you need to. Frans Bouma made that comment on the blog as well, that you can't solve virtually any problem without Recursion. So, maybe that wasn't a good example. I think that the real question is...

**Carl Franklin:** The difference between a Process and a Thread is little different I think though.

**Scott Hanselman:** Certainly. And when you are looking at 'patterns type' questions, like, if there is a known Computer Science thing whether it be Recursion or lists or whatever, do you know when to recognize that you need it; whether you have used it or not, or whether you have vast experience in it, is very different from, is it the time - is this the time to use that? Now, there is a lot of controversy about whether or not we should actually have people code at the interview? - Seems to me that if you are hiring someone to sing, they should be singing at the audition.

**Carl Franklin:** I agree.

**Scott Hanselman:** I don't want to find that they can't sing on opening day; but a lot of people, a lot of companies will hire somebody, put them through training, teach them about their product - they may be at the company two or three weeks before they actually have learnt -- have actually written some code rather. Do you think it is



reasonable to pull a laptop out and say, "Hey, get this out of the database." Select "star" from authors; I will be back in an hour.

**Carl Franklin:** I would absolutely do; in fact, I would try to make up some sort of FizzBuzz thing that is simple, that obviously isn't all over the Internet like that it is now; I would certainly do that, I mean, especially given that the results of that test, which I agree, is pretty disturbing.

**Carl Franklin:** There was a guy named Mark Freedman on my blog who said that they give the candidate 45 minutes; they give him access to the Internet, and they really do expect them to look stuff up, and they stick them in a room; they want to find out if they work well under pressure, what do they do when they get stuck, and then, how do they communicate their issues, and if they even try to communicate issues; do they stay in the room or they come out and say, "I have a question."

**Carl Franklin:** It is a great idea.

**Scott Hanselman:** Yeah, I think it is a great idea; I think I want to implement more of that kind of thing. We have people write some code on the board but bringing it on a laptop I think is a good idea. Now, one point was that, many developers will never do new development; they will just do maintenance development.

**Carl Franklin:** True.

**Scott Hanselman:** Greenfield File, New type stuff they are saying is very rare. So, this particular gentleman Mark Freedman also gives them an existing ASP.NET application; and in his case they have both VB and C# applications that they will give them. And they'll say, "Here is a bug or two. I want you to go fix them." Let them go through the code, so it teaches them how to read code. How do you become a really great writer of literature?

**Carl Franklin:** You read all the time.

**Scott Hanselman:** You read - or you can't. I have learnt a lot from reading Open Source; where else can you find just a fantastic amount of source code of questionable value and quality? But you don't know until you read it, until you actually take a look and find out, is this the kind of programmer I want to be? That is the kind of stuff that I think about when I see source out in the wild. The other thing is to watch for the kinds of requirements - questions that they ask. If you simply say, 'Write FizzBuzz', do they say, "Well, is this a console app or is it on the Web?" "Is there a performance requirement? Is this working

on Linux?" Now, here is another one - supportability. Who is going to support this application later? Things like that show that the...

(00:15:02)

**Carl Franklin:** I would be concerned if they asked me questions that they could easily Google. I mean that is like, the warning bell should be going off.

**Scott Hanselman:** Yeah, absolutely.

**Carl Franklin:** Even though I know you don't like to work blue but this is a URL; there is a great Website called <http://www.justgoogleit.com>, which you can send to people when they send you an email that says, "What is that?" So -- anyway.

**Scott Hanselman:** Yeah, I definitely do not want someone to come in and have me Google for them. So, that gets into the "What kind of a problem solver are they?" - Because you want them to self start; you want them to work hard, but you don't want them to suffer. We are a team, so you want to understand that working as a group involves walking around; it involves talking to people. You can't just slide pizza under the door and figure that people are going to solve that problem.

**Carl Franklin:** Here is a question; what if you notice something like that right off the bat, and everything else about them seems to be pretty satisfactory, and yet like, they have this one bad habit of, let's say, not Googling for information that's easily Googable? Do you say, "Well, they could change?"

**Scott Hanselman:** I have a tendency to be analogy boy; and I know I take it a little too far, but I'll give you two analogies, one on my blog and here is a fresh one. So, the first one was what I said on the blog, which was, "When you are trying to put together a basketball team, you want to hire tall people." So, I want tall programmers; I don't really care -- I can't teach height.

**Carl Franklin:** Somebody could learn to use Google, when they have a dumb question.

**Scott Hanselman:** They can learn VB; they can learn Ruby; they can learn whatever, they can learn our product, they can learn our processes, but I can't teach them to be tall; I can't teach they are a mental giant or not, and when I say mental giant, I am not talking about like, SAT scores here, I am talking about, "They either get it, or they don't." Now, that said, there are always the -



kind of the rudies and those kind of - I am 5'3 but I can still dunk type of people; so these are all guidelines. The second cheesy analogy, when the question was, "Do you think someone can get better?" Well, when people get married, and then they get divorced, and then you hear about them actually marrying the person that they divorced...

**Carl Franklin:** Yeah. I heard Larry Miller gave a good analogy of that. He says, "It's like opening up the door to the fridge, taking the milk, go, 'oh, this milk is sour; well, maybe tomorrow it would be fresh' - and you put it back and you close the door."

**Scott Hanselman:** Exactly; and that totally applies from a developer's perspective. Is it your job to hire the best person for the team, or is it your job to teach someone? Well, if you are hiring an internal or junior developer then their basic development as a programmer is your best interest; but if someone has a title of senior staff engineer, while, their development is important to me; they are being hired to turn to build a particular crank. It's really not my job to give them the basics. There is a certain level you expect - and this is why that FizzBuzz thing, while people have felt the need to solve it and said that it's a silly question, I think the best way to phrase it is that it doesn't tell you if they are a good programmer, but it sure tells you if they are a lousy one - if you can't solve a problem like that -- and that is just one example; you could certainly come up with your own example -- then you might not be a particularly, initially talented programmer to do the basics. The idea is to lower the bar to kind of, keep the chaff from coming in the door and then get down to the real programming questions. So, here is -- let me interview you for a second. This is just one example, so don't -- nobody write in and say, that was a crap example --- I mean, maybe it was a crap example - it's just one. So, I'd like to talk about debugging problems that we've had, and walk people through it, like they are a member of the team and we are trying to solve the problem for the first time.

**Carl Franklin:** Okay.

**Scott Hanselman:** So, it's kind of like CSI; it's like Crime Scene Investigation .NET. Okay, so I am going to hire you Carl, and I'll even just say, "Hi, how are you", and then I'll immediately start talking about the problem. And it's kind of like, "Ooh wow", and if they light up, then it's like, "Hey, there is a problem".

**Carl Franklin:** Okay.

**Scott Hanselman:** So, I have got an ASP application and an ASP.NET application - it's the same app. So, one virtual directory; there is classic ASP, and there is ASP.NET in the same app. It's a .NET 1.1 application, so I have gone into IIS and I have told it, this should be 1.1.

(00:19:58)

What happens is, if someone hits the ASP.NET page first, then it loads up .NET 1.1 and everything works great; but if someone hits the ASP page first, somehow the whole virtual directory turns into .NET 2.0; it gets stuck that way, and then if I go to another page after that, it's too late; .NET 2.0 is the de-facto framework for that virtual directory. Where do we start to solve a problem like that?

**Carl Franklin:** So, just to clarify the problem; the problem is that - now, the first place they go being this session -- in that session, or since the application came up?

**Scott Hanselman:** Yeah. So someone opens their browser, it's never come up before - exactly, good question - not this session, but it's the first time we've hit the app.

**Carl Franklin:** Okay, so it's the beginning of the application cycle?

**Scott Hanselman:** Right, at the point where the framework gets loaded.

**Carl Franklin:** Okay. And if you go to an ASP page first in that virtual directory, if then you go to an ASP.NET page, the 2.0 framework is going to load.

**Scott Hanselman:** Ah! If they hit the classic ASP page.

**Carl Franklin:** Just what I am saying.

**Scott Hanselman:** Yeah. If they hit the classic ASP page first, and only -- .NET 2.0 gets loaded.

**Carl Franklin:** That is what I mean.

**Scott Hanselman:** They don't even have to hit an ASP.NET page.

**Carl Franklin:** But if they go to an ASP.NET page, 2.0 is going to process it.

**Scott Hanselman:** Yes, exactly.

**Carl Franklin:** Okay. But if they go to an ASP.NET page first, 1.1 is going to load.



**Scott Hanselman:** Exactly.

**Carl Franklin:** I would take a look at the Internet Services Manager on the machine level, not at the application level. Is this a virtual directory?

**Scott Hanselman:** This is a virtual directory set up as an application.

**Carl Franklin:** Alright, then I would look at the application level first.

**Scott Hanselman:** It's set to 1.1.

**Carl Franklin:** And it's set to 1.1.

**Scott Hanselman:** But 2.0 of course, is on the machine.

**Carl Franklin:** Of course, 2.0 is on the machine.

**Scott Hanselman:** Let's say that there are no .NET 2.0 applications configured anywhere in IIS.

**Carl Franklin:** Interesting; okay, that's key. Now the thing that I would want to do is, I would want to see what's going on under the hood. So, I would probably look at some performance counters in .NET 1.0 and .NET 1.1, just to take look and see what's happening.

**Scott Hanselman:** 2.0 and 1.1.

**Carl Franklin:** 2.0 and 1.1.

**Scott Hanselman:** Okay. So, here is one additional information; there is a COM object that we are using on the classic ASP.NET page. Okay?

**Carl Franklin:** Ah! Sure.

**Scott Hanselman:** That COM object was written in .NET. So, keep going.

**Carl Franklin:** Was that COM object written in .NET 2.0?

**Scott Hanselman:** No, it was written in .NET 1.1.

**Carl Franklin:** Written in .NET 1.0.

**Scott Hanselman:** The ASP page doesn't know it's .NET of course; it's just calling the Runtime Callable Wrapper.

**Carl Franklin:** So, it's the COM interop layer, the thing that is loading 2.0, because that's the most current version of .NET on the machine.

**Scott Hanselman:** Exactly. That is exactly what's happening. Even if you go in the registry and lift -- is that the sound of cheers?

**Carl Franklin:** That's the sound of cheers, yeah.

**Scott Hanselman:** Well, we are not done yet, we haven't solved it; we know why the guy was shot but we don't know -- I mean, we know that the guy was shot, we don't know why he was shot.

**Carl Franklin:** COM Interop layer is a good place to consider when weird things happen.

**Scott Hanselman:** Okay. So, I go into the registry and I look at the ProgID for that COM object, and I see that it's actually marked to say .NET 1.1; it says that in the registry, it has the version number and everything but it's just not happening that way; what's the deal?

**Carl Franklin:** Alright. So let's look at the Machine.config file, and perhaps that object assembly is getting redirected somewhere?

**Scott Hanselman:** Okay - good idea - assembly redirect. So there is no assembly redirects and there is no GACKing going on.

**Carl Franklin:** There is nothing in the GAC; there is no assembly redirects going on. Is this COM object calling another COM object?

**Scott Hanselman:** No, it's all by itself.

**Carl Franklin:** It's not calling another .NET Object?

**Scott Hanselman:** Nope, it's all by itself. So, you've found the issue; the issue is that the COM object is loading .NET - and I'll give you one more information; it appears that, that registry key is completely ignored, because any time you make a COM object on a Windows system, you'll get the latest version of the framework, if the framework isn't already loaded - that's just how it works. This is why we don't write Explorer plug-ins and write click menus in .NET.

**Carl Franklin:** So the problem is that it's not looking in the registry to make sure that it's loading the right version of the framework. That's the real problem.

**Scott Hanselman:** And this is just a feature; it's a 'By Design' - it picks the latest version. So whoever gets in first, is in. So let's think about that here, it's the solving of the problem.

**Carl Franklin:** Alright. So, that's what's happening. So now let's solve the problem -



whoever gets in first. So in other words, how can you force the COM object to load framework 1.1.?

**Scott Hanselman:** Exactly. So now you have identified the problem. How do I solve that?

(00:25:02)

**Carl Franklin:** How do we force it to load Framework 1.1? Well -- and you built the COM object in Visual Studio?

**Scott Hanselman:** Right.

**Carl Franklin:** So the COM object is a .NET object that's being called with a COM Callable Wrapper through ASP.

**Scott Hanselman:** Right, we call it Regasm on the --

**Carl Franklin:** Yeah, Regasm; exactly.

**Scott Hanselman:** So, what are some opportunities in IIS for us to modify its behavior?

**Carl Franklin:** Let's see. I got a couple of things going in my mind; one thing is that I would want to try -- because I haven't had to solve this problem before.

**Scott Hanselman:** Exactly. Which is why it is such a good question for an interview; of course, now that we have broadcast it, it becomes a crappy question. This is an example of the kind of stuff we are doing.

**Carl Franklin:** It's a great exercise. So the first thing I would like to do is, I would like to load up the source to the object that I am creating in .NET in Visual Studio.

**Scott Hanselman:** Let's say it's just 'Hello World' - it's one function 'Hello World'. There is nothing there.

**Carl Franklin:** Okay, is it something that's compiled into a DLL beforehand or it just -- of course it is; it's not a VB file?

**Scott Hanselman:** It is foo.dll - Regasm foo.dll - Server.CreateObject("Foo.Foo").

**Carl Franklin:** And when it was created - when it was compiled -- I guess at the command line, because you are not a Visual Studio guy -- it was compiled with 1.1.

**Scott Hanselman:** Absolutely compiled with 1.1.

**Carl Franklin:** Absolutely compiled.

**Scott Hanselman:** And we talked about how the COM...

**Carl Franklin:** There is no way to redirect it...

**Scott Hanselman:** No.

**Carl Franklin:** And it doesn't access the registry. Before you said, you had put a thing in the registry to tell it to load 1.1 - but it wasn't working?

**Scott Hanselman:** When you do a Regasm, they put in there which MS CoreLib - depending on what it was built with, they put in the assembly qualified name of the MS CoreLib and that was a hint to load the framework.

**Carl Franklin:** So pretty much I'm convinced now that it's not at the .NET Assembly level.

**Scott Hanselman:** Yeah, there is nothing in the assembly that's causing it.

**Carl Franklin:** It is between ASP and the COM Callable Wrapper - that's what I think.

**Scott Hanselman:** So, if we assume that it is something about .NET - and I can't change .NET's behavior, nor I can I get Microsoft to fix the bug, or the feature, or whatever you want to call it; I can only influence what I can influence. How can I, the very first time someone hits a fresh site, how can I get .NET 1.1 into memory?

**Carl Franklin:** Well, I'm thinking about Hooks; and there is a Hook in Classic ASP - and it's been a while since I have done Classic ASP, but there is an Application Start that will get hooked the first time the application starts...

**Scott Hanselman:** Okay. You are thinking about hooks, that's good, but the only thing I can do in there is make Com objects.

**Carl Franklin:** That's right.

**Scott Hanselman:** So what other hooks are there in ASP.NET that you can potentially use.

**Carl Franklin:** Well, we have hooks in ASP.NET, but those -- Application Start happens in Global.asax.

**Scott Hanselman:** And by then the framework's loaded.

**Carl Franklin:** By then the framework's loaded, so it's too late.



**Scott Hanselman:** That was a tough question.

**Carl Franklin:** So it is too late.

**Scott Hanselman:** So what other hooks are there in IIS proper? Old - older?

**Carl Franklin:** Yeah, I'm thinking back.

**Scott Hanselman:** What did people write before?

**Carl Franklin:** Before ASP?

**Scott Hanselman:** Yeah.

**Carl Franklin:** We are talking an ISAPI Filter.

**Scott Hanselman:** Exactly. What can I do in an ISAPI Filter?

**Carl Franklin:** Well, I have never written an ISAPI Filter.

**Scott Hanselman:** Sure. And this is the point; most people who are being interviewed haven't. We are talking about hooks.

**Carl Franklin:** I know that that's the hook; I mean, what you can do in an ISAPI Filter is essentially redirect and get control when -- ASP is an ISAPI Filter. So basically, when there is a file extension, you can associate that with a Filter, you get control before that sub-system even loads. So, at that point however, I'd be hacking.

**Scott Hanselman:** Okay. That's cool; but you are getting there. So, the solution that we ended up using was, we wrote an ISAPI Filter that watched and loaded -- simply called one function - it was like, core bind to runtime.exe. Basically we made -- there is a COM call that loads the .NET framework. So we basically, in unmanaged code, in C++, just fired up the framework. Once it gets loaded, it's stuck there if you only have one framework per process, so it sticks there, and you are set. So, we just have a little filter, it's like six lines of code and it says, "Hey, I want the framework" - and it explicitly asks for 1.1.

**Carl Franklin:** Man, I am just amazed that that problem exists it all.

**Scott Hanselman:** Dude, that problem was a problem. We solved it; it took a day or two, but we had smart people sitting around, walking through the flowchart exactly like you and I did, and none of us knew the complete solution. And it is one of those problems where you look back

and you go, "Of course" - but that's the kind of problem, right?

(00:30:02)

**Carl Franklin:** Right.

**Scott Hanselman:** So, during the interview process I like to ask questions like that CSI kind of questions because of course you are not going to have -- maybe you would, but of course you are not going to have written an ISAPI Filter unmanaged thing that talk --

**Carl Franklin:** Right, but it is all about knowing where the points are and knowing where -- being able to isolate it.

**Scott Hanselman:** Right. And then you could have used that to 'Google' on the problem. But knowing that, "Well, I can't do it in ASP" - all I have is COM available - I can't do it in ASP.NET - it is too late - frameworks are already loaded. Those are the kind of things that tell me that you understand the life cycle - the lifetime of an ASP.NET application. Another example question that I offer sometimes up to people is, describe an HTTPGet from the moment it leaves your browser till it gets down to ASP and then back - stuff like that.

**Carl Franklin:** That's a good one.

**Scott Hanselman:** Because then you find out, well, of course they don't understand the exact details of, I don't know, IIS 7's pipeline or whatever, but they are not suppose to - but at least they know what they don't know.

**Carl Franklin:** Yeah. And then some magic happens.

**Scott Hanselman:** Indeed! And then a miracle happens.

**Carl Franklin:** I think you need to be a little more exclusive here. Alright Scott, what a great show! Thanks for dragging me through that. That was fun; made my day.

**Scott Hanselman:** I'm glad.

**Carl Franklin:** We'll see you next week on Hanselminutes.