



RUNAS RADIO



<http://www.runasradio.com>



Richard
Campbell

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.



Greg
Hughes

Text Transcript of Show #075
(Transcription services provided by [PWOP Productions](#))



Michiel Wories Uses Powershell with SQL Server!
September 17, 2008



Michiel Worries Uses Powershell with SQL Server! September 17, 2008

[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 #75, with guest Michiel Worries, recorded Thursday, August 28, 2008. RunAs Radio is produced each week by PWOP Productions, providing professional media and podcasting services online at pwop.com.

Richard Campbell: Yes, you're listening to RunAs Radio. I'm your host, Richard Campbell, and with me as always, my fine friend and co-host, Greg Hughes.

Greg Hughes: Here with you for the 75th time.

Richard Campbell: Seventy-five shows, man.

Greg Hughes: Yeah, it's sort of like a landmark-ish kind of thing, isn't it?

Richard Campbell: It's getting there. Well, they say the big one is a hundred shows. If you make it to a hundred shows, you can do anything, so we're getting close.

Greg Hughes: Yeah. That will have to insane in some way, shape or form.

Richard Campbell: I definitely -- you know what? We should take suggestions. What do you guys want to see for a hundredth show? What should we do? Send us an email, info@runasradio.com. I would be remiss if I didn't call out the TechEd Europe sweepstakes going on over at .NET Rocks!

Greg Hughes: Yeah, what a cool thing. Win a trip to TechEd EMEA, Europe, Middle East, Asia, right?

Richard Campbell: Right.

Greg Hughes: You could choose this year or next year.

Richard Campbell: Right, because the contest runs almost right up to TechEd itself. You may not be able to take advantage of it right away so they will carry it over for the following year. Air fare, hotel, admission...

Greg Hughes: Very cool.

Richard Campbell: All you got to do is win the sweepstakes which means answering questions about .NET Rocks!

Greg Hughes: Now, is it for both weeks or do you get to choose the week or is it for dev or...?

Richard Campbell: You get to pick the week. It's one week or the other, so you can go to the dev week or the IT week, either makes them happy.

Greg Hughes: So the IT week is the first week, huh?

Richard Campbell: Yeah. This year, the IT week is the first week because I think they want to get away from the PDC in Los Angeles.

Greg Hughes: Ah, makes sense.

Richard Campbell: A long trip for us because I'm doing PDC and then we're going to Barcelona to do the IT week and then I've got to get back for DevConnections in Las Vegas.

Greg Hughes: That's right.

Richard Campbell: Mark Dunn is going to be there for the dev week. We're holding up the Pwop banner so we're still be represented on the dev week in TechEd Europe.

Greg Hughes: That's great.

Richard Campbell: All right, Greg, let's get to our guest. Michiel Worries has joined Microsoft 7-1/2 years ago in the role of Senior Program Manager for Microsoft SQL Server and is currently working as a Principal Architect on defining the next generation SQL Server management platform infrastructure. Before he joined Microsoft, he has spent a large part of his professional life at a large ERP company where he was responsible for the architecture of the tools and runtime and development environment in his capacity of Principal Architect. Michiel has worked on various large-scale development projects on UNIX, Windows and various other platforms. His hobbies are photography, hiking, independent movies and classical guitar. Oh, we got to hook him up with Carl.

Greg Hughes: Very cool.

Richard Campbell: Most of which are suspended because he's got a 2 year old. So, that will take over your life. Welcome Michiel.

Michiel Worries: Yeah, thanks for having me.

Greg Hughes: It's good to have you.

Richard Campbell: So, originally from the Netherlands, but living in Redmond now?

Michiel Worries: That is correct. I came over here seven-and-a-half years ago.



Michiel Wories Uses Powershell with SQL Server! September 17, 2008

Richard Campbell: Wow.

Michiel Wories: Especially for this job and I happen to like the environment as well.

Richard Campbell: Well, it is a fun place and the weather is not that much different I think between Seattle and Amsterdam.

Michiel Wories: It's actually quite close.

Richard Campbell: Yes, it's very close.

Michiel Wories: A mixture of rain and sunshine.

Richard Campbell: That's right. If you can't take the clouds, you're in the wrong town.

Michiel Wories: Oh, I can take it.

Richard Campbell: PowerShell, do we have PowerShell for SQL Server now or is it still coming?

Michiel Wories: It is in SQL Server 2008, just newly released and it includes SQL Server PowerShell.

Richard Campbell: Awesome. So full meal deal, everything you ever wanted to do in SQL Server, we could do through PowerShell now?

Michiel Wories: Basically, yes, and I start to think about when PowerShell development started and you wouldn't believe it, it actually started eight years ago and that sounds kind of an odd thing to say because I don't think PowerShell development started eight years ago, but it actually started with the development of the XML object model and as you know PowerShell is really an object-oriented environment which could do scripting and what it really applies on very well is if there is a solid object model underneath and this XML object model which I actually designed and co-developed myself forms the basis of this provider and that work started eight years ago.

Richard Campbell: That object model actually got surfaced in a few different incarnations as well because there's been a number of, do they call it DMO? What was the original -- there was a library in SQL Server to access that object model as well.

Michiel Wories: That is correct. The story behind that is that at the moment that I came in actually, that was seven or eight years ago, there was an existing object model called SQL DMO. It was based on COM and as you know, at that particular time, .NET was starting to be phased in and it didn't seem to make any sense for us to continue the development on that old platform.

Richard Campbell: Right, because this was back in the SQL Server 2000 timeframe.

Michiel Wories: Just at the beginning of 2005 development.

Richard Campbell: Right.

Michiel Wories: Because indeed Mysql Marsden had been released already. I think it was two or three releases in some form or shape.

Richard Campbell: Yeah, it did get out into the wild. We've used it but it's like the last big COM library to come out of Microsoft almost.

Michiel Wories: Oh yeah. So, we decided to actually completely rewrite it instead of just shaping the old object model which, you see, is a study that sometimes is used. We decided to go for completely brand new development and developed an entire new object model with particular characteristics that our users want to have from it and so much of its characteristics were which already exists in DMO by the way, ease of use, discoverability. It needs to be an object model that is casually used or usable so to say.

Richard Campbell: Because the reality of course is that folks don't use it every day and they need to be able to just pick it up when they need it and do the things they need to do, but is this really something for developers?

Michiel Wories: It's a mixture of the two. It needs to be casually usable by certain group of users that don't use it every day and use it in scripting environment. On the other side, it needs to be rich enough to host an IC application, for example. There are several vendors out there that build their tools on top of an object models so it needs to serve both different groups but if you were to make a feature decision on a particular set of objects, then we can make it first workable for the casual user and then we'll make it work for the, let's say, advanced users.

Richard Campbell: Right.

Michiel Wories: We got several user models over here and internally are called Mort, Elvis, and Einstein.

Richard Campbell: Ah, yes.

Michiel Wories: And we would write to Mort first. If I have to make a decision between Mort and Elvis, we will go for Mort.

Richard Campbell: Okay.



Michiel Worries: And that's been really a key, let's say, driver for XML object model and that turned out to be a key asset in developing the PowerShell extensibility because guess what? When I started to think about designing a PowerShell extension on top of the SQL Server object model, it became really apparent to me that because all of the objects are extremely usable, discoverability is really easy, a very good way of serving these objects through a provider.

Richard Campbell: Right.

Michiel Worries: So, what you see in our environment, the SQL Server environment, is that you see not a whole lot of commandlets and the main reason is the objects are by themselves extremely usable and there is less need to develop let's say hundreds of commandlets. Secondly is that if we were to go through the commandlet route and provide a command for every possible path that could be done, we'd literally end up with a couple thousand commandlets most likely.

Richard Campbell: Right.

Michiel Worries: Then usability is probably out of the door because it's not very discoverable. So, what you see right now is a provider that gives you extremely easy way to discover SQL Server, just start it up, it starts in the right place, UCD into a machine and from that machine you can see the instances. You can UCD into an instance and then you see the whole hierarchy as SQL Server exposures and that turned out to be -- from my user perspective, we're initially somewhat skeptic to not have commandlets.

Richard Campbell: Right.

Michiel Worries: When they saw the strength of having a set of objects at your fingertips that are extremely usable, that immediately turned around and when we showed this for example to a set of users and MVPs, some of them call it an elegant design where you marry the two things, ease of use of the object model, very easy discoverability of those objects.

Richard Campbell: But the commandlets -- I mean exchange which of course I think was the real driver behind powershell more than anything is very commandlet-focused that that's the way they go about things but I guess the big difference when it comes to SQL is I have the SQL language and odds are if I'm doing this job, I know it so isn't the main thing I want to want to do, once I can get authenticated to a database, is issue SQL commands?

Michiel Worries: Not necessarily, no. The reality of what -- well, most of it.

Richard Campbell: Okay.

Michiel Worries: You would see users who would stay in PowerShell and use the objects only to perform operations on manageable entities such as a database or a variable or an index. However, what we also recognize is that there's going to be a ton of people who have a lot of scripts still in their back pocket to do operations and we said, "You know what? Let's make it very easy to integrate these scripts in a PowerShell environment where they could enumerate their servers, their instances, their databases with ease but within this context use these scripts that they already have developed before a provider actually came to life."

Richard Campbell: So rather than invoking that script through OSQL, I'm be able to do it through PowerShell without having to rewrite it.

Michiel Worries: Correct and what we also need to enable this, we will make sure that there's a deep integration between the commandlets that allow you to execute scripts in the PowerShell environment itself. For example, if you were to log in using SQL security in SSMS, you start a PowerShell from the database node that you want to manage.

Richard Campbell: Right.

Michiel Worries: You get a PowerShell integrated security, sorry, using exactly the same security context. Now, if you didn't start the commandlet called invoke SQL command from that particular place, it will yet again take over the security context that you already established and it makes it a whole lot easier and very intuitive to work in an environment that passes your credentials all the time, you don't have to think about it anymore.

Richard Campbell: So, effectively, you don't need to explicitly log in. The context that you're running in is all you need.

Michiel Worries: Correct.

Richard Campbell: That's a good thing but I also have the option of explicitly logging in as well.

Michiel Worries: Absolutely, so you can overwrite the settings if you need to overwrite those. We make sure that flexibility is there to do whatever you want within the context that you want to specify.

Richard Campbell: And I definitely could see scenarios for both. I want to save the effort of not logging in all the time combined with really I would think in the context of an elevation that my normal running mode has some additional privileges and then



maybe I log in as a higher administrator mode to do something a little more drastic on the database.

Michiel Worries: Correct, yes.

Greg Hughes: Is there an ability to turn that capability off? I'm thinking from a security standpoint. There might be times when I simply want to be as secure as possible. I don't even want that to happen.

Michiel Worries: Yes, you could actually disable that feature if you wanted to and then you would just go in using integrated security.

Greg Hughes: Got you.

Michiel Worries: So, there are ways to ever have thought about the security aspects of that as well to disable that particular feature to make sure that your script always runs in the context that you expect it to run in.

Greg Hughes: Got you.

Richard Campbell: I love the fact that I don't have to change the existing scripts I've got to utilize them in PowerShell as well. That's just a ton of work saved.

Michiel Worries: Yeah, that was a big thing for us to not have people throw away all of their existing text and SQL scripts.

Richard Campbell: Yeah.

Michiel Worries: It's a long run. It's not only about Transact-SQL. This is really about standardizing the language with which you will manage SQL Server. Right now, the provider is an extensible provider, by the way. The provider is implemented in such a way that actually doesn't necessarily know that it's actually working within SQL Server or within the policy based management system. It's very, very generalized.

Richard Campbell: Interesting.

Michiel Worries: And we actually can plug in new providers in our own provider infrastructure fairly easily and an example of that is we have an object model that have a data collector subsystem which collects performance metrics for users for whatever purpose they want and that subsystem has an object model that we want to plug in at the last moment. There was a discussion that lets add this to PowerShell. I think I had only week left of development time but it took me 10 minutes to plug it in and it worked right away.

Richard Campbell: That's nice.

Michiel Worries Uses Powershell with SQL Server! September 17, 2008

Michiel Worries: Literally 10 minutes, not to data debug the thing because some things broke but here's the thing, the provider model has been built to be extremely extensible.

Richard Campbell: Right.

Michiel Worries: Our whole vision around the object model is also based on that notion to be able to standardize the management operation. If you look now at any given object that exists in different management spaces, whether it is SQL, policy based management, data collection or other spaces; you will see that most of these objects have the same look and feel. They all have a standard create method. They all have the same metadata. You can actually ask the objects about a hierarchy, what child object it has and so forth and so forth.

Richard Campbell: Right, where it sits in the hierarchy.

Michiel Worries: Exactly. We've all been standardizing that and in fact there is an object query system that works the same across all object models and in the coming releases, you will see that we will expound on that vision. PowerShell is going to be a pivotal element in this where you will see new providers being plugged in or released. For example, in the next release, I expect there's a focus on analysis service. We might take on a couple of other ones and this will just appear as one of the namespaces so to say in SQL-PS and the idea is that all these object models will take on the standardized look and feel that enables easy extensibility again for us and at some point, we'd like third parties to plug in their extensions as well.

Richard Campbell: Sure. Yeah, that will be very compelling. I also like the fact that you're going to be able to ship new features without having to ship a new version.

Michiel Worries: Correct, yes, that was also one other thing, we didn't want to write a provider or a new version of the provider whenever a new server pops up.

Richard Campbell: Right. What it does is it mitigates fear. For me as an administrator, every time I upgrade a version, I take a chance that stuff breaks, but adding a new provider in, that's just not a big deal. It only deals with new things. It's not going to hurt what I've already got.

Michiel Worries: Correct, yes, and that's one of the goals.

Greg Hughes: You don't have to worry about breaking the contract, yeah.



Michiel Worries Uses Powershell with SQL Server! September 17, 2008

Richard Campbell: Well, it's all a part of that very suspicious mindset we need to have with people bound by SLAs.

Michiel Worries: So, this environment is really focused on DBAs, IT pros, casual users, but it enables cross server and cross service management. One of the things that we got from our initial user study, one of the things that I got back from a user base outside of Microsoft is that people would like to manage more things by SQL Server only. They are not interested in learning yet another Transact-SQL or other statement.

Richard Campbell: Right.

Michiel Worries: They are more interested in managing their servers and this kind of abstracts it away from them. This allows us to provide a common API that everybody talks to in the same manner and reduces ownership and yes, that's one of the goals as well. So, for our future releases, we have pretty big plans in that perspective. We'd like all of our object models to emit PowerShell.

Richard Campbell: Right.

Michiel Worries: We like the SSMS dialogue, the GUI dialogues to emit PowerShell instead of emitting Transact-SQL or NS2 actually. We'll still be able to emit Transact-SQL but PowerShell would be the primary choice most likely.

Richard Campbell: You're actually seeing all of it. This is something that IIS 7.0 did where essentially the IIS 7.0 and management console is firing PowerShell under the hood. Do you see you're going the same way?

Michiel Worries: That's been under debate right now. Yes, we'd love to do that. There's a whole lot of an infrastructure that we might need to change in order to enable it but that's actually one of the goals in the long run. Also, we'd like to embed the PowerShell inside the engines so you can imagine you write the scripts for the client and we think it also should be runnable inside of the server so you don't need to switch to Transact-SQL if you're going to do something that's inside of the server. Another thing, what we're thinking about is integrate the PowerShell editor into our management environment executing as well so you can execute a PowerShell script and getting the results in the DataGrid as you know it right now if you've seen Visual Studio.

Richard Campbell: Yes.

Michiel Worries: There are a lot of plans and a lot of work ahead of us to provide new features.

Richard Campbell: What object models are covered right now in this first release? Is it just SQL Server? What about analysis services, SSIS, and so forth?

Michiel Worries: Currently, we have four object models that are covered by this. There is SMO that manages SQL Server.

Richard Campbell: Right.

Michiel Worries: DBM or DMS as it sometimes called which is, what is it? Desk management? You can actually create, enumerate policies, and evaluate them as well. The Data Collector Management object and lastly, probably it is the identity servers that reside on the client or on the central server.

Richard Campbell: Okay.

Michiel Worries: The interesting thing there is that that allows you to stay logged to your PowerShell scripts up to whatever amount of service you put into that. I've been writing a couple of samples for our sample database, for example, that enumerates this collection of registered servers on a central server and performs operations all over them. One of the goals with this environment is also to allow people to scale up this type of management scripting to hundreds or thousands of servers.

Richard Campbell: Wow. That was one of the visions behind PowerShell was this ability to use enumeration and looping to sort of pull in larger sets of data whether it was manipulating users or manipulating some other class of objects and I guess servers would qualify as well.

Michiel Worries: Yeah, exactly that and in the next release, you will see us focus on that quite heavily actually.

Richard Campbell: So the server registration library is interesting. Maybe we need to define some of these other object libraries. Data Collection, what's that about?

Michiel Worries: The data collector is an environment that is typically used to collect runtime data on which you can do reports. There are a couple of standardized reports and collections being shipped to monitor SQL Server performance-based data. It stores the data in a data store that's queryable and allows you to, for example, do growth projection or do capacity management to look at your servers through the performance angle.

Richard Campbell: Is this another way of getting at the data management objects, the queries that



Michiel Worries Uses Powershell with SQL Server! September 17, 2008

essentially tells the state of the database and the load that it's under, that sort of thing?

Michiel Worries: Not quite. It is roughly a -- how would I explain that? I'm not a specialist here, I have to warn you.

Richard Campbell: Oh sure.

Michiel Worries: It actually loads up data from SQL Server and currently uses parts of DTS to do that.

Richard Campbell: Okay.

Michiel Worries: And stores it in a particular set of data tables. That data can be queried or reported on to do performance analysis on your data.

Richard Campbell: Right.

Michiel Worries: There's not much I can say about it. You're probably not talking to the right person here to get more detailed information.

Richard Campbell: All of the details on it, but it's interesting to see that that's exposed because there are lots of potential there for tying in monitoring mechanisms, ways to analyze data server. That's a very usual chunk of data.

Michiel Worries: Oh yeah and actually we will be starting on focusing on making that data more usable in the next release.

Richard Campbell: Oh yes.

Michiel Worries: By building a modeling environment around that so you can actually reason better over the data. The data that's currently stored is kind of the raw data and we're going to introduce models around that data so you can reason over the data, for example, if you see a CPU spike or a disk IO spike in disk usage, then you should be able to analyze this not by tying things to get us to a code which often happens...

Richard Campbell: Right.

Michiel Worries: But by looking at the metadata and seeing what possible causes can be and actually dive into this particular data source to get more information.

Richard Campbell: Where that would get cool is when you start drilling against profile, what's currently running, event logging, it's that combination of data that really reveals what's going on.

Michiel Worries: Exactly that. So, a correlation of data, building information, let's say you could imagine you should be able to read the data and the metadata as provided to build a cube around your information as well to do analytics on it.

Richard Campbell: Right.

Michiel Worries: So, there's some early thinking on it right now but it's about as much as I can say about it as well.

Richard Campbell: Yeah. There's a lot more. The fun part here is I think you've made a really cool version but it's apparent that there's a lot more to do. SQL Server is such a rich product. There are a lot of libraries and a lot of things that could be done in PowerShell. I start thinking in terms of -- I'm a big OLAP buff and so I start thinking about what I could do from a PowerShell perspective to get at OLAP data and to help generate a cue. It's a little lower level way of going about it but sometimes you don't want the UI, you want a different way to get that stuff created.

Michiel Worries: Yeah and that's what we started to unlock in the next release or releases. We'd love to make all of these services available through this provider model. Now, that said, it doesn't mean that we're going to completely shy away from providing commandlets. We'll listen to the customer base and see why we have to build commandlets for the most common use cases that might not be so intuitively done through the object model. So, we will try to strike a balance between having a strong object model but also having a set of commandlets available for the most common scenarios and the key really here is reduce the amount of time that people have to spend on scripting.

Richard Campbell: Sure but you don't want to have thousands of commandlets either, that's not easy to manage either.

Michiel Worries: In the end, these commandlets would be merely shims around their object model.

Richard Campbell: Right.

Michiel Worries: The object model, if you look at individual methods, take a table and there's some methods that validate the data, and that will probably be a commandlet, right?

Richard Campbell: Right.

Michiel Worries: And that would merely be a shim; however, a commandlet development is fairly expensive compared to expose of an object though



the provider because each commandlet needs a separate help text file. It needs separate testing.

Richard Campbell: Right.

Michiel Worries: Whereas, the object model already exposes most of the functionality at the level that the user expects it to be exposed.

Richard Campbell: And it's already gone to the testing and so forth so you were able to ship more and less time that way.

Michiel Worries: Exactly. It will be an extreme cost burden actually and it will take away from all the features that we've left developed. So, it turns out that having a very strong object model which is very usable turned out to be a key asset in this particular design of SQL PS and I'm eagerly waiting to hear the users' feedback on what their experiences are going to be. So far, responses have been pretty good and people like the intuitiveness of it. We also made sure that whatever we expose at the object level looks very similar to what you see in the UI. If you were to do a *dir* at the database level, you will see the same field that you would normally see in the middle pane. So we try to make things look and feel quite the same. The object in Explorer 3 is very much the same. In the end, we like them to be the same, exactly the same as PowerShell.

Richard Campbell: So you don't notice any differences.

Michiel Worries: Right.

Richard Campbell: But I think the big thing is if you're familiar with the UI environment, the management tools in SQL Server, you know the tasks you want to do, you should be able to discover those in PowerShell as well.

Michiel Worries: Uh-hmm, yes, and integration with the UI was a key thing for us as well. You can actually walk up to the UI, right click on the node and say, "Well, let me start doing something with PowerShell from this node on," and it will bring you to the exact context that you have in the UI as well. So, you don't need to CD into that space. It's already there and I think that, let's say bridging the gap between UI and scripting is extremely important to me.

Richard Campbell: Right.

Michiel Worries: Making sure that that gap is very, very small is going to be one of the focal points for us.

Michiel Worries Uses Powershell with SQL Server!

September 17, 2008

Richard Campbell: You've mentioned this very early on is the fact that you have this choice between using SMO to, say, create a table but could just as easily use your invoke SQL command to send a create table statement. Why both? I always think it's SQL so I think I would tend to invoke a SQL command. Is that really impairment for some folks? Doesn't it require more work for you to do that?

Michiel Worries: No, it does not, by the way, but in our discussions that we had with people, we had many people asking us this particular question and what turned out is that there are two types of users, users that are new to SQL and don't know Transact-SQL very well and they will naturally gravitate to the object model.

Richard Campbell: Right.

Michiel Worries: There are the advanced users that will continue to use Transact-SQL for quite a while. We have to serve both types of users and that's why you see this environment that supports both. There's no magic there. There's no other reason to do that. We would like people to leverage their existing knowledge and not force them into PowerShell if they don't want to but we'd like them to have this discoverability because I think that is one aspect a lot of people will start to use, the navigational aspect of it, ease of use of this navigation environment that is outside of the UI.

Richard Campbell: That we could just browse the object model with the tool.

Michiel Worries: Yup. Even if you don't know SQL Server, you see it for the first time, starting for example VBScript and DMO, it's fairly impossible for most users.

Richard Campbell: Absolutely.

Michiel Worries: I think they would need to learn quite a bit about the object model and about SQL Server to be able to use that; however, if you start SQL PS and you CD into a server, it's immediately apparent what you see there, you see database, you see tables, the objects that are returned. You have to understand something about PowerShell. If you understand that these objects are table objects, then you will understand there are methods on them that allow you to operate on these objects. So, it's a very small step from getting an object to working with it. I'll give you an example, for example, re-indexing all your indexes is a matter of two lines of PowerShell script.

Richard Campbell: Right, as opposed to knowing the command you'd have to pass to invoke a re-index.



Michiel Wories Uses Powershell with SQL Server! September 17, 2008

Michiel Wories: Exactly and knowing that you have to instantiate a particular object in a particular part of the hierarchy.

Richard Campbell: Right.

Michiel Wories: So, it is extremely user friendly from that perspective. It allows people to gradually learn the environment and not having any barrier to step into it. It is like a disk drive. Everybody knows how hierarchies in disk drives work.

Richard Campbell: Sure.

Michiel Wories: It has exactly the same look and feel. We didn't introduce any oddities there. We made it as naturally as possible so that people don't have a learning curve there to learn this new environment. They will almost immediately feel familiar in that environment and say, "Ah, that is a database directory that contains databases. Let me do something with it." Right?

Richard Campbell: Right.

Michiel Wories: So, usability was really a key focal point here. Now, if SQL Server were to be a small service with limited set of objects and limited set of operations, I could have argued that a set of commandlets would have done the same job.

Richard Campbell: Yes.

Michiel Wories: However, that not being the case, SQL Server being an ecosystem of different services, having to exist next to Exchange server, next to other services that need to be managed by IT professionals, this strategy makes more sense in that perspective.

Richard Campbell: Considering the diversity of SQL Server, you bet. So, one of the areas I'm immediately sort of latching on to with PowerShell would be scripting failover scenarios like switching over from a log shipping mode from one to the next. Have we got all that implemented like I'm able to write the script that I could just make the invocation at PowerShell to complete a full failover?

Michiel Wories: Yeah, absolutely, yeah.

Richard Campbell: So, all of that sort of management level stuff and so forth, that's all there.

Michiel Wories: Yeah, this goes back to the initial goal of XML. XML's focus was really to provide full manageability of all the manageability aspects and that includes scenarios that I can't talk about right now. So, there's hardly any area that it is not covered

by XML. If it's not covered, I consider that to be backordered at this stage.

Richard Campbell: Excellent.

Michiel Wories: So, if you look at the new release of SQL Server, all of the services that have been developed right now immediately includes XML support so that every individual team that works on a feature set within the SQL team, and the SQL Server team is by now a fairly large team, takes that almost naturally as a task to include manageability as one of the aspects of the feature that they leverage. From that perspective, the XML object model has been proven to be pretty successful because it scales to a larger organization like this. There are many, many people that need to provide new manageability features.

Richard Campbell: It's also abundantly clear to me that this would be great on a small instance like an express instance of SQL Server on a workstation. It would be nice to release PowerShell scripts there as well.

Michiel Wories: Oh yeah, yeah, absolutely. So, if you're a developer and you don't want to use UI even if you run server, it is still a very useful environment because you've latched quickly to do things.

Richard Campbell: Right.

Michiel Wories: I find myself in development phase also doing this. I also almost automatically gravitate towards PowerShell to do my initial testing, for example.

Richard Campbell: Sure.

Michiel Wories: The reason is I don't have to develop a C# program, compile it, run it, put it in a test server, do initial proof of concept testing using PowerShell. I expect other users to do the same thing, other developers, for example.

Richard Campbell: Also deployment and installation and stuff is just a so much better way to go about this.

Michiel Wories: Yes, so this is one of the areas where I expect this to be used quite a bit as well. Accidentally, I was listening to using PowerShell with VMware, one of the programs that you had a few weeks ago with Hal Rottenberg and Carter Shanklin.

Richard Campbell: Yes.

Michiel Wories: And one of the things that they mentioned was to provision a server and then poking



through the VM actually into these surfaces to configure them.

Richard Campbell: Right.

Greg Hughes: Right.

Michiel Worries: I expect that to be PowerShell as well and this will, SQL PS for example, and all the other providers are being developed or already have been developed by the different server teams will be used to configure this in those types of scenarios.

Greg Hughes: Not only inside of Microsoft but as you're sort of alluding to out in the community, there's a strong community around PowerShell. Are there some specific resources for PowerShell, for SQL Server that you suggest people take a look at in the community?

Michiel Worries: Yes. Well, I don't have them at the top of my head, but I can issue some of the references that I have after the show so you can put them on the show's blog.

Richard Campbell: Yeah, you bet. We'll link to that so people have access to it. Now, I'm certainly going to link to your blog as well so folks can get your comments on where PowerShell is going and get back to you too on their impact on PowerShell.

Michiel Worries: I'd like to get more feedback on it, on the uses of it, so getting more exposure is greatly welcome.

Richard Campbell: Absolutely. Michiel, thanks so much for coming on the show. We really appreciate bringing us up to speed on SQL Server and PowerShell.

Michiel Worries: It's great being here. It's fun to talk about this project.

Richard Campbell: Excellent, and we'll talk to you next week on RunAs Radio.