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 #327
(Transcription services provided by PWOP Productions)

Jon Goodyear on Instant Messaging APIs
March 25, 2008

Our Sponsors

Lawrence Ryan: 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 polishing your device 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 #327, with guest Jon Goodyear, recorded live, Tuesday, March 18, 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 would really like to send the Finalizer to his mother-in-law's house, Carl Franklin.

Carl Franklin: Thank you, thank you, thank you. It's Carl and Richard here to bring you some .NET Rocks! happiness. Richard Campbell, how are you man?

Richard Campbell: I'm well, sir. How are you?

Carl Franklin: I couldn't be better. If I was any happier, I'd have to clone myself.

Richard Campbell: Very nice.

Carl Franklin: I don't even know what that means.

Richard Campbell: And all recovered from your weekend?

Carl Franklin: Yeah, I had a good time. I had fun. I'm not sure when the show is going to be published, March 25th?

Richard Campbell: 25th.

Carl Franklin: So, last week of course, big St. Patrick's Day bash here in New London. New London is weird. It's a small town and it really is a town that looks like a city but there are really only like a few blocks of city that you would think of as a city, very old New England, a lot of churches, a lot of brick buildings and stuff like that, turn of the century stuff and a lot of ship captains' houses and things like that. So, it's very small, but they know how to throw a party. This was the first year that they did an Irish parade in New London and it turned out to be huge like a -- not a lot of people showed up, but in terms of the parade itself, big, big, big, big big.

Richard Campbell: Nobody showed up because they were all in the parade.

Carl Franklin: I'm not going to say nobody showed up, but it was the first one, right? So I mean it wasn't jam-packed, but it was enough people to make it a good party and one of the best features of it is that Chester's BBQ was doing smoked corned beef and cabbage.

Richard Campbell: Same guy who did your birthday.

Carl Franklin: Yeah, the pig roast. There is nothing in this world -- go to your barbeque place and ask them to do a brisket corned beef style.

Richard Campbell: Nice.

Carl Franklin: But smoke it. There's nothing better in this world. Okay, enough about that. Let's get on to Better-Know-A-Framework.

[Music]

Richard Campbell: All right, sir. What have you got for me today?

Carl Franklin: This is turning out to be one of the more popular aspects of .NET Rocks!

Richard Campbell: I totally agree. It's been a lot of fun to listen to it too.

Carl Franklin: Yeah. I'm really glad I paid that consultant five bucks for the idea. So today's class is an interface and it's in System.Configuration. It's the IPersistComponentSettings Interface and this defines standard functionality for controls or libraries that store and retrieve application settings. The remarks are as follows. Typically, you can add application settings support to an application by creating a settings wrapper class, which is derived from ApplicationSettingsBase, and then add special properties to this class. However, this approach does not sufficiently encapsulate application settings for a control so that its container is shielded from the details. The IPersistComponentSettings interface provides an application with a standard interface for communicating application settings state change requests to a control, component, or library module. Design-time tools also depend on this interface to

properly manage controls and components. So, there you go. Read all about it in your local documentation or the online MSDN documentation.

Richard Campbell: Excellent.

Carl Franklin: So, Richard, what you got for me?

Richard Campbell: I've got a nice, quick email here for you. "Dear Carl and Richard. Guys, fantastic show with Stephen Forte on Remote Teams and even more intriguing, scrum."

Carl Franklin: Yeah. That was great.

Richard Campbell: "About three years ago, I was working on a high visibility project with a highly distributed team. Ultimately, that project failed because of constant thrashing and a lack of focus and defined roles. I could have used Stephen's passion and insight on scrum back then. Until your show, I had heard of scrum, but never looked too much into it. As Stephen described scenarios, basic scrum methodology, and pitfalls of traditional software development life cycles, I found myself yelling at the windshield 'Yes! Yes!' as I was flying down the I-684 to work and yeah, I'm from the lower New York area." Why do I have a vision of When Harry met Sally on that.

Carl Franklin: Yeah, the lower New York State area is above New York City and it's in the southwestern Connecticut corner. Yeah.

Richard Campbell: "So, now I am sold. I picked up Agile Software Development with Scrum by Schwaber and Beedle and have started reading. I hope to apply the practice to both personal and professional projects. Keep up the good work. Here is to more shows in the future discussing software development processes, Peter Meyer, Poughkeepsie, New York."

Carl Franklin: Poughkeepsie.

Richard Campbell: Poughkeepsie. I got it right. I was looking at it because it looks like Poughkeepsie, but it's Poughkeepsie.

Carl Franklin: Yup, all those great Native American names in New England, good stuff, like Schenectady.

Richard Campbell: Yes, that's a good word.

Carl Franklin: Schenectady, sounds like God bless you. Schenectady, yes, Gehsundheit. So, with that, let's introduce Jonathan Goodyear. He is of course President of ASPSOFT, Inc. He has been working with .NET since before it was made available to the general public. He is a contributing editor for both Visual Studio Magazine and asp.netPRO Magazine, and frequently speaks at major technology conferences such as VSLive, ASP.NET Connections and .NET user groups through the INETA, through the International .NET Association. Jonathan wrote one of the first books about .NET development, Debugging ASP.NET, by New Rider's Publishing, and appeared in a video, Visual Studio .NET - An Introduction, by WatchIT.com. More recently, he was co-author of ASP.NET 2.0 MVP Hacks, by Wrox Press. Jon has been awarded the Most Valuable Professional status by Microsoft and is also a Microsoft Regional Director and a heck of a good guy. Welcome Jonathan Goodyear.

Jon Goodyear: Hi. How are you doing?

Carl Franklin: We're doing great. It's about time you're on our damn show, man.

Richard Campbell: Yeah, what's up with that?

Jon Goodyear: Yeah, I know. I think you guys have been trying for about three years to get me on the show.

Carl Franklin: Yeah.

Richard Campbell: I'm going to credit myself with that. I think I only had to spend about two months getting you logged down.

Jon Goodyear: Pretty good, pretty good.

Carl Franklin: Now, you're in Central Florida, right?

Jon Goodyear: I am indeed, Orlando.

Carl Franklin: Yeah. Richard loves Orlando, by the way.

Richard Campbell: Oh, come on.

Carl Franklin: He loves it, loves the -- especially Disney. He's such a fan of Disney.

Richard Campbell: Okay, now you're just making fun of me.

Jon Goodyear: With Orlando, you either love it or you hate it. There's really no in between.

Carl Franklin: Now, Richard, you have said disparaging things about Orlando and I want to give you an opportunity to come clean and apologize that you did not mean the people of Orlando because we know lots of great people down there.

Richard Campbell: There are fine people in Orlando. You've even introduced me to a few of them.

Carl Franklin: Yup.

Richard Campbell: Of course, most people are transitory there too.

Carl Franklin: Right.

Richard Campbell: So that makes it very challenging and I would admit, the conference center there is extraordinary. I think that's why Microsoft keeps going back.

Carl Franklin: I think you just had enough of the mouse and the goofiness and the tourist thing, right?

Richard Campbell: Oh yeah, you know, I'm pretty tourist resistant. I'm the guy who likes to go off exploring on his own.

Carl Franklin: Yeah.

Richard Campbell: But be that as it may, Jon, my first knowledge of you was an RD meeting…

Carl Franklin: Oh, this is great.

Richard Campbell: At TechEd where you brought out that robot.

Jon Goodyear: Ah, yes, the Finalizer.

Richard Campbell: The Windows-powered robot. It was unbelievable.

Carl Franklin: Jon, you got to tell the story from the beginning because it's a great story.

Jon Goodyear: The beginning of the meeting or the beginning of the project?

Carl Franklin: No, no. Tell us about the robot and what it was all about, what it was for.

Jon Goodyear: Well, if you remember back when .NET 1.1 was first coming out, there was a little bit of a movement to get .NET on embedded processors, you know, there was a company called .NET CPU that came out, and they had embedded a beta version of the .NET framework version 1.1 on this board. So, when I was reading about it, I thought, "Well, wouldn't it be cool if we could build something out of this?" because one of the consultants that works for me, Brian Peek, who is a Microsoft MVP and writes the Coding4Fun column on the Microsoft site, he does a lot of hardware integration playing around. He's the author of the Wiimote library, which I don't know if you've talked about that at all, but he's the author of that.

Carl Franklin: We have.

Jon Goodyear: Then he went, he said, "Oh, this would be a great idea. Let's control something. Let's figure something out." I was watching Robot Wars on TV with my son and we're like, "Oh, we got to build a Battle Bot. That will be sort of a neat project," and so I spun it to Jen Ritzinger.

Carl Franklin: At Microsoft.

Jon Goodyear: And she thought it was a great idea and decided to commission it and we went ahead with it. It was interesting because we thought we were going to build the whole thing ourselves and so there's like a surplus store, surplus electronic store around here called Skycraft and so we went there and we bought an old, you know, those Hoveround motorized wheelchairs and bought one of those. I'm probably one of the only programmers that has his own arc welder, I bought an arc welder, yeah. We tore this thing apart and we're putting it back together and bolting motors on it and we came up with something that looks sort of like a Frankenstein tricycle. At that point, JRitz calls me up and says, "Oh, by the way, we want to use this Battle Bot up on stage during one of the keynotes at TechEd," which was 2005. So, we started to freak out because, you know, this thing was fine and dandy for a pet project, but really wasn't the caliber that you could really roll out on stage during a keynote.

Richard Campbell: Right.

Jon Goodyear: So, of course, we agreed and told her everything was great and, you know, everything's progressing right on schedule. So, we teamed up with a company called The Machine Lab out in Los Angeles who does robotics for police bomb-disarming robots and they do robots for Hollywood movies and things like that. They took pity on us and pretty much took the rest of the money we have allocated for the project and built us sort of a combat chassis with various weapons on it and all we had to do is snap in the .NET CPU.

Carl Franklin: Now, this thing had like a saw on an arm, right?

Jon Goodyear: It did, it have a retractable arm with a saw on it and it had an axe on it…

Carl Franklin: Great!

Jon Goodyear: That it could – it had quite a bit of power, you know. We were splitting logs, you know, like 2x4s…

Carl Franklin: Wow.

Jon Goodyear: In the garage with this thing.

Carl Franklin: Watch out.

Jon Goodyear: It was pretty powerful and it had a big lifter arm underneath it. We had a good time with it, but, you know, we started putting it together and we ran into some trouble and we called up the folks at .NET CPU and they stopped taking our calls and they folded up, you know, the doors.

Carl Franklin: "Go away! Go away!"

Jon Goodyear: So, it really got stressful for us, so I'm like, "What do we do now?" So, we opted to go ahead and use a PDA with a .NET Compact Framework on it and that proved to be a pretty good option. We found a serial port interface for it with CompactFlash. It was able to talk to the servo controller and Finalizer, and Brian who incidentally lives in Schenectady…

Carl Franklin: Oh.

Jon Goodyear: Yes, he's a "Schenectadian."

Carl Franklin: "Schenectadiite?"

Jon Goodyear: Yeah. He went ahead and just purchased some spare parts from Radio Shack, one of those tiny little robot model kits you can get, and he used that as his demo because they shipped the robot to me in Orlando, so he couldn't even -- so, he was building the logic for this thing "sight unseen" and so he build this and he comes down to TechEd. I flew him down about two days early before the conference and then he gave us the logic and we plugged it into the thing. Little did, Jritz know but we actually plugged the logic for this thing into the robot, we loaded it up about two days before TechEd.

Carl Franklin: Wow.

Richard Campbell: Oh man.

Jon Goodyear: Yes, a huge amount of testing that we did. So, yeah, we threw it in there. We got it working. Believe it or not, only after a few modifications, we got it to work.

Carl Franklin: Now, you said you wanted to use -- she said she wanted to use at TechEd, but you didn't really say what, what part of TechEd.

Jon Goodyear: Yeah, the part that it was going to play was at the time, Microsoft wanted to do a demonstration of SQL Server failover, what happens when network connectivity is lost from one SQL Server and they wanted to do a swap over to a backup machine and so we came up with a concept of the Finalizer rolling out on stage and spinning around and showing off its weapons and then going up to a router that was strategically placed on the floor, or actually a switch, and using its axe to chop a cable that was on top of the router and smash the router and so thereby causing the failover, I mean just in case, you know, just realistic scenarios Microsoft is famous for on their demos but… So, that was the idea and we went through it and…

Carl Franklin: Wait, wait, wait. What I'm trying to get you to say is this was during Steve Ballmer's keynote, right?

Jon Goodyear: It was indeed. Steve Ballmer was out there and there were several people out there. We had -- was it Samantha Bee from…

Richard Campbell: The Daily Show?

Jon Goodyear: Jon Stewart. Yeah, The Daily Show.

Carl Franklin: Yeah. It's a big deal.

Jon Goodyear: Yeah, it was a pretty big deal. We were pretty excited. I was kind of freaked out about it, but we went through the dress rehearsal the night before and we were whacking the switch. It wasn't causing that much excitement, so we decided to get some guys put some pyrotechnics underneath it to give it some extra flare. One of the toughest parts about it was I had to control this robot using -- we had it rigged up so that an Xbox 360 controller was the controller that we were using hooked up to a laptop computer and there was about 30 feet of dead space where you're completely blind when you were using this thing and so if you went an extra 5 feet beyond where it needed to go, it would have gone off the stage and into…

Richard Campbell: Into the press.

Jon Goodyear: Actually, into the plasma panels that they had. They were so angled up you can't…

Carl Franklin: Chop, chop, chop, chop, "Oh no, I can't stop the chopper! The saw won't turn off."

Jon Goodyear: And so one of the things -- we had a threshold of thing, saying, "Oh, if we lost the signal to the robot, it would shut off." It had been more than a couple of seconds, but then we were

having some connectivity issues, which I'll talk about in a second. So, we had to sort of amp that up to about 5 or 6 seconds, so, really, the robot could have gone on its own for 5 or 6 seconds before it actually shut down due to connectivity issues.

Carl Franklin: So, tell the story about the RD meeting Richard brought up here. It was day before the keynote, right?

Jon Goodyear: Yeah, so the day before the actual keynote, we had this idea. Since RD program had technically put up about half the money to put this robot together, JRitz goes, "Oh, let's do this private showing for the RDs," and I said, "Oh, it's great idea, great idea." So, we brought it out there and one of the things that we have done is we tested this thing in my garage and in my garage, of course, there was only the Wi-Fi, the laptop, and the PDA. Now, with all the RDs, they all brought their laptops, they spun them all up, and so at the time when we went to do the demo, the Wi-Fi spectrum was flooded and what ended up happening is the robot just malfunctioned completely and started twitching around and freaking out. So, of course, John Alexander -- with probably one of the best quotes I've ever heard -- spouts out with the "Uncle Owen, Uncle Owen! This R2 unit has got a bad motivator."

Carl Franklin: That was the best.

Jon Goodyear: I mean it was right when it happened and of course, you know, it was…

Carl Franklin: That was the best line. "Uncle Owen!"

Jon Goodyear: If it wasn't happening to me, it probably would have been the funniest thing I ever heard in my life.

Richard Campbell: It wasn't happening to us and it was the funniest thing we'd ever heard in our lives.

Carl Franklin: But at the right moment, I mean there was this silence and then the dread just came over you and then right there, boom! He was right there with that line.

Jon Goodyear: Oh, the timing was impeccable. Incidentally, I've seen John probably five or six times since then and he apologizes to me about that every time. I'm like, "Yeah, man. It was perfect, man. I would've done the same thing."

Carl Franklin: I would've done the same thing, yeah.

Jon Goodyear: But we took a break for lunch after that and over the lunch break, we secure, we locked down the Wi-Fi and we bashed a whole bunch of things. JRitz was coming up to us and saying, "Are you sure this thing's gonna work?"

Carl Franklin: That's right.

Jon Goodyear: "You know, we're going to be rolling the thing up on stage. It's one thing to fail from the RDs."

Carl Franklin: Let's not go after Steve Ballmer with a saw.

Richard Campbell: Yeah.

Jon Goodyear: Exactly, and of course, we really didn't tell him, the folks that were up on stage, that we had disabled some of the failsafes on it in order to get it to work.

Richard Campbell: Let me point out the man *is* a billionaire. He can mess with your life in ways you can't imagine should you, you know, cut off a toe or something.

Carl Franklin: He could move your house one foot back.

Jon Goodyear: Incidentally, the Finalizer, its resting place is in my garage now and I've got it in a surface-to-air missile case that we purchased from the military supply store. It's a little bit creepy, you know, because you've got this huge, gigantic missile case, but it's sort of a fitting place for it to be. It's very military looking.

Richard Campbell: Where did you get a missile case from?

Jon Goodyear: The Skycraft place. They sell all kinds of surplus and they had this big, green metal case and at the time, it wasn't just used at TechEd. It was actually used at two different Microsoft employee internal meetings.

Richard Campbell: Right.

Jon Goodyear: They shipped it to the, you know, I think it was the Visual Studio 2005 launch event. It was crisscrossing the country and we needed a protective case and so I went to Skycraft and they've got this huge green case out there. I said, "What is that?" They said, "Well, for a surface-to-air missile," and I'm "Hey, this is great."

Richard Campbell: "I got to get me one of those."

Carl Franklin: Yeah.

Jon Goodyear: So, we stuck it in there and it had to cut some foam interior. We really sort of went crazy with it and ended up -- although incidentally the military really doesn't care about the size of any package that they send because they've got their own, you know, they transport everything using their own planes.

Richard Campbell: Right.

Jon Goodyear: So, we didn't realize that this case was about 6 inches wider than a pallet and about 4 inches longer than a pallet.

Carl Franklin: Oh.

Richard Campbell: Uh-oh.

Jon Goodyear: The problem is that I learned that the post office has how much something actually weighs and how big it is and they have like an inferred weight based on its dimensions.

Richard Campbell: Right.

Jon Goodyear: If you try to ship something that's bigger than a pallet, they give it an inferred weight of how much it would weight if it took up two pallets.

Richard Campbell: Oh man.

Carl Franklin: Wow.

Jon Goodyear: So, this thing ended up -- it went from costing $500 to ship it across the country to about $1300 to $1400 to ship it across the country.

Richard Campbell: So, you know, Jon, you're going to have to dust off that device now. With .NET 3.5 shipping, I think somebody needs to write a provider for LINQ to axe.

Jon Goodyear: There you go. LINQ to axe.

Richard Campbell: There you go.

Jon Goodyear: Incidentally, if anybody wants to check out, if you go to www.finalizer.net, we still got link to the video on there and some pictures of it and so if anybody wants to check it out, you can…

Carl Franklin: So, the end of the story is the keynote went off without a hitch.

Jon Goodyear: It did, it did. It was fantastic and it did three subsequent events without a problem. Whenever it decides to malfunction, it was always off the stage, but when it got up on the stage, it went to a few harrowing moments, but we pulled it off.

Richard Campbell: Awesome.

Carl Franklin: So, you've been doing a lot of work with SMS, the text messaging, lately in .NET.

Jon Goodyear: Yes, you know, I wanted to -- ever since I've seen so many kids and teenagers and adults and everyone using text messaging, I thought there's got to be a business surrounding that that extends further than voting for American Idol. I mean there had to have been something that could be used as real software-based around text messaging. At the time that I started looking into this a couple of years ago, the canonical application for using text messaging that wasn't anything simple was Twitter. I mean, well, it is simple, but at we said software. It's not just voting or buying ring tones and things like that. At the time, you could go to twitter.com. Of course, it was founded by Biz Stone of Blogger fame. So, I go to the Twitter site and I'm like, "I wonder what they're using for text messaging," because it's got to be costing them a fortune to be powering all these people following everyone else using Twitter. The Contact Us page at the time actually had this cell phone number for Biz Stone incidentally. So, I called it up and I asked him. I said, "What are you using for your text messaging?" I said, "I'm not trying to copy what you're doing, but I'm building some software and I want to know what you're using to power your text messaging," at which point he hung up on me. No, quite literally, I think I got five words out of him or six words and then there was a click. That's my Biz Stone story. Anyway, he sort of left it up to me to do my own research and so I did and, really, the way -- If you know about email, you know, email, anybody can spam anybody else because even if you get caught with the United States rules you can always go and get some server over in the Ukraine or Russia or whatever.

Richard Campbell: Right.

Jon Goodyear: And essentially run all of your spam through there and there's not a whole lot you can do about it, but SMS, which is short message service, it's a lot different because it's directly controlled by the carriers and so if you abuse anything, they can shut you down and it's all heavily regulated, heavily controlled. You have to negotiate individually with each one of the carriers. If you don't want to do that, you can go to what's called an SMS gateway. Essentially, what they do is you broker an agreement with the gateway and then the gateway deals with each individual carrier and you communicate with the gateway when you're sending and receiving messages and they go ahead and they've got what's called a bind to each one of the major mobile carriers.

Carl Franklin: What's the advantage of using a gateway if you can just do it yourself?

Jon Goodyear: You mean, pertaining to -- I mean the only way to send a commercial text message is using a gateway because if you do it any other way, then it's against regulation.

Richard Campbell: And they'll cut you off.

Jon Goodyear: For instance, most of the major carriers have a way to send a text message through email. So, for instance, if you send a text message to your phone number at Tmail.net, that's for T-Mobile and there's Cingular like the phone number at cingularme.com and then you've got Verizon which is vtext.com. So, they've all got this numbers that you can go ahead and send a text message. The problem is that's for personal use only and you can't send any commercial. If you try to send any commercial, then they can actually shut you down and fine you and do all kinds of nasty things.

Carl Franklin: So, why is it that a gateway lets you send commercial messages? Isn't that spam or is it tightly controlled for your particular application in certain numbers?

Jon Goodyear: At that point, what you do is get a short code. Now, a short code, if you've ever watched American Idol, I suppose you've seen all the big thing everybody is familiar with and they say vote for this singer, text vote to 5634 or whatever on AT&T network. That 5634 is what you call a short code and a short code can be equated to -- it's very similar to a domain name, the parlance of short message service. The difference between a domain name and a short code though is that with a domain name, if you register a domain name, then you set it up and everyone who goes to that Internet address and the browser goes to your website. With a short code, you have to negotiate with each individual carrier to ask them or get approval, to have your short code work on that carrier, which is…

Richard Campbell: Wow.

Jon Goodyear: Yeah, it's sort of a pain, so we…

Richard Campbell: How many carriers are we talking about here? Just to get a scope of the problem.

Jon Goodyear: Well, there are several big carriers such as T-Mobile, there is AT&T, there's Sprint Nextel, there's Verizon, there's Alltell, there's Virgin Mobile, Boost Mobile, MetroPCS. The list goes on and on.

Richard Campbell: Are you just talking in the US so far?

Jon Goodyear: Yeah, those are just in the US and of course over in England, you got Orange and a bunch of other providers.

Carl Franklin: Rogers in Canada.

Jon Goodyear: It gets to be a huge issue. Now, if you want to send text messages to a person, a foreigner who is in the United States, it's not so bad because most of the foreign carriers have roaming agreements with a local carrier. So, if you send it to their number, as long as they're in the United States, it will forward it to them, but if you actually want to send a text message to somebody who has a non-US number while they're in their country of origin, then of course you have to negotiate that with the carriers over there as well as get a separate short code because the short codes are country specific as well.

Richard Campbell: Right. When you talk about negotiating deals with these different carriers, you're talking about a rate for each message?

Jon Goodyear: It's not a rate because the rates are pretty -- I hate to use the word commoditized, they're not exactly cheap, but I guess the term commoditized means it's pretty standard across the carriers, but what the negotiation is, is that they have to agree that what you are trying to do with your short code is something that they agree with.

Richard Campbell: Right. So they're really assessing your business model whether or not they want to tolerate it.

Jon Goodyear: Exactly. So, what you do is, well, the first thing you do is you have to register your short code and you can go to usshortcodes.com and register short code in the United States. It's the same way with registering a domain name. You search for one, see if it's available, and you can go... Incidentally, it's too pretty Wild West, young time for short codes because you can actually click a link to look at every short code in existence or at least every US short code. There are only 500 of them registered.

Richard Campbell: Wow.

Jon Goodyear: Yeah, it's still extremely brand new and one of the things it's probably leading to is not everybody can do it because the cost, at minimum, even if you just get a random short code is $500 a month.

Richard Campbell: So, you're not going to keep one around recreationally. That pretty much ends the squatting business right there.

Jon Goodyear: Yeah. It's extremely expensive. If you want a vanity number, then you're looking at $1000 a month and that's fairly pricey and the random codes, of course, they give you one with a lot of zeros in them because they don't have any letters associated with it on the keypad and so you're pretty much locked out of doing anything vanity oriented with it.

Carl Franklin: Do you know how to build web 2.0 AJAX applications with web 1.0 components? Right. You just can't. In order to have next generation web apps, you need next generation components and that's exactly what our friends at Telerik have for you. Their upcoming product codenamed RadControls Prometheus is a huge pack of web controls built on top of Microsoft ASP.NET AJAX, which will add previously impossible performance and interactivity to your next project. Just listen to this. The new controls mirror the ASP.NET AJAX API, so development is straightforward. Client scripts are shared, so loading time is pretty much instant and if you just set a couple of properties, you'll be able to automatically bind to Web Services for even more efficient operation. After all, the facts speak for themselves. The new RadEditor for ASP.NET AJAX loads up to four times faster than before. Similarly, RadGrid handles thousands of records in mere milliseconds, but, again, it's best to try it for yourself. Visit telerik.com/aspnetajax and download a trial. And don't forget to thank them for supporting .NET Rocks!

It seems to me that this system is in place partly to control it, but also to prevent spamming because that is obviously a concern that everybody has, that you have to pay for your tax messages that you received and if you get a, you know, it's just like the whole fax problem, right? You have to pay for that, but it's worse because they can add up, but yet, while I was in Las Vegas, 2:00 in the morning I get this instant message from some number that's like, "Are you awake? Call me." It's like -- so, the hookers are using it in Las Vegas.

Richard Campbell: Yeah, which is just the kind of message you might get from a friend who is wondering if you're in town.

Carl Franklin: How is that possible? Is it corruption? How does that happen?

Jon Goodyear: Oh, I've even sent an accidental text message. I actually sent a text message to my sister for her birthday and I didn't realize she had changed her cell phone number and so some random person text me back. It's like, "Who are you?"

Carl Franklin: But you've seen spam though, I mean…

Jon Goodyear: Oh yeah. It's quite a problem and if you get more than one, the company who sends it can be severely fined. For instance, Sprint charges $10,000 per incident.

Richard Campbell: Wow.

Jon Goodyear: Yeah. They're not messing around at all.

Richard Campbell: Just to be clear here. I know on my bill, I don't pay for incoming SMSes. I pay to send them.

Carl Franklin: Really?

Jon Goodyear: Yes, and people...

Richard Campbell: And it's to send.

Jon Goodyear: I pay $10 or $15 a month and get unlimited just because my…

Carl Franklin: Well, I have unlimited too but in Verizon, I'm not sure, but I know people who have to pay to receive text messages. It depends on your plan.

Jon Goodyear: Speaking of the cost of sending and receiving, we did a little bit of math and, sure, SMS messages are the most expensive bandwidth in the world.

Richard Campbell: Oh yeah because their maximum 160 bytes.

Jon Goodyear: Yes. Just for instance, if you are sending and receiving a $0.10-cent per message and want to send for instance an MP3, if you divide it up into 160-byte increment…

Carl Franklin: Whoa.

Jon Goodyear: And sends it over, it will cost you $6000…

Richard Campbell: Nice.

Carl Franklin: Wow.

Jon Goodyear: To send a single MP3 file through text message.

Carl Franklin: Unless you have unlimited.

Jon Goodyear: Or if you want to download the entire contents of a medium-sized iPod, it will be approximately $30 million. It's definitely an expensive bandwidth.

Carl Franklin: So be careful before you download your iPod via SMS, okay?

Jon Goodyear: Exactly.

Carl Franklin: This is all we're saying.

Jon Goodyear: And that's for a consumer at $0.10 a message. Now, you're thinking, "Well, what about the gateway people?" you know, the people like me that are creating software and they're sending it, you're still negotiating it $0.05 to $0.02 a message. Now, there's some maximum that you can reach in order to get much reduced rates, but even at the standard rates, it works out to about $125 a megabyte or $125,000 a gig and so it really is an expensive protocol. You're saying, "Well, is it costing the cell phone companies all kinds of money?" It's not.

Richard Campbell: No.

Jon Goodyear: So, a little bit of background about SMS is SMS was not created to be SMS. It's actually sent over what's called the control channel. Now, if you have your mobile phone and, say, when somebody makes a call to you, your phone starts ringing because it knows a number is coming in. That's because your phone is maintaining sort of a lightweight link to the nearest cell tower and it maintains it over what's called the control channel. Well, a few years ago, maybe I think it's been seven or eight years ago, some folks over at Nokia said, "Well, we can actually use this for sort of an instant messaging type of application and send signals over," and so, really, when you're sending an SMS, you're just sending a 160-byte signal over this control channel, which costs the cell phone companies zero because you need to maintain that control channel relationship anyway.

Richard Campbell: Yeah. This is just added on to the standard charter between the phone and the node.

Jon Goodyear: Yes. So, essentially, it's free to the mobile companies and they're charging $125,000 a gig to send it in.

Carl Franklin: Wow! Supply and demand.

Jon Goodyear: So, it's pretty amazing -- you've heard of printing money. I mean this is printing money with no paper and no ink.

Carl Franklin: Wow.

Jon Goodyear: It's pretty amazing.

Carl Franklin: Well, that's a classic story of supply and demand.

Jon Goodyear: Yeah.

Carl Franklin: They control it and there you go.

Jon Goodyear: Incidentally, that's where the 160-byte limitation comes from because they were like, "Why did they do that?" Well, it's because they essentially shoehorned this instant messaging capability into a protocol that really wasn't designed for it. It just happened to be convenient that they could do it.

Carl Franklin: What is that protocol? Is it just the SMS protocol or you're just talking about the underlying protocol?

Jon Goodyear: Yeah, it could be whatever control channel protocol is. I mean it sends a 160-byte packet back and forth between the cell tower and the mobile phones. It's like, well, perhaps a little bit more, but of course you have to have routing information.

Carl Franklin: I wonder if it's like XMODEM? Something like that.

Jon Goodyear: The control channel protocol is similar to what ICMP does for the IP stacks. It's transmitting and routing and error information and things like that…

Carl Franklin: That's good.

Jon Goodyear: Just to keep your phone in tune with what's going on with the tower.

Carl Franklin: That's good.

Jon Goodyear: Now, MMS messages are sent over the GPRS standard data networks at still fairly minimal cost, but it's a tiny bit more than SMS protocol.

Carl Franklin: What is SMS and MMS? What are the differences there? What are they?

Jon Goodyear: SMS is -- there are actually three forms. There is SMS and then there's like EMS, which is Enhanced Messaging Service. Now, SMS is just text only. EMS is Enhanced Messaging, which means you can do things like bold and italics, ringtones, things like that, a little bit of text formatting,

sort of an intermediary between SMS and MMS, which is Multimedia Messaging Service and that's where you can send rich media like audio, video images, things like that. Now, there's been some contention that MMS really isn't going to go anywhere because it hasn't really taken on too much by now and now phones are getting so smart that the feeling is that people are just going to switch directly to doing things through a browser.

Carl Franklin: Yeah, that's what I do.

Jon Goodyear: So, the MMS sort of fad is really sort of dead, whereas, SMS because it's just simple text, it's almost like email on your phone, the feeling is that that's still be going to be around for quite a while.

Richard Campbell: I think everybody gets SMS and, hence, uses it. It's one of the few sorts of universal technologies across virtually any carrier. So, it's pretty compelling in that respect and it's just like instant messaging. That's how people use it.

Jon Goodyear: Exactly, exactly. From the concept of what happens, you know, how do you get involved with this, I mentioned you register your short code and you have to find a gateway and there's several out there. I use a company called Open Market. They used to be called Simple Wire, but they were purchased by Open Market, but there are several others like Clickatell, Celtra, Ez Texting. If you look online, there are dozens to choose from both in the United States and abroad. England is extremely big into text messaging. There are a lot of companies that are based over there. I decided to go with a domestic carrier. Open Market had a pretty good reputation and so I stuck with them and what you have to do with them is you submit what's called a program summary and the program summary is where you have to -- when you open up a program summary, the file that looks a lot like a movie script because you have to essentially describe the conversation that's going to happen between your customers and you as a service. So, it's a series of lines to say MO, meaning Mobile Originated, which means they're coming from your customer, and MT, which is Mobile Terminated, meaning it's a message that you're sending to the handset. You have to essentially document every interaction that you're going to have with them down to the last detail and then you submit that in the carriers then test your -- they essentially put your short code live for a limited white listed set of numbers and they test it out and make sure that it complies and you have to have things like if they submit help, if they text help to your number, it has to come back with a phone number or a website or an email address and there are certain regulations for it that has to be if they text quit or stop to the short code, those are industry standard for "unsubscribe me immediately from everything associated with this number" and those are the things that you have to comply with. So, there's definitely a lot of standards that have put into place across all the carriers to sort of minimize that spam, so if you get something, a message from someone and you say, "Well, I don't know how I got subscribed to this, but I want off." If you text back stop or quit, then you should get a message back saying that you've been unsubscribed completely.

Carl Franklin: So, you said that MT is mobile terminated, MO is Mobile Originated.

Jon Goodyear: Correct.

Carl Franklin: What about OM? That probably means that it just goes off into the ether, into eternity, om…

Richard Campbell: Nice. Oh geez.

Jon Goodyear: Yes, exactly, exactly.

Carl Franklin: You like that, Richard?

Jon Goodyear: That's also one of the things that you get by going through a gateway using a short code is your messages that you send through the gateway, essentially they get premium treatment. They bounce to the front of the cue before sending the messages. For instance, I mentioned before that the SMS is sent over the control protocol. Now, obviously the cell phone companies don't want the spectrum to get flooded with all these text messages, so what they do is they throttle it, so you could end up -- that's why when you send text messages to somebody, sometimes they don't get it for an hour, a day, sometimes it just gets caught somewhere and you just don't hear from it for a while.

Richard Campbell: And there's no guaranteed delivery either. I've had text messages that just disappeared into the ether.

Carl Franklin: Om…

Jon Goodyear: Yes, occasionally. One of the things that I found out through working through Open Market is I get an email every time one of the carriers is having a problem if they're having a slow down or something breaks and it's pretty scary actually if you see how many times the text messaging, the SMS infrastructure breaks down, and they'll say "Verizon is having a problem," or "AT&T is having a problem. We'll keep your apprised and let you know when it's back up and running." So, even going through a gateway, there's still this feeling of it's not really guaranteed and so the software that we were considering building and which we're eventually going

to build, so I can't mention exactly what it is right now, but it relied on having extremely prompt -- essentially, the messages had to be delivered extremely quickly. Really, the technology is not there yet. There's still too much break down in it. You really can't get that guaranteed delivery. Now, through the infrastructure, when you're sending a text message, there are several ways that you can do that once you actually get everything set up. You can get a ticket back and…

Carl Franklin: That was my next question. How do you know it was delivered?

Jon Goodyear: When you send a text message over the web service or however you send it, you get a response back that has a ticket ID on it and then you can subsequently ping the service and say, "Whatever happened to this ticket ID?" A big problem with that though is that you can only query that for up to three days and after three days, you can't query anymore. So, if your message didn't get delivered for three days, which occasionally happens, scary as it seems, then you really have no indication whether that message ever got delivered. Some of the carriers, AT&T in particular, offers the service where it will send you a message from the gateway back to your service letting you know when a message was delivered, which is sort of neat, but not all the carriers support that yet.

Carl Franklin: Is that how your phone knows to say message sent or the message could not be sent or is that just basic we completed the protocol of sending?

Jon Goodyear: Yeah, that just means, "Hey, I delivered it to the other carrier and the other carrier said I got it," but it doesn't necessarily mean that it made it to the other person's phone.

Carl Franklin: I got it.

Jon Goodyear: Which is unfortunate. There's a lot of both political and technical murky waters you have to get through in order to get these things approved. For instance, some companies like Virgin Mobile and, believe it or not, Verizon, they take a look at our program summary for five minutes and said, "All right, this looks good," and they turned us on. Now, other carriers, like T-Mobile and AT&T, they had this whole established process for analyzing your program and going through all the settings. We told them and said, "Let us know when you're about to do it. Let us know when you're about to do it. We'll make sure everything is going…" Well, they tested it without us knowing and of course they failed us and if you failed twice, then you have to go back to the beginning of the cue and the beginning of the cue means six to eight weeks and so it's extremely stressful when you're going to get approved for this carrier thing, these mobile carriers, because they have extremely long cycles for approving the codes.

Richard Campbell: Is that just because they're serving so many potential vendors that there's not much interest in this space?

Jon Goodyear: I think there is a lot of growing interest and the problem is I don't think they've really expanded their infrastructure for it.

Richard Campbell: Right.

Jon Goodyear: Now, recently, Sprint went through a revamping, but we submitted one of our proposals on like December 12th or 13th of 2007. They said, "We're on a one-month hiatus for the holidays."

Richard Campbell: Nice.

Jon Goodyear: It's like we had to wait until mid-January before we got them to approve it, but they actually came out with a brand new system that allows it to be approved a lot faster. So, I think some of the carriers are starting to realize that there is more and more interest. Like I mentioned, there's 500 codes registered right now and I would say at least 300 of those 500 codes are owned by radio stations.

Carl Franklin: Really?

Jon Goodyear: You can actually go, if you go to usshortcodes.com, you can do an advanced search and click on company names, start with all, and there's a link for all and you can just scroll through them and see who owns the code.

Carl Franklin: Wow.

Jon Goodyear: Can you imagine what a web page would look like where you can just scroll through who owns every single domain. It makes no sense at all. There's billions of them.

Richard Campbell: Yeah.

Jon Goodyear: It's definitely such a new concept of using it, text messaging for anything other than purchasing ringtones and things that the carriers have done themselves.

Carl Franklin: Well, there's a job for LINQ,

Richard Campbell: So, you talked about working with Open Market as a gateway to all of the carriers, but you're also trying to work with the individual carriers as well?

Jon Goodyear: Well, you actually submit the program summary. It gets approved by your gateway and then your gateway negotiates with each individual carrier, but you still have to sort of deal with them on a little bit of a tangential level.

Richard Campbell: But then it kind of makes sense to me that Verizon would just sign off on the "if it's good enough for the gateway, it's good enough for us" kind of thing.

Jon Goodyear: Exactly. Of course, the thing about it is though is they're not just shooting you off into the ether and saying go do to it whatever you want. They reserve the right to audit you at any point in time.

Richard Campbell: Of course.

Jon Goodyear: Or if they get complaint, they know they can just shut you down. So, if you say you're going to run a chat service and that chat turns a little bit too much like porn, then they can just turn you off. You definitely have to stay on your toes. For instance, you are not allowed to -- if you've ever sent a lot of newsletters out, most people don't really care if they're sending emails to a lot of dead email addresses. They just send them off, send them off, and whatever, they get bounced, no big deal. Well, the cell phone carriers, every four hours, they update their canceled number list and with the canceled number, if you send more than one text message to a cancelled number list, it's up to $10,000 fine per incident. They take it very seriously. One of the services that Open Market has provided to me is they monitor the dead phone number list for me so that if I try to send a message to one of those numbers, then they actually swallow it and they send the message back to me saying, "Hey, don't do that again." So, at least they prevent me from getting fined because that could be pretty traumatic if that happened to me.

Carl Franklin: And expensive.

Richard Campbell: So, how are you writing the code against Open Market?

Jon Goodyear: Open Market has three different ways to interact with their system. The first one is they have sort of an SDK that you can use and the SDK, unfortunately, they've got a Java one and a PHP one and a Perl one. They really didn't have a .NET one which was a little bit upsetting to me. The second way is you can just do XML over HTTP which is really the same as using their SDK because their SDK is just, "Okay, there could be XML over HTTP. We just wrapped a bunch of objects and given it to you," but actually they allow you to go directly against the service.

Richard Campbell: And you call it XML over HTTP? It's not actually a Web Service?

Jon Goodyear: It is not a Web Service. It is actually just they post it to this URL and in the post value -- well, in the post value, you are posting XML, in the post body, but it's not…

Carl Franklin: Not SOAP.

Jon Goodyear: It's not SOAP, no WYSDL, no anything else like that. They actually have their own protocol called the wireless messaging protocol and it sounds awfully industry standard, but it's just really Open Market standard.

Carl Franklin: "woomusee..."

[Laughter]

Jon Goodyear: I'm not sure if any other gateways have adopted it just through…

Carl Franklin: No, wait a minute, I got it, "whimsy."

Jon Goodyear: Exactly, exactly. It's basically, you know, I don't know, through osmosis, maybe some of the other gateways have adopted it because it's relatively powerful. The third way that you can interact with it is through the short message peer-to-peer protocol or SMPP and that's a direct bind into the protocol. You maintain an open connection at all times for when you're sending messages, but oddly enough, you get less throughput using that than you do with the XML over HTTP, the way they've got it set up. One of the advantages of using SMPP is that if your connection gets broken, you know immediately and so the fail-safes are a little bit better, but the XML over HTTP which is what we're using is a little bit better for performance. The gateway, they maintain their own binds which each individual carriers so when you send it to -- like when I send it to Open Market, they have a direct bind to each one of the carriers so they can send the message along and they can process 30-40 messages a second, so it's fairly robust how much they can process. We use XML over HTTP. They didn't have an SDK so we went ahead and built our own using .NET.

Carl Franklin: Why not?

Jon Goodyear: It's got a request object and a response object and an SMS object. They have some static methods like send message. Essentially, you *"new up"* an SMS request object which we created and you set some properties like, for instance, you'll set like the source carrier and the source carrier is a number code, for instance, T-Mobile is 79 and I think AT&T is 238 and you set the

source type of number, which is either a network short code or a standard number depending on what type of number you're sending from. You set up the destination carrier and the destination address which is the phone number you're sending it to and then fill in the message and send it off. Now, the message can be either plain text or it can be text and coded and of course if you're sending something like a ringtone or an image, then you would text and code it versus sending it through a standard ASCII type.

Richard Campbell: So, why do you need to know the source carrier?

Jon Goodyear: The phone that you're sending it to processes that.

Richard Campbell: Right.

Carl Franklin: Okay.

Jon Goodyear: They want to know. They won't process it through unless you specify.

Carl Franklin: And the gateway doesn't just add that information.

Jon Goodyear: What we've got is we've got this object and we pack it all up and then we have an SMS object and we say, "Send SMS," and we pass in their request object and it turns all of those properties into a big XML node with a bunch of sub-elements that define all those different things I just mentioned and then it sends that across the wire and what it gets back is a response that says either "no, it didn't work and this is why it didn't work," or it gives you a status code that said it worked and "here is the ticket ID for that particular message that you sent across the wire." When you post it, you post it with an ID and a password that is specific to you so that you can send it over SSL if you don't want people sniffing your ID and password and using it for nefarious purposes.

Carl Franklin: So, Jon, do you give away your code or is this just part of your proprietary software?

Jon Goodyear: Oh, you mean the source code?

Carl Franklin: The wrapper, yeah, the wrap around the source code.

Jon Goodyear: Yeah. By the time this show is on the air, we're probably pretty ready to release that SDK. We're making some -- like right now we're making some enhancements to it to do some more interesting things. For instance, there are multiple pipes of requests that you can make. One is called the Submit, meaning I'm just sending a message and some of the other ones that you can do are like a query, which is if you send a query request, it's like saying "whatever happened to that message with a ticket ID," it's a big GUID-looking thing. There's also Preview and you could send a preview on a number. Say you don't know the carrier for a particular number, you can send a preview request over and it will do a lookup and send it back to you and say, "Hey, this phone number is with this carrier. Here's an approximate GPS location for them," which is not as fun as it sounds. More of "here's a GPS location of wherever they registered their phone," or whatever cell tower they are using. It's not extremely accurate. It even lets you know that if that person switched from a different carrier, which I thought is a little bit of a privacy thing, but it will actually tell you if you bought a new iPhone and you transferred from Verizon to AT&T, it will tell you that in the preview function. Those are the three going out and then coming in are Deliver and Notify which delivers the message that comes to you and says, "Hey, this message was delivered," which is only supported by a couple of carriers and then notifies something where it will tell you, "Hey, this number is no longer valid," or something like that if you try to send a message to a number that is no longer valid on that carrier. So, we're making some enhances to our library to make sure that we handle all the different scenarios as well doing MMS messaging.

Carl Franklin: Will the library be free?

Jon Goodyear: I can't imagine. It will either be free or the cost will be negligible. I would say it's probably 50:50 between being something negligible like less than $50.

Carl Franklin: Shipping cost.

Jon Goodyear: It's definitely not something we're looking to get rich on, something that we built out of necessity because we didn't want to have to deal with XML parsing all the time. That library that we built is for sending messages out and then what happens when the carrier wants to send you something? What they do is they do the exact same thing. Unfortunately, they won't call one of your web services either.

Richard Campbell: Right.

Jon Goodyear: Really, what ends up happening is they want to post it to a URL and so the best way to do that we found in ASP.NET is just to build an HTTP handler. So, you give them a URL to an ASPX page and then they call that, your HTTP handler grabs it, and there's a big block posted value called XML and you'll grab that. We've got a request object also if you feed it the XML for the call that's coming in, it will parse it out and populate all the properties and say, "Here's all the stuff and everything coming in." One of the unfortunate things about

dealing with short codes is that they even charge you to test it. So, we've got a test code and the test number, when you send messages -- first of all, they charge you $450 a month just to maintain a test account. This can't be used for production, only for testing. It's like $450 a month and not only that, it's $450 a month plus messaging fees. You have to buy message credits.

Richard Campbell: Wait, wait, wait, and that's just for the test service?

Jon Goodyear: That's just for the test service.

Richard Campbell: Oh man. The phone companies are amazing, aren't they? Holy cow! What a scam.

Jon Goodyear: Well, see, that's the gateway and apparently because the phone companies are charging the gateway, the gateway has to pass it along to you.

Richard Campbell: Right.

Jon Goodyear: What they do to stem the cost a little bit, they still charge you $450, but the reason it's not even more than that is because they'll let more than one person share the same testing short code. So, for instance, any time I want to send a message to a test short code, I'll have to prefix the message with a secret code and then I'll have my real message after that, so when the gateway receives that message, they know which HTTP handler to route it to. What we did to get around that a little bit is we built a host. Think of it as kind of like Rhino Mocks for SMS. Instead of mocking it, it actually implements it. It's faking out the whole system such that it looks exactly what you're sending an SMS message out but it intercepts it and just routes it to a local HTTP handler wherever you want to catch it. So, you can test out your whole process unlimited without having any short code or anything at all, sort of practice everything and getting it all squared away before you go ahead and take the punch and pay all the money. We thought that was kind of a cool way to sort of get in the game and what we'll do is make that available as well so people who want to download our library and sort of play around with it and they also want to play around with our test carrier, they don't have to pay the $450 a month. They can just sort of wire up our whole little infrastructure and test it out and play around with it and decide if they want to really make the investment because once you make the investment, you get a live code and a live code starts at $1500 a month.

Richard Campbell: Yeah. The ticker is now running and you better be making money on it. So, where are we going to be able to get this from?

Carl Franklin: Yeah. That was my next question. Where can we download this when it comes out? ASPSOFT?

Jon Goodyear: My company website, my consulting company, ASPSOFT, has a download page and so we'll probably make it available through that.

Carl Franklin: Okay, aspsoft.com.

Jon Goodyear: Yeah, aspsoft.com and we have a download link and you can just go there. It talks about some different things that we do and so we'll just make it available through that, but I think once we get it really stabilized, we'll release that and hopefully that's pretty soon because we're going to be launching our product which I'll talk briefly about. What we want to do is we want to build some software on text messaging and we prefer to find a problem and implement a solution rather than come up with a solution and then go around looking for a problem which happens all too often these days. What we decided was that when people go out and they go somewhere -- they go to a restaurant and they pass by on the road and they're like, man, I like to go down there and eat but you have a coupon and I have a coupon at home, and idealistically, I can't bring myself to eat at that restaurant knowing I have a coupon sitting at home. That sprung the idea of creating a system whereby it used your cell phone to act as the medium for which you receive coupons. That in itself is not revolutionary because there are a couple of other services out there that will allow the companies to send coupons. Our spin on it though is that each one of these establishments, we install a PDA that has a Wi-Fi link to a web service that we've implemented and so when you go into the store, you can actually punch in your coupon code, and it will display what the coupon is for thereby allowing the store vendor to track the success of their campaigns from start to finish. We call the service plumreward.com, so that should be live by the time this show goes live and we're beta testing a couple of smoothies shops. I don't know how popular smoothies are up in the northeast.

Carl Franklin: We love smoothies.

Jon Goodyear: But down here in Florida, smoothies are huge and Planet Smoothie is the king of smoothie companies. They've got about 60 or 70 stores down here in Florida, so we're going to be beta testing with them and so hopefully it will be successful. People will go in and sign up when you go in there. Essentially, it's a loyalty system that goes with you everywhere. So, your cell phone number becomes your loyalty card instead of having to carry a

loyalty card or one little keychain fobs or anything else.

Carl Franklin: Sweet.

Jon Goodyear: You just go in, you punch in your cell phone number and it tracks that you did your visit and the storeowner can see who their best customers are and send them out wonderful coupons and you come back and enter in there and then everybody can keep track of little reports.

Carl Franklin: Sounds fun and delicious.

Jon Goodyear: That's sort of our idea, but really, through this whole process, we discovered that not too many people are building any kind of software at all using text messaging and the reason we know that is because we had to deal directly with the carriers to get our short code approved and we when we handed them our program summary, they had no idea what we were doing.

Carl Franklin: It seems like a great business case just for simple notification of problems that require your immediate attention. It is actually for critical systems.

Richard Campbell: Funny you should mention that Carl. I worked on a project a number of years ago, I'd literally say 2000 where we were doing exactly that using the SMS to notify when servers were down in far away locations.

Carl Franklin: About medical?

Richard Campbell: Interesting enough, we had a bug in the very first version and our test run sent 30,000 SMSs to one phone number.

Carl Franklin: Oh! Oh, dude.

Richard Campbell: Of course, back then in 2000, there wasn't a nice interface for handling SMS. You literally would say SMS arrived, you would then select it and say read SMS. It would show it to you and then you'd have to individually delete it. The moment we realized what was happening and it kept happening and stopped it, there was already 30,000 in the system.

Carl Franklin: Yeah.

Richard Campbell: And we had to get on the phone and get all the way through to the carrier to say, "Can you empty that out, please?"

Carl Franklin: I could see if gets more reliable, it could take the place of pager protocols.

Richard Campbell: That's exactly what we're thinking but of course that time, it was so early on in the SMS days that nobody hated it yet. There was no punishment for that, the carrier didn't care, they laughed at us, and cleaned it up, but just thinking about trying to do something like that today, no way.

Jon Goodyear: Yeah. The people are a little bit unfriendly about it. We ran into some different problems because these people, they looked at our program and they said, "So what are you selling?" I said, "We're not selling anything." They said, "How much are you charging to receive the messages? We're like, "Oh, we're not charging anything. We're charging the Planet Smoothie guy, to send now the coupons and have our service and PDA and installed in the store, but we're not going to charge you to receive a coupon other than whatever your carrier charges you to receive text message if you're not on unlimited plan." They said, "What ringtones are you selling? What are we voting for?" They have this idea on their head of here's the sixth or seventh things that text messaging are used for.

Richard Campbell: Right.

Jon Goodyear: They have no concept of using text messages for real software.

Carl Franklin: Yeah.

Jon Goodyear: This is coming straight from the carriers who have to approve everybody. So you should add that not many people are taking advantage of text messaging. There are some severe limitations because things you have to think about, for instance, there's no concept of history so you get fooled when you buy an iPhone and you get the threaded chat conversations or threaded SMS conversations, but in reality all that's doing is just stitching it together at the client's side and all of these another text message came in from this number so I want to stitch it to the end. In reality, it's not keeping track of any histories, so when you send a text message into our number, we don't know whether you'll respond. If I send you two messages and you respond to one, I don't know which one you've responded to.

Richard Campbell: Yeah. It's purely state management. I think you have a great idea here, Jon. I'm just imagining a restaurant can take advantage of this with your regular customers just to say, hey, for you, today, 10% off the meal if you come in today and give me this code."

Jon Goodyear: Exactly. We've got analytics that can say, they can tell you when you're slow periods are, when people are coming, when people are not coming and so you can actually say, come.

What makes the coupons powerful is that they can be time sensitive so that you can say, "This coupon is only good on Thursday between 4:00 and 6:00 p.m.," whereas, before, I mean otherwise, you can only do that with the coupon that arrives on your phone and they have no way of validating because the guy behind the register, he's not going to know.

Richard Campbell: He's not going to know. Now I can see why you're sensitive to delivery time.

Jon Goodyear: Exactly.

Richard Campbell: If that thing doesn't show up for the day, now you have upset your customer.

Carl Franklin: You guys realized what you're saying, dudes? Did you realize that you're planning spamming people? Dude, you know, this is not good.

Richard Campbell: It's all totally opt-in.

Carl Franklin: I want this phone to read my mind.

Jon Goodyear: Yeah and that's one of the things where you're allowed to send one text message and we actually had to negotiate this because technically, the rule is zero, you're only allowed to send zero unsolicited text messages to somebody, but we convinced the carriers to allow us to send one and you're allowed to send one message, we're allowed to send one message, and then they have to reply back saying they agree to it and then we have to send a confirmation back to them. For instance, if I go in to Planet Smoothie and I buy a smoothie and I give your cell phone number, then it's going to send one text message to you saying, "Do you want to sign up to receive coupons?" and if you just ignore, you're not getting anything else. So, that presents the whole signing other people up for newsletters kind of thing to happen.

Carl Franklin: This is your last warning. We are going to go away. We promise. If you do not act now.

Jon Goodyear: Yeah, exactly.

Carl Franklin: And then we'll change our name under another business and send you another message.

Jon Goodyear: Yes. We certainly don't want to become the spam kings of SMS.

Carl Franklin: Yeah, you to be careful there.

Jon Goodyear: We definitely only want people associated with it if they don't want to be and at the end of the day, it's a loyalty system based on your phone number even if you decide that you don't want to receive the text message coupon.

Carl Franklin: Well, Jonathan, we're just about out of time. This was a great show and I'm looking forward to your toolkit because I can find some uses for it right away. I mean just as you are talking about it, I've got several places where I want to put this.

Jon Goodyear: Yeah, really, the key to it though is the fact that we've implemented this testing framework for it, so it really allows people to say, "Well, you know, I'm really interested on how this SMS thing works or at least how Open Market has implemented it and the terminology." You can download all the documents for how everything works and then just use our toolkit to sort of simulate it, sort of increase your knowledge without having to fork over thousands of dollars a month in order to reserve codes and get everything going.

Richard Campbell: Yeah, Jon, you definitely lowered the bar on the cost of testing this that anybody can test anything they want until they've got some confidence on what they're doing and then talk to the carrier.

Carl Franklin: It's cool.

Jon Goodyear: Exactly, and just a little bit of advice is to give yourself about two months to get all this stuff in place once you decide to say go. This is not something you can just come up with something at a whim and decide that I'm to troop a website next week and you're ready to go. It actually takes close to two months to get all these stuff squared away. That's really frustrating to get everything going, but it's worth it once you get everything set up. You definitely need to plan ahead. Don't leave this as the last step in your process. They don't charge you huge money, you only have to pay the $400, $500 a month while you're in application process which is, I guess if you're going to set up a business, that's just the cost of doing business. The sad part is once you get everything live, you have to maintain the test code as well, because otherwise what are you going to do? You're going to stage and test your new build of your software using your life code? You can't do that, especially when you end up having to pay the $1500 a month for your life code, plus another $500 for your test code, plus another $500 for your short code, I mean you're out many a thousand dollars just to…

Carl Franklin: Send some messages around.

Jon Goodyear: Just to get it around, but you know I think there's a lot of innovative opportunities, people that feel that they've missed the boat when it

comes with the web, email, and all those things that people did innovate on things but it's still pretty Wild West when it comes to text messaging because there are a lot of text messaging going around but all doing the same thing. There are so many ideas out there that haven't been explored.

Richard Campbell: Yeah, no innovation.

Jon Goodyear: Yeah, they said in the United States, there are seven billion messages March alone or something, or February alone, that were sent through text messages, but it's just people talk to other people, or people voting on American Idol, there's really nothing innovative going on aside from just like what I said Twitter. Twitter is the apex of innovation for SMS. Twitter is a great application, I have nothing against, it but it's not hugely innovative.

Carl Franklin: Yeah.

Richard Campbell: Still trying to figure out how Twitter makes money.

Jon Goodyear: Yeah, I'm a little curious on that as well. They're definitely hemorrhaging money when it comes to messaging cost but… I guess there are some maximums that you can achieve once you sent a certain amount of messages, they let you get them for free which I'm sure they're hitting, but it's definitely an interesting landscape because you feel like the days back when the .NET framework first came out and people just exploring the framework and figuring out, here's the new class, it does something cool and so every time we think about text messaging, we're having our design meetings and we go, here's another cool thing that we can do so we've got probably on our play, we've got a dozen different business ideas that we want to try out.

Carl Franklin: It's exciting.

Jon Goodyear: The hardest part is just deciding what you want to do and not actually doing it.

Carl Franklin: Very exciting stuff. Jonathan, thanks very much. This is great. I can't wait to use it.

Jon Goodyear: Great, great. I appreciate you having me on the show.

Carl Franklin: Oh, you bet. And 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.