RunAs Radio

http://www.runasradio.com

RunAs Radio is a weekly Internet Audio Talk Show for IT Professionals working with Microsoft products. The full range of IT topics is covered from a Microsoft-centric viewpoint.

Text Transcript of Show #093
(Transcription services provided by PWOP Productions)

Chris Jackson Makes Our Applications Compatible!
January 21, 2009

[Music]

Brandon Wenn: From runasradio.com, you're listening to RunAs Radio, the Internet audio talk show for IT professionals with Richard Campbell and Greg Hughes. This is Brandon Wenn, announcing show #93, with guest Chris Jackson, recorded Monday, January 5, 2009. RunAs Radio is produced each week by PWOP Productions, providing professional media and podcasting services online at pwop.com. You can follow the guys on Twitter at twitter.com/runasradio.

Richard Campbell: You're listening to RunAs Radio. I am your host, Richard Campbell, with me as always my co-host Greg Hughes.

Greg Hughes: That's me. Hey everybody. Richard, how are you?

Richard Campbell: Are you dug-out?

Greg Hughes: Dug-out. I was dug out and now I'm snowed back in. Actually, looking outside it's been shifting between snow and rain so I was able to get out. People most probably don't know I have about a little more than a quarter mile driveway and I actually can get snowed in because of how steep the driveway is on two hills so I've been parking my truck, a 4-wheel drive truck by the way, and I need to park it out by the road and use an ATV to get out to the truck on these days. So you know, it's an adventure, it's a great spot but it has its little minor challenges.

Richard Campbell: For me, it's just shoveling, shoveling, shoveling. It just keeps snowing in Vancouver, I'm going to die.

Greg Hughes: Yeah, it's been doing the same thing here.

Richard Campbell: It's all good. It's nice to actually be home for a change. We did quite a conference run in the fall so it's great to be doing regular old fashion shows the way we always do.

Greg Hughes: Yeah. I think when you're on the road enough, then you get to the point where it's like I don't want to be on the road anymore, I just want to be home. Then you get snowed in for five days and you're like, wow, I wish I was at the conference.

Richard Campbell: I wish I could get out.

Greg Hughes: I think it's the anywhere but here kind of mentality, right.

Richard Campbell: Absolutely. Hey, we got some hilarious feedback from those two Mark Minasi shows we did at DevConnections which were barely shows.

Greg Hughes: Oh yeah.

Richard Campbell: I think we were both mostly just trying to make each other laugh the whole time.

Greg Hughes: They were barely shows but they were -- well, no, they were totally shows, they were just kind of barely organized.

Richard Campbell: I've come to appreciate that Mark Minasi and I are a dangerous mix. We just tend to go off the track instantly when we talk to each other.

Greg Hughes: Well, Mark is always a fun guy to talk to. He has interesting opinions and they tend to actually be very thoughtful opinions. You sort of listen behind the words and even directly to the words at some point. He always has a pretty well-inform and interesting point of view.

Richard Campbell: Absolutely and it was very interesting to me to hear how he thought about going to TechEd EMEA and actually picking in sessions on Windows 7.0. Windows 7.0 traction is really exciting.

Greg Hughes: Yeah. It's a pretty big deal, sort of a different take on how the bait is being managed this time around and stuff. It's pretty interesting as well and really it looks like there is pretty cool stuff in there.

Richard Campbell: All right. Why don't we hop over to our guest here. Chris Jackson is the technical lead for the Windows Application Experience SWAT team. He has worked with enterprise customers around the world to help them investigate and mitigate application compatibility issues, as well as providing instructional training about Windows application compatibility for numerous industry events. Chris has been a software developer for over 12 years, 5 of them spent with Microsoft. His certifications include MCP, MCAD, and MCSD, and he was formerly a Microsoft Windows MVP known in the community for his technical insight and problem solving abilities. He currently resides in Chicago, Illinois. Welcome, Chris.

Greg Hughes: Hi, Chris.

Chris Jackson: Hello, how are you?

Richard Campbell: Good to talk to you again and we met at TechEd EMEA as well as I recall.

Chris Jackson: Yes, we did indeed.

Richard Campbell: At that time, I did not know that the SWAT team exists so I think you probably should start there with. So what is the SWAT team?

Chris Jackson: Sure. With the SWAT team, our goal is to help enterprise customers who are trying to get to a more modern version of Windows and they are currently either coming from 2000 or they're coming from XP and today they could be going to Vista, soon enough they'll be going to Windows 7.0 and we're very closely tied with the product team to try and get that enterprise feedback into the application compatibility equation because while we can go to local best buy just as easily as you can buy all of the things that are available commercially for sale to the general public, it's a little bit harder to understand the impact we have in compatibility in the enterprise unless we get in there. You may not have heard of us because there are only five of us in the entire planet so there are not a whole lot of us. You know, we try and get as much coverage as we can but we certainly just can't be all places all the time.

Richard Campbell: Dude, you're outnumbered.

Chris Jackson: Well, a big part of our goal is to try and help people get better at what they're doing. The thing I would say about my job is there is enough work out there that there's never any reason for me to hold back information and hoard it in order to try and preserve my job. My job is not going away anytime soon.

Greg Hughes: Sure. So what are some of the low hanging fruit from an application compatibility standpoint? What are the common threads that you seem to be addressing over and over again?

Chris Jackson: Well, the big thread probably is no surprise, it's just running as a standard user, running as a non-admin. With Windows Vista and also with Windows 7.0, most people even if they're under the administrator's group don't run with their full credentials all the time and a lot of people, in fact, are using the opportunity to try and get away from being a member of the administrator's group altogether because there are a lot more mitigations and fixes available in Windows Vista and Windows 7.0 to help you get there.

Greg Hughes: Right.

Chris Jackson: But there's still a lot of legacy software that assumes that you can just use do whatever you want on the machine whenever you want because, of course, everyone is an admin because back in the days they used to be and that's probably the biggest problem, it's just going to be addressing that.

Richard Campbell: And we're talking specifically about custom apps built internally to the company. This is not so much the shrink-wrap app world?

Chris Jackson: Depending on the shrink-wrap you've got. I mean, there are certainly funny shrink-wrap apps that aren't in broad distribution, and even the ones that are, they still have some little dusty old corners of code that have been in the product for 10, 20 years that every now and then surfaced themselves as a goal, here is a bug in here that I didn't realized I had. I mean, I found them all over the place. I found bugs inside of the different parts of Windows itself running without permission, very interesting stuff.

Richard Campbell: Yeah, no kidding. Right off the bat, I'm thinking the sort of the classic you're still writing to the global parts of the registry, you're still trying to write to INI files in System 32, you're still trying to write anything to System 32. Are those the usual scenes we're talking about?

Chris Jackson: Yeah, those are the big ones. I mean, there are all kinds of ways that you can write on what we call a Lua bug. Because there are so many ways, it can be difficult to squash them out in some of the complex software that's out there, but for the most part it's not it's not that people are trying to do these incredibly complicated things, it's they need to store a file out somewhere. Hey, here's a directory, it happens to be sitting in program files but it has my name on it so what a good place to write stuff.

Greg Hughes: Right.

Richard Campbell: Right and it's not, yeah, but we've been -- how many years have you guys been preaching this basic requirement? We're not supposed to be doing this; we're not supposed to be doing this for a long time.

Chris Jackson: The analogy I've been using lately is kind of like if you're driving on a road and the road has a speed limit sign that says please go 55, but there's guarantee that there'll never, ever be a police officer on that road.

Richard Campbell: Right.

Chris Jackson: So of course what are people going to do? They're going to go as fast as they want to go and we suddenly put a hole out of police officers out there so that you couldn't go above 55 and that suddenly caught a lot of people off guard. Yes, there are some people that are always obeying the speed limit, but hey, a lot of people didn't read the sign and one of them who did was like why would I because I'm never going to be in a situation where that will matter.

Richard Campbell: Right and suddenly they are in a situation where that matter.

Chris Jackson: Yeah. I mean, I have a lot of empathy for the people who write this stuff because it's hard enough to write software without us expecting everyone to read all the rules from start to finish. What we want to do is help people understand which of the rules really matter and the principles behind them so that you're not going against this huge list of rules but instead saying hey, here is the goal, and if you just sort of keep that in mind you'll probably do pretty well in writing software in general.

Richard Campbell: So just some sort of key core techniques that you need to use to get over those things. Of course, that all presumes that we got a codebase to work from that. We're going to recompile the app, right.

Greg Hughes: Exactly.

Chris Jackson: Sure. We can do a lot of fixing without actually touching that though.

Greg Hughes: So how do we do that?

Chris Jackson: Well, we have had built into Windows since Windows 2000 a shim infrastructure which until Windows Vista really not many other people use, but the idea is we have this technique that allows us to go in and intercept calls to the operating system and run some additional code before we get there, and because we're running additional code we can change what you're asking for. So let's say, for example, that you're trying to write something to the root of the drive. Now, in Windows Vista and Windows 7.0, we have this feature called file in order to see virtualization that covers rights to program files in the Windows directory and in the program data directory so if you're trying to write anything that is not executable, they will automatically catch it and will stick it some place else. It happens at the driver level, it's very transparent and it works really, really well.

Greg Hughes: Right.

Chris Jackson: If you're trying to write to the root of the drive, we don't cover that.

Richard Campbell: Oh, sure.

Chris Jackson: We could intercept that with the shim and say okay, if you're trying to write this path names, C:// datafile.txt. Before I get to the create file API which would create the file in the first place, I can modify that argument and say no, let's put this somewhere in the user's directory instead.

Greg Hughes: Okay.

Chris Jackson: Then we hand it over to the actual API so that before the API would say no to, it can now say yes to, not because we're loosening security but because we've just modified the argument before we handed it over to something the operating system could say yes to.

Greg Hughes: Sure. For an IT person who is maybe on the support and infrastructure side and maybe responsible for making applications work but is not the development stuff, what's the process that the IT person goes through to solve this compatibility problem. Are there a set of standard steps that the five of you who teach people to follow or some critical pieces of knowledge that people can really maybe take away from the show to help them get a good start?

Chris Jackson: Well, that's certainly the thing we are continuing to try and do better at because we have a lot of materials out there today that's design for the person who wants to become an expert at this. Because we had to go from zero information to lots of information, believe it or not, it's easier talking to Microsoft engineers to develop what I refer to as the Harvard level education than it is to develop a trade school education because they can take the technical details and just basically surface them in an accessible way. What we're trying to do now is get to a point where, hey, if you don't want to be an expert but you just want to cover most of the problems you'd come across, what can you do? A lot of that is let's really leveraged the tools we have. The application compatibility toolkit, for example, contains a tool called Standard User Analyzer. We've been talking about Standard User issues, that's one example of compatibility problem. Well, that one is really easy. If you have a problem running as a standard user, that's pretty easy to verify. You know, if you run it and it doesn't work, you right click one of the admin that does, that's probably your problem and if so, then you use this tool, point it at the executable, run it. The tool will catch all the things that are doing bad and additionally will suggest the mitigation for a lot of the problems that it affects.

Greg Hughes: Got you.

Chris Jackson: So now you don't even have to read the documentation. You can just use the tool and that can get you quite a waste.

Richard Campbell: A bunch of this stuff is now totally visible in Vista. I mean you just right click on any shortcut and then you could see running XP modes and so forth. Are those parts of the shims?

Chris Jackson: If that is the same infrastructure, so we have the ability to do -- you know, we've got 700 so much shims in Windows Vista. A lot of those are what we call our specific shims which are shims that are designed for a specific application so the default fuel of compatibility administrator filters that down to about 300 or so we call general purpose shims. You know, things to fix, hey, this thing, if you're doing this thing that's not going to be acceptable, then the shim will align to that fix.

Richard Campbell: Right.

Chris Jackson: That can be fairly complicated. It's good that we have those, but we still haven't documented all 300 somewhat general purpose shim. You know, we're at about 50 that we have fully documented that a lot of users are taking up by name. The next question is how do we take that? Because 300 is still a pretty big number, 50, that would still be a pretty big number and consolidate that into something someone could do and what that compatibility tab exposes, it's the most common one that you would use, one that don't require configuration to get their job done, they are just to be click at, you turn it on and it automatically start changing some behavior.

Richard Campbell: Right. What about the program compatibility assistant. What exactly is this? I've heard about it but I've never used it myself.

Chris Jackson: You don't have to actively use it, it actually pops on all the time but the idea here is, this is probably more of the either a consumer feature or a testing feature, but in essence creating a job object that collects information. We can watch and detect if something bad has happened to an application that would have caused it to fail. One scenario for example is I can watch and see if you try to create a new folder in program files and then you try to put an executable file into that folder. There are virtualization events that will show up in the event log that the program compatibility system can also go look at and we can look at that and go hey, that looks like the behavior of an installer because that's what installers do. They create directories in program files and they try to drop executable files in there. So then we'll ask, hey, it looks like you have something that's trying to be an installer and I didn't spot it because we have some installer detection but it's not perfect. I didn't spot it, would you like me to try again this time running as an administrator which you would need to be in order to install this program.

Greg Hughes: It's kind of like the clippie for application compatibility, right. I think you're trying to do this, would you like to do that?

Chris Jackson: A little bit less sensitive than clippie. I've never heard anyone say, oh, I can't stand PCA.

Greg Hughes: Nowhere near as annoying of course but analogies at least in terms of what it's doing, so that's pretty cool.

Chris Jackson: Yeah maybe, but here's the catch though. In order for it to pop up, it has to watch you doing something naughty. That means you get one failure before you fix it.

Greg Hughes: Right.

Chris Jackson: So from the perspective of someone who is running IT for an entire organization, a) they don't want everyone to have one failure before their program works.

Richard Campbell: Right.

Greg Hughes: Sure.

Chris Jackson: And b) a lot of these fixes that it recommends, you know, a lot of the scenarios that detects our scenarios running as administrator like that's their solution, let's say sort of the big hammer to just sort of pound in whatever looks vaguely like a nail. Yes, it will probably work but there maybe more elegant ways to fix it without giving someone that privilege. So because a lot of people are going towards the non-admin desktop and because people don't like their users to see failure, they'll use that in the test labs in an enterprise environment, if that pops up they'll go, oh, here is something I need to go and investigate, but in production, a lot of people either turn off and you can configure it and be a group policy some of scenarios that, for example, will result in elevation request because in a lot of configurations people just say auto deny elevation request in my standard users, they don't have credentials anyway so why give them a credential box. You know, we also turn off the feature altogether in production and in the enterprise that's somewhat typical once you get to a fairly well-managed point. At home however, you don't have an IT programming your shops, so yeah, one failure is better and then the rest success is certainly better than infinite failures.

Greg Hughes: Sure.

Richard Campbell: Right. It makes sense to me that this is something we figured out in the lab with you do the install, you see the error messages. I mean, the big thing for me is I'm okay to elevate the privileges or to find the reasonable way to solve the created folder and the program folders and add exe into there, but often installers do a bunch of other things that are stuff that doesn't make me happy like

writing to inappropriate places in the registry and so forth so I guess the advantage of the view approach of this is to catch all of those things into which ones am I going to tolerate, which ones I'm not.

Chris Jackson: Yeah, and you could pull apart, there's actually a registry location where you can see exactly what shims have been applied because the fixes that PCA uses, they're also shims so we can apply shims be at a compatibility tab, we can apply shims via the program compatibility assistant, or you can also create your own custom shim database which you can use in an enterprise-wide deployment. So rather than just a couple of registry entries, you can specifically target it and wired it in exactly the same way that we deploy a custom shim database with every copy of Windows that's out there in the world because Windows comes with around 5000, 6000 apps pre-shimmed to address known compatibility issues in commercial software.

Richard Campbell: And of course the challenge here is that, yeah. Shim, I think, to some folks sounds bad. It really means that we're -- it's like a work around to a problem which is really what it is. It's just that these are not bad work around, they are effective work around and it makes no sense and it's not likely that these apps are going to get fixed.

Chris Jackson: Yeah and it kind of depends on a lot of variables as to when you would want to use this approach. So when I talk to an enterprise customer about, hey, when are we going to use this, are we going to use this, the first barrier is obviously, hey, I've never heard of this, I don't know what to do with it, why would I be using this mysterious technology that no one knows anything about. Well, we use it a ton. We use it in XP to get all of the nine next software working, so tons and tons of games in there that are like ship from XP, and we use it in Vista. The big thing we hit with Vista is all the UAC and the standard user issues so we did a lot of work there and that's where you find most of these hopping up. So we explain what that is then we start thinking about when would I want to use it. Now, if I need support from my vendor, well, we think about it. I have an application that I know doesn't work because if it work, they wouldn't need a shim.

Richard Campbell: Right.

Chris Jackson: If it's known not to work, what are the odds the vendor supports it?

Richard Campbell: Probably pretty low.

Chris Jackson: Exactly. I've actually come across exactly two commercial pieces of software where the customer got support for the shim version. One of them was because this particular customer represented about 65%, 70% of the market share for the ISV.

Richard Campbell: Right.

Chris Jackson: And a little bit of poll with them, and the other was a case where v6, the update for Vista compatibility was just a custom shim database so they actually use that as their solution. We don't necessarily recommend that but here they fully supported the thing as long as you install those shims that will support it. For the most part, you don't get support.

Richard Campbell: Right.

Chris Jackson: So you start thinking about it as either the vendor is gone, like they've gone out of business so I can't get an updated version, the source code is gone, no one can get it and so I use this to keep it working for now.

Richard Campbell: Right.

Greg Hughes: Sure.

Chris Jackson: Typically in that case, I'll start thinking about, well, if the vendor is out of business, this really is bringing it to surfacing the fact that vendor is out of business and that kind of represent some long term risks for me so perhaps I want to start thinking about what I want to do to replace that.

Richard Campbell: You're migrating away from it.

Chris Jackson: Right, because if anything comes up whether it's compatibility or security or whatever the case maybe, nobody is there to fix it.

Richard Campbell: Right.

Greg Hughes: Yeah. I've been in situations where even if the vendor still exists, from an IT management standpoint you have a version of sales force application. Automation application is one real example where you end up with application compatibility problems as you change OS version and I have had to make decisions because of financial reasons that for some extended period of time I'm not going to be able to upgrade the software, the vendor software. I have to find some way to make it work on the operating system that I'm deploying across my enterprise. In that kind of situation, shimming and using the application compatibility tab in Vista and things like that really is a pretty useful thing.

Richard Campbell: Definitely I think the common scenario of this is there is an old version of a product and for whatever reason you're stuck with that old

version, you can't upgrade, you're not willing to upgrade, for whatever reason you're staying with that version of the product, and that the vendor's answer is upgrade.

Greg Hughes: Regardless of what upgrading breaks, right.

Chris Jackson: That's very common and that's the challenge and a lot of conversation that a lot of people haven't thought through completely, but when you start to think about it, the one thing about software that a lot of people, you know, like certainly wouldn't necessarily get this feeling from everybody, but software is not like milk.. Software doesn't expire, doesn't go bad.

Greg Hughes: Right.

Chris Jackson: So I need to think about someday it's just going to go away. This could be and should be here for a really long time. I also then think about scale. A lot of customers have thousands and thousands of pieces of software. I've seen as high as 92,000. If now upgrading my operating system meant re-buying all 92,000 pieces of software, nobody can afford that.

Richard Campbell: Right.

Chris Jackson: So then we start thinking about what do we do about this. I know that software doesn't expire and that's okay. I want to manage my risk and I want to manage my cost. So the big conversation is I like to -- if you have just tens of thousands of software out there, be thinking about reducing your app proliferation. Ninety-thousand applications are too many apps.

Richard Campbell: Yeah. You have other issues. Compatibility is a small one in these set of issues.

Chris Jackson: Yeah, and it's a big one. I want to be actively managing. When do I want to upgrade, rather than just sort of saying, hey everyone, just go upgrade all these things. I want to be pretty intentional about this. I don't want to let things get 17 versions out of date.

Greg Hughes: Yeah. At the same time, every time that I go and buy a new brand of milk, I don't want to upgrade my refrigerator. So there's a balance to that, I suppose.

Richard Campbell: I think that was meant for abuse.

Chris Jackson: Exactly.

Richard Campbell: I think that was meant for abuse right there.

Greg Hughes: I think probably so, but it's true.

Richard Campbell: The other side of this of course, I think we're really talking about a lot of internal apps but this must get really hairy when you're talking about companies that are shipping apps to their customers and they've been dealing with those kinds of compatibilities. Maybe they're shrink apps after all, but I guess those guys are going to be on top of this more of the time. They're doing their testing upfront, but do you get called in on crises like they've done their testing or miss something and now they deploy the version and what do we do.

Chris Jackson: Oh yeah.

Richard Campbell: I remember the pain of doing a 10,000 plus deployment that the prospect of rolling back was so daunting that finding a way to patch that deployment even though it had a problem would be the preferred outcome.

Chris Jackson: Yeah and that certainly happens to me and there are some customers who are, you know, they are not really ISVs, I mean they're business who are designed to produce some product or deliver some service but they happen to have software that they share so they have some attributes of being a software company without actually being a software company.

Richard Campbell: Right.

Chris Jackson: They're sort of looking for guidance on how do I test this on any configuration that any customer I have could potentially have. How do I sort of help manage that risk and do things in a way that's going to help me minimize any of the disruption of my customer's business.

Greg Hughes: Right.

Chris Jackson: The other aspect is how well have we taught people how to write compatible code so that they don't have to catch these things in testing and learn everything, you know, everyone doing it independently. Can we do a better job of saying, hey, here's how you would do this? Because I think one of the big gaps that we have on our developer documentation is you can't necessarily get all of the information you would need in a language you understand. I can talk to you about how do you check for the existence of this feature in C++. I'd be hard pressed to find this same example in C# or VB.NET which is what a lot of people are using in the enterprise world to build their software.

Richard Campbell: Right. Although you'd think that a lot of -- as soon as you're working in a managed language like that, a lot of these rules are automatically complied with. Because you're running in a managed world, that sort of set of problems are managed.

Greg Hughes: As long as you're running properly in the managed world.

Chris Jackson: I think 10% of them certainly are, but not a hundred percent. I can still pass a file argument of C: my programming file which is a) a hard code attack, and b) a protected location and doesn't prevent me from doing that. I mean all the hit issues or taking dependencies on internal implementation, those tend to go away.

Richard Campbell: Yeah.

Chris Jackson: But there is still plenty of ways to end up with something that could cause you a problem.

Greg Hughes: The automation and some of the rapid ability to write good code doesn't prevent people from necessarily writing crappy code.

Richard Campbell: These are also things that I think that are almost out of the curve view of the average IT guys. Some of these hard coded test into the app, we're at a very strong tough disadvantage to deal with that.

Chris Jackson: Yeah, there are a lot of examples of enterprises that will have installed to SQL like Win NT because they just can't move over because of all their hard coded test.

Richard Campbell: Yeah. That's frustrating stuff. I don't think there's any -- are there any good answers there? Are there virtual directories we could be building that will overcome that?

Chris Jackson: There are certainly work around in mitigations where I could create a directory junction there then actually installing it there and result in hard coded test. That way I can target specific applications with the correct file tab shim. I mean, there are numbers of fixes. The more important solution would be to get people to understand how they can get those tests without hard coding them. Today, it's actually fairly easy to do that. The challenge is we kind of keep changing the APIs around a little bit particularly in the native code world. Vista introduced yet another way of getting special directory that's new, but only works in Vista so probably no Win could use that. There's a little bit of sort of catching up and they're good reasons why we do that but I can't fault the developer for not knowing which one to pick.

Richard Campbell: Yeah. Those are tough options there. So Chris, any specific points around what Windows 7.0 is going to do differently from Vista as far as compatibility is concern?

Chris Jackson: That's the question that I get a lot. It's hey, initially from the side of Vista and compatibility and hey, this is really hard and then all the Windows 7.0 information starts coming out with PDC and there's all the speculation on when the next bit of information is coming out. People are kind of on the fence and they're sitting still. The story with application compatibility on Windows 7.0 versus Windows Vista is essentially this. There's no new special magic sauce coming in Windows 7.0 that's going to suddenly make all of the applications that stop working in Windows Vista start working again.

Richard Campbell: Right.

Chris Jackson: It just doesn't exist because what causes them to have problems is the new security environment...

Richard Campbell: It's really the enforcement of the rules we've told, like you said, with the whole speeding thing. We're now having those rules enforced and so those rules still are true and you're going to have same issues whether you're in Vista or in 7.0.

Greg Hughes: Yeah, that's good reason for doing that so we're going to do it.

Chris Jackson: Yeah. I mean, it's a little bit harsh to do it but people want those rules. If you look at what people are saying like hey, I don't want everything, and this is sort of like the conversation I had with people a lot, it's hey, we're not trying to restrict your ability as the administrator of your computer to do everything, but you delegate your authority to every single file you click on. So what we're trying to do is help you pick and say okay, I'm not going to give you all of my authority. I'm going to give you a lesser subset of that.

Richard Campbell: Right.

Chris Jackson: We also want to make it easier for people to not have to be an administrator at all to run software because you shouldn't have to be unless your job is to actively manipulate the computer. You should be able to run without being an admin. So all of these security settings are things that people have been asking for and that doesn't mean that we actually want to go back on. What we'd like to do is get over the compatibility hurdle so we get there. So

the big thing is there's no new special software to fix things that aren't working but at the same time we are bending over backwards to make sure that anything that works on Vista continues to work on Windows 7.0 and not just applications that you fix yourself. One of the demos that has been done in a couple of keynotes now is an app that I wrote as a shim demo where we take it on Windows Vista, we apply a bunch of shims which I use in my sessions to teach people how to use shims, but we take that shim, that version, we take the actual STV shim database files, takes on a USB key, move over to Windows 7.0 key, install it there and boom, the same app which continues to not work on Windows 7.0 with the shim database that has been in Windows Vista is still fixed.

Richard Campbell: Nice.

Chris Jackson: So any investments you do in fixing your applications either by changing code or by shimming them should continue to apply in Windows 7.0. It's a huge goal and the exceptions would be either if you're doing something really crazy because there are a couple of deprecations but you have to really kind of almost be trying to get something to be fragile to get some of the accepted regressions or if you're really just taking a dependency on the internal implementation. I mean, there are a couple of pretty well-known apps that are having some problems right now because they are assuming that this function will always be located at this address in memory when this program is loaded and all of these really sort of low level like hey, if we even change a line of code we break your app type of things. Those will break.

Richard Campbell: Right and there are a dozen of things you shouldn't have done in the first place once again. You know, we hear about Windows 7.0 changing UAC giving it more granular control but I don't think that has much to do with Windows compatibility per se. It's more about the warnings, right?

Chris Jackson: Well, what about the UX, but if you look at the slider, and my favorite way of referring to this is Chris Pankow is a senior PM on the UAC team describes it as he likes to have the UX for the slider be when the slider is all the way at the top where it's most secure, the guy has his pants all the way up and as you pull the slider down, the pants go down. But the settings that you have in Windows 7.0, with one exception, are all available to you via group policy in Windows Vista.

Richard Campbell: Right.

Chris Jackson: The exception is you now have the ability to say, hey Window, if your component sign with the Windows certificate, I can elevate entirely.

Richard Campbell: But that really is the only difference.

Chris Jackson: That's the only difference. The rest of it, if you just wanted to just sort of simulate elevate every thing which is an option on the slider, you don't have it in the slider, you just have the on and off button in Vista but you can go in the group policy and turn it on and say auto-approved elevation request. It just means that all it takes for someone to get elevation of privileges is to ask for it. So you're certainly less secure but you had that same option on Windows Vista as you have on Windows 7.0. You can turn on or off to secure desktops, you have all this configuration but was kind of buried in group policy where a lot of people didn't really sort of understand what it was or how to operate it.

Greg Hughes: So for people who want to find out more and find resources maybe on Microsoft's website to learn more about application compatibility and take a dive into this, where should they go?

Chris Jackson: Well, there is the TechNet application compatibility center which is part of the whole springboard series, you know, the information consolidation, that's available in the URL. The short wind to that is just TechNet.com/appcompat and then I have a blog where I kind of go into, you know, I answer questions that come to me via email, I talked about things that sort of come up as FAQs, I do some talking about shims that aren't yet documented, sort of pre-documentation if you will, at my blog which is blogs.msdn.com/cjacks.

Greg Hughes: Great.

Richard Campbell: Chris, thanks so much for coming on the show.

Greg Hughes: Thanks Chris.

Chris Jackson: Oh, it's my pleasure. Thanks for having me.

Greg Hughes: Good to talk to you again.

Richard Campbell: And we'll talk to you next week on RunAs Radio.