



<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 #036
(Transcription services provided by [PWOP Productions](#))



**Kim Tripp and Paul Randal Answer Our Queries on SQL Server
2008!
December 12, 2007**



[Music]

Carl Franklin: 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 Carl Franklin, introducing show #36, with Kim Tripp and Paul Randal, recorded Friday, November 16, 2007. RunAs Radio is produced each week by PWOP Productions, offering professional media and podcasting services online at pwop.com.

Richard Campbell: Hi, this is Richard Campbell. You're listening to RunAs Radio. We're here on the last day of TechEd IT.

Kim Tripp: Woo-hoo!

Richard Campbell: Guys...

Greg Hughes: Goodness!

Richard Campbell: Greg, say something.

Greg Hughes: It's been a long week.

Richard Campbell: Yeah, it's been a long week. It's been a long week for you two yo-yos.

Kim Tripp: It's been a long interview and it just started.

Richard Campbell: I think we're going to have to leave the whole warm up to this end just to give the folks an impression. We're sitting here with Kim Tripp and Paul Randal and you're doing 12 sessions this week?

Kim Tripp: We have actually done 11. We just finished our 11th session, although not all of them were break-outs. We did three break-out sessions, four ILLs, our last ILL is today. We did a demo session, a fast paced demo session and then we did, two or three talkshows?

Richard Campbell: Yeah, I see a QA as well? Yeah.

Kim Tripp: Yeah, we did two or three talkshows?

Paul Randal: And a partridge in a pear tree.

Richard Campbell: Yeah.

Kim Tripp: It feels like that, actually.

Richard Campbell: Yeah, you guys were really busy as usual actually. You never go to a conference lightly.

Kim Tripp: No, that's actually true. I don't know why, but it's like they keep asking me, "Hey, do you want to add on another demo session?" I'm like, "Sure." I don't really do the math.

Paul Randal: It's because you're a celebrity.

Kim Tripp: Actually, I think it's the blonde hair.

Richard Campbell: Ah, here we go.

Kim Tripp: I think that's actually the problem more than anything.

Richard Campbell: Fame, well, it does have its trials and tribulations.

Kim Tripp: Oh, all right. Stop. Move on, move on.

Richard Campbell: Okay, okay.

Kim Tripp: No, I'm still having fun. I think that's part of the problem and the benefit.

Paul Randal: We wouldn't do it if we're not having fun.

Kim Tripp: Yeah, absolutely.

Richard Campbell: Well, the two of you are crazy together too. I mean it's just taking the whole thing to another level. As an attendee and a viewer in the audience, it's just been madness with both of you on there.

Kim Tripp: You know, actually, I have to make a joke about this because today we did a session together and we rehearsed, of course, but not real formally. We don't do real formal presentations at rehearsals, but at one point I was going through SQL 2008's DMF, Declarative Management Framework, and I was showing some policies that I had enabled on some databases and I had restored a SQL 2005 database that was actually originally a SQL 2000 database, so it had a database page verification setting to torn page detection. Well, there's a policy that you can set in the DMF that says your databases have to have a checksum instead of torn page and Paul can definitely go into all the nitty-gritty details on why this is so much better if you guys want to talk about that, but the point was I was specifically demoing what happens after a policy is checked and what happens is they'll actually mark the database with a little tiny error and then up the chain, just like the replication monitor that we have today, and you'll actually see all the way up the chain this little error and I'm demoing specifically talking about the error and Paul was doing something completely different.



He looked over and he goes, "What's that little icon that is there?" I'm just kind of standing there on stage and the whole audience is like, "That's actually what she's talking about, Paul?"

Richard Campbell: Yeah, where have you been?

Kim Tripp: Yeah, where have you been? I mean I know you're standing right next to her there, but apparently you're not really there. It's the "looks alive, but not *is* alive." So, I said to Paul, "I guess you weren't listening, dear." So, yeah, we're having a good time. We're speaking together on a lot of topics and bouncing things off of each other and they have a bunch of sessions where they've just listed me and I dragged him on stage and they just list them and he drags me on stage and I think out of the 12 sessions, there was only one session of each that we weren't actually very actively co-presenting.

Greg Hughes: Right. So, speaking of SQL Server 2008, I mean we have the god and goddess of SQL Server sitting with us here, so...

Paul Randal: Where?

Kim Tripp: Yeah, where are they?

Greg Hughes: So, what's coming up in SQL Server 2008? What do IT professionals need to know about?

Kim Tripp: Well, I can start with the DMF. I mean just to continue on that same line.

Greg Hughes: Sure.

Kim Tripp: So, the Declarative Management Framework is really building on some of the past features that we've used and leverage a lot like BPA, the Best Practices Analyzer, the SQL Server area configuration tool, and the agent. So if you think of all the policies that you have enforced by going to the SQL Server configuration tool, checked it, and then basically set it so that it wasn't enabled or at least verified that it was or wasn't enabled, or you've run BPA which unfortunately is a separate download, but when you've run it you've been able to check some of the Best Practices that you should be adhering to in the environment. So, my point is, they've taken these Best Practices, they've taken the SQL Server configuration tool, and they've taken those policies and put them into this thing called DMF where you can actually go to some of these Best Practices like don't run Auto Shrink and that could start off...

Paul Randal: Evil, evil.

Kim Tripp: Yeah, Auto Shrink is evil, but you can basically take the Auto Shrink database property and say that that should be false. You can then target it at a Server or even at a group of servers for that matter and you can have the DMF check to see if the servers are set, generate errors if they're not, and you can even enforce the policy on demand. You can actually click and say enforce this now and it will go on actually change all of the databases to the correct setting.

Greg Hughes: Across even a farm of servers, it sounds like you're saying.

Kim Tripp: Yeah, absolutely. So, you can take in the registered servers dialog a group of servers, manage them as a unit and the agent jobs can actually get set up on all the different servers.

Greg Hughes: What does it mean in the real world? I mean what sucks about previous versions of SQL Server that this really solves? What's the pain that was being addressed?

Kim Tripp: Yeah, I think it's not that anything sucks per se, although database shrinks sucks and Auto Close sucks and not having checksums...

Paul Randal: Newest versions do suck, but...

Kim Tripp: Okay, but it's not -- you know, now that you don't work for Microsoft, you're way too...

Paul Randal: I can say whatever I like.

Kim Tripp: I can't believe that I'm being more politically correct than you these days. That's pretty scary. My point though is there were some things in previous releases that were set for, I don't know if I would say the wrong reasons, but like on lower end editions of SQL Server like Express or Workgroup, there were some even default settings like in Express that were Auto Shrink, Auto Close, and a lot of Developers will develop on a lower end system backup restore and those settings end up getting propagated all the way up into production and sometimes the Enterprise. The end result of that is it might have been fine for a development environment, but it's absolutely abysmal in production.

Greg Hughes: It's a pretty optimistic way of setting things in the lower level if you're really going to have to spread it up to something that has to scale.

Kim Tripp: Yeah, and not everybody knows all...



Richard Campbell: Yeah, nobody thinks that way.

Kim Tripp: Exactly, nobody knows all of the options at the time they're developing or even by the time they go into production. Depending on what their background is, they might not know all of these Best Practices, so bring in them all together in one single central place with recommended settings because that's even -- it's not just how do you want to set this, it's this is how we recommend that it's set. Literally to enforce these policies, you enable it, set a schedule and you're done. I mean it really is rather simple, so I like that aspect of it.

Richard Campbell: I'm thinking from a consultant's point of view, the ability to go in and look at somebody else's database and take your policy set with you and say, "Okay, how do these policies apply to their database?"

Kim Tripp: Yeah, absolutely. You can even add your own custom policies as well.

Richard Campbell: Right.

Kim Tripp: In the future, they're hopefully going to make it even more extensible where you can do kind of more generic even I want to try to enforce a maximum amount of downtime or data loss of X minutes and then all of the things that are impacted by that kind of a problem, and it's amazing when you really think about it, but it's not there yet.

Richard Campbell: No.

Kim Tripp: I mean as far as the...

Richard Campbell: But suddenly you can see a date, you know, how many times have we battled this? We go into the whiteboard, you go into the war room of a company and the VP there says, "We'll have 100% up time." My immediate goal is to say, "All right. Here's 99% up time, here's 100% up time. Here are the differences. Here's the price difference." I go there first just from experience, but to have a tool that would help you. All right, let's look at your current configuration. This is the expected up time with this configuration. Now, 100% look like this. Here's all the things we're missing. Once again, we get back to here's the price of your "I want 100% up time."

Kim Tripp: Yeah, absolutely. We're a little bit away from that, but I think that's incredible.

Richard Campbell: Oh, so I still have a job. Is that what you're telling me?

Kim Tripp: You still have a job, my friend. You still have a job.

Paul Randal: So, speaking of 100% up time and I mentioned earlier that some things in previous versions suck. One of the things that suck was around peer-to-peer application. Peer-to-peer application was put into SQL Server 2005 and it's a way of being able to scale out your workload across various nodes. Once you've defined your peer-to-peer topology and it's all running, this is all very cool, right? Until you want another node. If you want to add another node to your peer-to-peer topology, you have to acquiesce the entire thing. All the databases in the entire topology, no activity at all. So, you're damned. Then you can add another node, then you can bring it all back up again.

Richard Campbell: That doesn't seem acceptable. I mean obviously, anybody who needs this feature in the first place is so busy, one set of servers couldn't hack it.

Kim Tripp: Well, okay, so yes and no. To a certain extent, peer-to-peer is more about scalability than availability, so I could argue that in a lot of implementations they are going for the scale out, not the availability side but I...

Richard Campbell: You're right. She is more politically correct.

Paul Randal: She is.

Kim Tripp: I know, I know, but seriously though I ...

Paul Randal: I'm in recovery from Microsoft.

Richard Campbell: You know how when you recently quit smoking, you're sort of against all smokers all of a sudden, right? He's swung the other way. He's gone mad.

Paul Randal: Yeah. Anyway, so in 2008, they've actually fixed that problem so you can actually add/remove nodes on peer-to-peer topology on the fly.

Richard Campbell: Live.

Paul Randal: Live without having to quiesce anything, which is fantastic.

Kim Tripp: Yes, and they've actually made the management of the topology a lot easier in SQL 2008 as well as in terms of having a kind of topology viewer and even a graphical way of adding nodes essentially into the peer-to-peer topology. So, in terms of manageability, that's the session that we just did, so we happen to be kind of on a high from manageability, but in terms of a lot of the features, they've made a lot of them more easily manageable



like peer-to-peer. We talked actually about some other features like the Performance Collector and the Data Warehouse capability. In 2005, we have the ability with DMVs to persist information that is related to a query statistics, Index Usage statistics, missing indexes and then...

Richard Campbell: Right, and we're talking Dynamic Management Views here.

Kim Tripp: Yeah, the DMVs, yeah, absolutely. The negative part with DMVs is that the data is not persisted for the life of the server, right? I mean each DMV has a different kind of persistence model and when it goes away and in some cases like in the Index Usage Stats, that's a great example, every time your databases effectively closed in any way, shape or form, you lose all your DMV information about usage stats, so the Server bounces, you lose all your Index Usage Stats. If you close, detach, move a database, restore, you lose everything about the statistics, about that database's usage for indexes. So, my point is, a lot of people are coming up with their own algorithms and deciding that they want to persist this data and there's a lot of different ways to do it and different strategies and while doing that on your own is possible in 2008, they've added something called the Performance Data Collector and it by default will gather some things for you and even set up the automated jobs to collect them, let's say, every 15 minutes and persist the data for up to 14 days and then they have some default reports that are possible so that you can actually go into a certain timeframe and find the different blips per se of activity and kind of zoom in and start to find the expensive queries and queries by CPU, queries by duration, queries that are missing indexes, and all sorts of stuff.

Richard Campbell: The biggest thing I'd want to look at with that kind of persisted data is comparing before and after some changes in the configuration. I go and add my new indexes and I've got my old DMVs that I've saved from yesterday before I made those changes, now I let it run for a couple of days and then compare up.

Kim Tripp: So, one of the things that they will do if they have enough history in terms of query plan history is they will actually show you multiple query plans and they will keep for each of the different query plans the query statistics that tell you the execution duration reads and writes. So, depending on how much information they've got and how many times you've collected the information, you will be able to see a particular query over a particular period of time and what the execution plans were and how they differ. So, yeah, that actually is possible.

Richard Campbell: Yeah, I mean the big thing here is co-relate that same query whenever you generate different query plans and say, "Here are four different kinds of plans and how many times they were used and the distinctions of them."

Kim Tripp: Yeah, absolutely.

Richard Campbell: That's pretty exciting actually. It's a very good idea.

Kim Tripp: Yeah, I agree.

Richard Campbell: All right, new feature, 2008. Your turn, Paul.

Paul Randal: Yeah. She concentrates on the tools. I like the down and dirty engine stuff, so for me...

Richard Campbell: You are the old storage engine guy.

Paul Randal: I know.

Kim Tripp: I like the engine stuff too. That's not fair.

Paul Randal: Well, I don't like the tool stuff. I can't say that, can I?

Kim Tripp: Actually, in the last session, just to make you laugh even more, Paul was showing a few things and he brought up Notepad to actually go and read the error log.

Richard Campbell: Oh no.

Paul Randal: It works.

Kim Tripp: So, I have to give him a little bit of grief on this because there is the log viewer now in SQL 2005, which actually allows you to search and filter on things like error, but what's interesting and maybe one of those still useful reasons of going back to the tools like Notepad is that when you filter, sometimes you'll only get the header and not the actual message because each header, error messages are separated to a separate line, so it's kind of, to be honest, I like the log file viewer but I'm hoping we see a few changes there. I actually don't know...

Richard Campbell: It needs to be a little bit smarter. It needs to be able to relay those lines.

Kim Tripp: Yeah, so sometimes you kind of have to go back to the root tools and even grab a tool like Notepad.



Richard Campbell: Don't you hate it when the guy is right?

Kim Tripp: Now, wait a minute here.

Richard Campbell: What?

Kim Tripp: I don't hate it when the guy is right. I have no problem with the guy being right. It's just that I am usually...

Richard Campbell: Right.

Kim Tripp: Not the guy and I am usually right. They made me say that.

Paul Randal: She is usually right, yeah.

Kim Tripp: Oh, that's a good answer.

Paul Randal: She is. I have to say that because I'm married to her.

Richard Campbell: So very married, yes.

Paul Randal: Okay. So, for me, the most exciting thing I think is compression that's been put into SQL 2008.

Richard Campbell: Really?

Paul Randal: There are three kinds of compression that's been put in. The first kind is base compression. So, one of the major road blocks to people being able to migrate from, say, Oracle or IBM to SQL Server is the fact that on Oracle and IBM, they have data compression. So, consider something like a 1 terabyte compressed database in one of these other systems, you move it over to SQL Server and it might expand out to 5 or 6 terabytes. Now, the cost of having to manage that storage is kind of prohibitive.

Greg Hughes: It is huge.

Paul Randal: Yeah, so that's been a major road block. So, now, there's row and page compression in SQL Server 2008 and the actual compression rate is comparable to the use of other system as well. What's cool about it is you can define the different kinds of compression and features on different partitions so it's not just at a per table level. You can actually go down to the partition granularity. There's a tool that you can use to estimate compression ratios before you actually turn it on because the way you turn it on is you set compression on and you have to rebuild your indexes which is a kind of expensive operation. So, you want to know that you're going to get all the benefits on it before you actually turn it on. Even the DMV sys.DMV Index physical status has been upgraded to

keep track of the ongoing compression ratio of indexes once it is turned on. So, that's the first type. The second type is backup compression. At last, backup compression is native in 2008. So, backup compression, a lot of people has used things like Lightspeed, but if you use Lightspeed, you got to pay for another license and some companies, they don't allow anything but Microsoft software in their IT shops which means they can't use Lightspeed. So, by having backup compression inside SQL Server 2008, you've got it native in the product. You don't have to buy another license and it's actually on by default. That is discovered today. It was supposed to be off by default and they changed its behavior to be on by default now. So, it's pretty transparent, okay. The backup restore command is supposed to know exactly what to do. It means that if you compress your backups now, then obviously it takes less storage space. They're easier to move around because they're smaller. It's faster to do the backup and it's also faster to do the restore, which I love because in a disaster recovery situation, the restore time is probably the major component in getting yourself back up running. So, the faster the restore time, the less downtime you're going to have to take in a disaster.

Greg Hughes: As an IT guy, one of the pains in the past of going from one version of SQL Server to another has to do with backwards compatibility and performance of the new version in a backwards compatible mode. What does it look like going from a SQL 2000 to 2008 and 2005 to 2008 from a performance standpoint now?

Kim Tripp: Well, the first thing that I'll say is they usually don't do all their intense performance testing until all the code is baked for the most part. So, I haven't seen any major performance differences or deltas between the different releases. Hopefully, that's about now when that will start being done because even today they're releasing the next CTP of SQL 2008 so...

Richard Campbell: Yeah, it's imminent, isn't it?

Kim Tripp: Yeah.

Richard Campbell: The next couple of days.

Kim Tripp: Yeah. In fact, they announced yesterday that it was going to be today.

Richard Campbell: Oh okay.

Kim Tripp: So, at least just some internal groups, but I don't know if that will exactly play out to be today, but end of November basically, we're looking at the next CTP.



Greg Hughes: So, by the time this show is aired, we should see the next Community Tech Preview of SQL 2008.

Kim Tripp: Yeah, hopefully. That should be true, yeah. One thing that I would say is it's really nice that you can take a database directly, backup restore it from SQL 2000 to 2008, 2005 to 2008. I've already been doing that now, but it is true and we saw this a lot in SQL 2000 to 2005 migrations that sometimes query plans weren't the same. One of the biggest factors was like updating statistics on the update and you still are going to want to do things like that when you upgrade. I haven't seen anything directly and I haven't even personally done any -- to be honest, I don't do a lot of performance testing until more code is baked because then you will see how the features react.

Greg Hughes: Yeah, that makes sense.

Paul Randal: So, actually it's interesting because one of the things where when you do an upgrade, you are going to see a transparent increase in performance is the third kind of compression which is log stream compression and base to base mirroring. If you have a workload that generates so much transaction log that the network bandwidth you have available, it means that you can't officially send that network, the log over the network to the mirror. When you upgrade to 2008, the log stream will actually be transparently compressed on the principal before it is sent out to the mirror, so reducing the amount of network bandwidth you need and consequently increasing your transaction through on your workload performance.

Richard Campbell: You're just trading processor work for network work.

Paul Randal: Absolutely, yeah. I mean all the things that I've said, there's always a tradeoff between the amount of CPU use to do the compression and the extra performance or compression that you get for instance.

Richard Campbell: So, jumping back to the data compressor because of course I'm always fascinated by that. Is it only via partitioning that I would be able to specify a particular column to compress or can I actually...?

Paul Randal: No, absolutely not. No, you can specify table level, index level and partition level compression.

Richard Campbell: So, pretty much any angle I can go in and decide what I want to compress.

Paul Randal: Absolutely.

Kim Tripp: There are two forms of compression as well. There's basically what's called low level compression and page level compression. Each one has different pros and cons. There have actually been some great resources that are already written in SQL 2008. Paul and I have been blogging about it. This is exactly what we're targeting doing over the next few weeks and months is starting to really get into the cost of the CPU that factors in, but the good news is, is when you're processing less pages and you are using less cache and you're able to have smaller indexes even though you're paying a penalty in terms of CPU, most people aren't CPU bound, they're memory bound, they're disk bound, and at that point you're getting some significant gains even though you are paying for the CPU cost.

Paul Randal: I wonder if the query processors are actually going to be smart enough to realize that an index, even though it is smaller, is actually compressed so there can be extra CPU involved in decompressing the compressed pages before they can actually be...

Richard Campbell: Right, the query plans got to be modified to assess the cost of the decompression.

Kim Tripp: So, there's probably something in the optimizer, I haven't looked into this at all, that would basically look at an index, assess the usefulness of the index and add in to the cost of utilization the cost of compression.

Richard Campbell: I think it would be fascinating to get into a point in a query where you could see how you use this index if it's not compressed and not use it if it's not compressed.

Kim Tripp: Yeah, absolutely. I know Soneil who did some discussions, talked about the relative cost of a particular step if that step was on compressed data, and again because it's smaller that can sometimes be significant when we reduced.

Richard Campbell: We already know this happens sometimes, it's just there are cases where the query processor realizes I'm not going to be banished going to this index, I'm just going to go grab the table.

Paul Randal: Don't get her started on indexes.

Richard Campbell: Oh, don't say the I word.

Kim Tripp: I'll behave.

Richard Campbell: Okay.



Kim Tripp: You know; there are new features in indexes in SQL 2008.

Richard Campbell: Why don't we go there?

Kim Tripp: Actually, this isn't something we talked about today, but SQL 2008 is going to have some new features around the indexes something called the Filter index and what that's going to give you the ability to do is effectively create an index on a certain set of data rather than the entire table. Like today, every index has one entry for every row of the table in the leaf level of the index.

Richard Campbell: Wouldn't I do this with indexing on a view?

Kim Tripp: So, today you can create an index over a view and a view could be on a subset of data, absolutely, but you have to be using the Enterprise edition. Oh, I'm not sure if Filter index is going to be on the Enterprise only. Do you know?

Paul Randal: I don't think those decisions have been made yet.

Kim Tripp: Yeah. This is a good question, right, but Index views historically have been only in the Enterprise, at least fully functional on the Enterprise Edition. You could leverage them if you only access the view and use hints in earlier editions and not smaller editions.

Richard Campbell: Yeah, you had to use the view. I mean it's the advantage of doing a filtered index on the table is now you can continue to query the table and if it finds a case where it make sense to use of filtered index, it will use it.

Kim Tripp: Yeah, and the other thing that's really cool is you can end up saying I want to filter this particular sets of data or create an index over this particular set of data or where the data is not null because you might only need to do your point queries on the not null values, and if the majority of data is null, then having an index that has 80% of its data null...

Richard Campbell: Is useless.

Kim Tripp: Is wasted space, yeah.

Richard Campbell: Well, this also seems like another opportunity to delay archiving, that I could leave my sales data in place for a year, but put additional indexes on the current month.

Kim Tripp: Yeah, or you know even add some indexes over some of the more active data over

the historical data and lessen indexes over the OLTP portion and then...

Richard Campbell: Right, the transactionally speed sensitive data.

Kim Tripp: Exactly, exactly. That's also the beauty with partitioning or not partitioning. With compression that Paul mentioned and partitioning, because you could take the active stuff and if you're finding too much of CPU head, you can actually make the active partition not compressed and the really old data compressed to save space.

Greg Hughes: You know what I was thinking, Richard? They should do their own show.

Richard Campbell: Ah, no, none of that. They are awful good with a microphone though, aren't they?

Kim Tripp: Okay. So, we've been alluding to that for a little while so we are looking at that option and possibilities.

Greg Hughes: I think it would be great. Certainly, there's a need for really smart people providing great information about SQL Server.

Paul Randal: Do you have anybody in mind that could do that?

Kim Tripp: Yeah, who would do that?

Richard Campbell: All right, you two. Have we missed any major features on the management side?

Paul Randal: Transparent Data Encryption. This is very cool.

Richard Campbell: Yes.

Paul Randal: So, in SQL 2005, you could put on Column Level Encryption. You have to encrypt each column individually. Once the column is encrypted, you can't define an Index Server for instance and you can't encrypt all data types. With SQL 2008, you can say "I want this database to be completely encrypted," and it's encrypted on disk, totally transparent. The application doesn't have to change it at all. So, when the pages are in memory, they are decrypted; when they are written out from the buffer pool, they are encrypted. What that means is if somebody, say, detaches a database or leaves a laptop lying around with the database on it, the database is encrypted. So, no more loss of like Social Security numbers by their thousands.

Greg Hughes: Right. I can see from like a PCI compliance standpoint that that can really be valuable. Also, allow application writes to use



encryption in the database or maybe they had to skip doing encryption before in order to get the proper performance out of the database server that they need for their high availability application.

Paul Randal: Absolutely.

Greg Hughes: That's terrific.

Kim Tripp: It significantly reduces complexity application because Column Level Encryption requires a lot of additional application intelligence to really do it well.

Richard Campbell: This is getting rid of the coding side of using encryption in the database.

Kim Tripp: Yeah. I mean it's protecting the database and even in a backup. I mean a lot of people have asked me, "Well, they have backup compression. Do they have backup encryption?" and the answer is yes and no. I mean they don't have it directly where you can say back up, compress it, and encrypt it, but if your database is encrypted using transparent dating encryption, then when you back it up, it will also be encrypted so that will give you double protection.

Greg Hughes: I can't count in my mind the number of architecture discussions I've had with software teams that were building really high availability and highly scalable web application. On SQL Server, they do a great job at the encryption because you have transparent encryption, this is great, is really has been the limiting factor in terms of being able to do the proper job of securing data on disk.

Paul Randal: Yeah, absolutely. One other thing about transparent data encryption is you can choose to have the back up of that database maintain the encryption. So, you can actually take an encrypted backup of the database. You can't have a database that's not encrypted and have a backup that is encrypted, so backup encryption isn't there.

Kim Tripp: You're not listening to me again?

Paul Randal: No, I wasn't listening to you.

Kim Tripp: It's really sad.

Paul Randal: I was watching Richard.

Kim Tripp: I just said exactly that.

Paul Randal: Richard is prettier than you. I was watching him, sorry. Actually, after 11 sessions...

Kim Tripp: You are so in trouble. Let's just put it that way. When this mic is off...

Richard Campbell: I mean you bring up one salient point there which was the database must be encrypted to have a backup that's encrypted.

Paul Randal: Correct, yeah, and you can choose to not have it encrypted as well.

Kim Tripp: I said that.

Richard Campbell: I was watching Paul, you know. He's so pretty.

Paul Randal: He was watching me.

Kim Tripp: I'm getting scared by these two looking across the table at each other.

Paul Randal: We're all speechless.

Kim Tripp: Where do we go from here?

Richard Campbell: Rooted poor Greg.

Greg Hughes: We're just about out of time, Richard, so is there anything else you want to wrap up with? From the perspective of the guy who installs, manages, and runs SQL Server in the IT department, what else do they need to know?

Kim Tripp: So, as far as I think the biggest feature, the one I started with, really, DMF. I see that as probably one of the most exciting, simplified ways to manage a large number of servers with policy based administration, which is just so much easier to manage.

Richard Campbell: Yeah, it's great to hear the word policy associated with SQL Server in general instead of property.

Kim Tripp: Yeah, exactly.

Richard Campbell: Because it's usually the way we've had to think about it, so it's...

Kim Tripp: Yeah, most of the time we're just looking, right? We're not really enforcing policies and now you can literally check, you can periodically recheck and you can even enforce.

Richard Campbell: I think that's very compelling. Ship date, do we have an official ship date yet? I know we're getting a CTP at the end of month.

Kim Tripp: Well, SQL 2009 should be out any... No, I'm totally kidding.



Richard Campbell: Nice, nice.

Kim Tripp: Totally kidding. The party line has always been second quarter in the calendar year of 2008.

Richard Campbell: Right.

Kim Tripp: You know, to be honest, I don't see too many of my customers going to production on the RTM release of any product and I will really wouldn't hold my breath on releasing, let's say in second quarter on let's say whatever the first month is, what, April 1st, right? I wouldn't expect that.

Richard Campbell: No.

Kim Tripp: Would I expect June 4th or 5th or so? Probably more likely and I'm not trying to -- you know, it just takes a long time to bake a product like this.

Richard Campbell: Yeah. Well, and it's SQL Server, it must be right, you know, more so than many others.

Kim Tripp: Yeah, I mean it is data, right? So, I'm always -- a lot of people get really irate when there's slips and slips and slips. You know, I've written a lot of stuff, white papers that I've had to rewrite when things change and it is irritating, but it's part of life. It's protecting our data.

Richard Campbell: Yeah. In the case of SQL Server, it must be right more than on time.

Kim Tripp: Yeah, exactly. So, I expect it in the second quarter or very close to it and I'm really encouraging a lot of my customers to start actively participating in CTPs and starting to get involved with a lot of the blogs that people are writing. There's a ton of us blogging on 2008 already and white papers that are getting written, so there's a lot of great resources out there to start learning, start becoming aware and start even doing some testing.

Richard Campbell: Awesome. We've been talking to Kim Tripp and Paul Randal. Any last words, Paul?

Paul Randal: I'm very tired.

Richard Campbell: Yeah. What's the session you've got left to go?

Kim Tripp: So, we have an ILL, an instructor-led lab, which is actually a repeat of our database mirroring lab and it's 75 minutes and we basically go through...

Richard Campbell: You're not still pulling the USB keys out of the box?

Kim Tripp: No, not for database mirroring, no, but yeah, I still do that demo occasionally, but this one is we set up a database mirroring partnership in the synchronous mirroring model with a witness. We do some failovers. We do application transparent client redirect. We do a WMI-based alert. We send email based on the failover scenario. So, we kind of go through monitoring, alerts, management setup, and it's quite a fun lab, actually. It's one of my favorites.

Richard Campbell: Cool. All right, Kim Tripp and Paul Randal. Thanks very much for coming and talking to us.

Paul Randal: Thank you.

Kim Tripp: It's been a pleasure once again, Richard.

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