



<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 #074
(Transcription services provided by [PWOP Productions](#))



Paul Randal Talks File Streams in SQL Server 2008!
September 10, 2008



Paul Randal Talks File Streams in SQL Server 2008!
September 10, 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 #74, with guest Paul Randal, recorded Wednesday, August 27, 2008. RunAs Radio is produced each week by PWOP Productions, providing professional media and podcasting services online at pwop.com.

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

Greg Hughes: Hey everybody. Richard, how are you doing?

Richard Campbell: I'm well sir, and you?

Greg Hughes: Doing good, doing good, looking forward to having another great conversation related to something SQL server.

Richard Campbell: Oh yes, without a doubt, there's no other way to go about it. Summer is finally over, we're into the Fall.

Greg Hughes: Yeah, it's real nice. It was a good summer. I had an opportunity to do a lot of fun stuff and back into conference season again, aren't we?

Richard Campbell: Oh, without a doubt, yeah. October, November, December, it's going to be just madness for us but in a good way.

Greg Hughes: In November, we'll be in Barcelona once again.

Richard Campbell: Yes, Barcelona, TechEd, Speaker Idol, the whole nine yards. Hey, do you want to be a part of Speaker Idol and you're going to be at TechEd in Barcelona? Send us an email, info@runasradio.com. I got inside connections, I can make it happen.

Paul Randal: Can we judge it again?

Richard Campbell: Would you like to be a judge, Paul?

Paul Randal: Yeah, I've got the time.

Richard Campbell: I'd love for you to be a judge, we're working on the schedule. It's always fun to have you judging too. You guys are great, you and Kim both are great speakers and I think you give good feedback to these...

Greg Hughes: They give excellent feedback, yeah.

Richard Campbell: Speaking of Paul Randal, guess who our guest is this week.

Paul Randal: Hello.

Greg Hughes: Paul Randal?

Paul Randal: It's me.

Richard Campbell: It's you again.

Paul Randal: It's me again. Wow, déjà vu, feels like we just did this.

Richard Campbell: It's just like. So you're the Managing Director of SQLskills but you used to have a real job.

Paul Randal: I used to have a real job, yes, before I became management. No, I'm kidding. I just work on my stuff. I was with Microsoft for nine years. If you want the whole story, listen to the show a couple of times ago because, you know, I don't like argument.

Richard Campbell: Yeah, you bet and certainly DBCC, but you even go back further than that. You were a VMS guy at one time too.

Paul Randal: I used to be responsible for the Open VMS file system and the Analyze/disk which was the equivalent of chkdsk for VMS files.

Richard Campbell: Right.

Greg Hughes: Cool.

Paul Randal: Good old days.

Richard Campbell: Well, I bring that up because our topic today is this whole file stream feature in SQL 2008 which I know is all about managing disks well.

Paul Randal: Yes, yes. A lot of it comes out well. Okay, not, well, kind of.

Greg Hughes: Kind of sort of, yeah.

Richard Campbell: Well, let's start at the beginning because in the beginning, I was a big advocate of not storing big unstructured data in databases. I thought it was just a mistake.

Paul Randal: Why?

Richard Campbell: The whole blob thing, because they were slow, they were cranky...



Paul Randal Talks File Streams in SQL Server 2008! September 10, 2008

Greg Hughes: Performance stuff.

Paul Randal: Performance sucks.

Richard Campbell: Yeah, I would rather store a reference to a file in a string in a database and then go see if I can find the file.

Paul Randal: Its funny you should say that because that's basically what file streams does.

Richard Campbell: Really?

Paul Randal: Oh yes.

Richard Campbell: I'm only trying to sound surprise Paul, come on.

Paul Randal: No, no, no, yeah. The whole problem that file stream solves is how do you keep your file streams and your unstructured data synchronized but still have high performance for your unstructured data.

Richard Campbell: Right.

Paul Randal: In other words, how do you have a database of large video clips and still be able to screen the video scripts from your web server efficiently and storing a large blob inside SQL server, it does not perform very well.

Richard Campbell: Right.

Paul Randal: As you decide and we all know. We haven't tried it.

Richard Campbell: I would like to have the transactional integrity, all of those nice things that I get with SQL server around these big blobs too.

Paul Randal: Right, and so until SQL 2008, you can't really do that. There's no way unless you kind of roll your own system to do it yourself and then different transactional integrity is incredibly hard to do if you have a crash because how do you make the structured data on the database synchronized with the unstructured data in the file system or the file...

Greg Hughes: Ah, you're right.

Paul Randal: Or some other remote blob storage system.

Greg Hughes: Recovery could be really difficult that way.

Paul Randal: Yeah, it's almost impossible.

Richard Campbell: And you're telling me 2008 solves this.

Paul Randal: 2008 solves that and more.

Richard Campbell: And more. I love it.

Paul Randal: Told you, and more. I'm not on the pay of the SQL market and team. File stream is actually pretty cool feature and I actually just wrote a 30-page whitepaper on file stream for the SQL team. That should be published real fast, so by November that should be published.

Richard Campbell: Excellent.

Paul Randal: The whitepaper deals with setting up a system for file stream, why you would want to use file stream in the first place and some of the performance you can choose. It doesn't deal with how you go about programming if you use file stream because I'm not a developer anymore.

Richard Campbell: That's a different problem actually, using file stream.

Paul Randal: That's a whole different whitepaper.

Richard Campbell: Okay.

Paul Randal: So one of the first things that you need to think about is what are the sizes of your blobs and by blobs we mean binary large object, something like a photograph or a document or a video streams, something like that. So there was a research paper that lays which Jim Gray and a couple of others did a couple of years ago called To Blob or Not to Blob, storing files in the database or in the file system. The upshot of that research was if the blob is 256k or less, then it's better to store inside the database. If it's one meg or larger, it's better to store on the file system. If it's in between or a little bit on the gray area, depending on what you want to do. So if you're going to store lots of very small things like widget icons, then you shouldn't be looking at file stream. If you're storing lots of things like videos, then you do want to use something like file stream especially if you want to keep the "methodizer" around the videos or blobs in sync with the actual file themselves. So that's the first thing if file stream is actually right for you.

Richard Campbell: Right.

Paul Randal: A lot of times, the SQL server team are very cool and they put all of these new features out. Some new features, some catching up with other RDM methods but they always have cool new features and they don't give very much guidance



on when you should use them, when you shouldn't use them, what different combinations are that work, that don't work, what the cautions are and so on and so on. So that's what this whitepaper shows like this, how to do this. It gives a little bit of guidance around the how do you go about doing things and what you should do in the first place kind of thing.

Richard Campbell: And a megabyte is not that big so as soon as we're dealing with anything that's a reasonably high res image or any kind of video, we're going to blow through a megabyte.

Paul Randal: Oh, absolutely, yeah.

Richard Campbell: So up until now, I've always stored that independently in the database. If I'm going to store it in the file stream, is it really stored in the database still?

Paul Randal: No. So the way that it works is, let's walk through to actually creating some file stream data.

Richard Campbell: Okay.

Paul Randal: The very first thing you do is set up your system for file stream. Let's put that aside and we'll talk about that later on the show. So you create a database that has a file stream file group. I created a space from scratch to house this or you add it on a file group to the database and you specify that this file group is for storing file stream. So nothing actually gets created in the database. What gets created is a directory structure on whatever file path you've set when you create this file group.

Richard Campbell: Interesting.

Paul Randal: That is called a file stream data container.

Richard Campbell: Okay.

Paul Randal: That is basically going to be a hierarchical tree of directories and files that stores all the actual file stream data. Now, the file stream files themselves have generated names and the generated name is actually stored in the row in the table that relates to that bit of data and so if you do a Select back from the table that house a file stream column, it will actually go out to the file stream file, open the file and suck the information back into SQL server.

Richard Campbell: Cool.

Paul Randal: That's the physical way of accessing the data. So the Win32 way of accessing the database would be you have to connect the database first and get a transactional context and

then you can use that transactional context to get a pathname to the file stream file that you want and then you say, okay, I want to use the open file Win32 API and you specify that you're opening this particular file. There's a file system filter driver that's installed when you enable file stream and it catches that open file request and make sure that you got the right security context to be able to do that and then allows you to go and open a file and then you can read information out of the file using the regular Win 32 APIs.

Richard Campbell: I have a sense that once it was inside of the file stream, the only way to get to it is the database but there are ways to get to it directly.

Paul Randal: That's really the whole point of file stream, it's that you don't have to go through the database to get the information back.

Richard Campbell: Right.

Paul Randal: If you think about it, the database is also stored at the file system. If you were to access it through SQL server itself, then all that file stream data would have to be read in the SQL server buffer pool and then stream out through whatever approach call you're using back to your client.

Richard Campbell: Right.

Paul Randal: And so it has to take a trip through memory. By using Win32 APIs, you're just directly accessing the file itself and it does not have to pass through memory at all.

Richard Campbell: Yeah. It's logical but that means there is no penalty here. If you know how to do this properly, it's going to work just as if it's a regular file system file except that it also has the advantage of being in SQL server.

Paul Randal: It has the advantage that you can tie it into the structured data as well and in a transactional consistent way.

Richard Campbell: Right.

Paul Randal: There are some caches, there are a bunch of different caches. You have to be careful, okay. One of the most interesting ones is, I guess, there is no such thing as a partial update to this file stream later. When you create, if you have a very large file stream file and you want to alter a 10-gig file stream file, you want to alter a couple of files in the file, it creates an entire new 10-gig file with your two bytes. It doesn't just go and alter the existing one.

Richard Campbell: Then throws the old one away.



Paul Randal: It eventually throws the old one away. It has to keep the old one around until the transaction obviously has completed because a roll back might need the old file back and you also has to keep it around so that you can back it up as part of the backing up transaction...

Richard Campbell: Right.

Paul Randal: Because the cool thing about file stream is in the file system, it has its own log as well, there's a file stream log which lives in the direct structure inside the file stream data container. So when you do a back up, it actually backs up all the relational data with the log plus all of the file stream data when its log so you can actually do point in time recovery.

Richard Campbell: That's very cool.

Paul Randal: Yeah. You can do Restore with "stop at."

Richard Campbell: But I also get to sense that what it's telling me is you're going to use way more disk space than you thought possible.

Paul Randal: Well, no, not necessarily because if you're storing a 10-gig image file, say, or a 10-gig video file, it's more like it, a 10-gig video file then in the database, it's going to store 10-gig plus the overhead of having to chunk it off into database pages and so on and so on. In the file system, you're going to store a 10-gig file.

Richard Campbell: Yes.

Paul Randal: So you could actually make the argument you're going to use less space because you don't have the overhead of having to chunk it up and store it in the database.

Richard Campbell: But the price here is that, if I alter that file a couple of times, I would end up with several copies of it until the transaction log gets back up.

Paul Randal: Absolutely right. So one of the kinds of early best practices, this feature has only been there, SQL server was only release a month ago but has been -- Kimberly is actually sitting here with me in the office but I'm not letting her speak though because she's, I don't know, it's kind of disgusting snowball, she's not feeling well...

Richard Campbell: She's not feeling well, that's the nice way you talk about your wife. She's not feeling well.

Paul Randal: And she's a disgusting snowball. But anyway, I was saying, what was I saying? Oh yeah, I remember the thing. So one of the early best practices is if you are going to do a bunch of partial updates to a file, gather them up into one big update.

Richard Campbell: Right.

Greg Hughes: Cue them up.

Paul Randal: Into one big update, okay. So you could have your own, say, bus driver or binary max column that you bunch up only different updates and then when you reach certain threshold, you kind of flush them out into your file stream later...

Richard Campbell: Right, while you change the ones and then you'll only get one additional copy of your big file.

Paul Randal: Right. So you might then say, okay, so you got this other copies of the file, when do they go away? So there's a thing called garbage collection, kind of the same way that garbage collection in managed code works.

Richard Campbell: Right.

Paul Randal: Every so often, the garbage collector will get rid of all the files that are no longer needed and garbage collector for file stream works as part of the database checkpoint per se. So once checkpoint has run and the files are no longer needed for a log back up, they can be safely covered though.

Richard Campbell: Right. So the usual time that files are going to disappear sometime after that log back up wherever that maybe.

Paul Randal: Right, right.

Richard Campbell: That might be awhile.

Paul Randal: There's actually another thing you can do. You can actually have remote access to this stage as well. You can specify a file share, a window share and you can access it using the same mechanism as you would for Win32 access with a remote path...

Richard Campbell: It's a sharable space. All right, so the thing you skip over here was prepare your machine for files.

Paul Randal: Okay. So the very first thing you're going to do is decide whether it's the right thing to store the data or not. So we got pass that, we're going to use file stream. So then you got to set up the hardware. So you're looking at what's the RAID level



Paul Randal Talks File Streams in SQL Server 2008! September 10, 2008

that you want to use. So if you're going to have a lot of file stream data that changes an awful lot, as right intensive, you want to use RAID5 even though that's what your sign guy might try and push on you because it uses the least amount of his disk.

Richard Campbell: You use number of spindles for the most amount of space.

Paul Randal: Exactly, right. You want to also use -- an older set up is basically around making sure you got the right performance.

Richard Campbell: Yeah.

Paul Randal: You want to make sure you're using an appropriate technology, iSCSI versus SATA IDE. So SCSI is usually faster than IDE; but it's going to be more expensive.

Richard Campbell: Right.

Paul Randal: They're smaller disk and they're running faster. So by running faster, the rotational speed of the disk is usually faster. You can get like the sync as in RPM SCSI drive.

Richard Campbell: Right.

Greg Hughes: Right.

Paul Randal: You can get 10,000 RPM SCSI drive name but I don't think they're very common, but anyway SCSI is going to have high rotational speed which means it's going to have lower latency in the disk heads and it's going to have lower seek times.

Richard Campbell: And I got to think for something like stream, rotational performance is going to matter because that will increase your transfer rate too.

Paul Randal: Absolutely. The higher rotational speed you've got, the probably the faster the transfer rate is going to be.

Richard Campbell: Any opinions as far as direct detach storage for SCSI and iSCSI store?

Paul Randal: I don't have any opinions on that.

Richard Campbell: Okay.

Paul Randal: I don't think it makes any difference really.

Richard Campbell: Yeah, if the half doesn't want SCSI, I was just curious if it really matters.

Paul Randal: Make sure that your transport is up to whatever speed you want to avail to. You'll pull the base to drive them.

Richard Campbell: Sure.

Paul Randal: And then whichever one of the technology should you choose, if you're using SATA, then make sure it's worth and secure, and if you're using iSCSI, then make sure it's worth, what's called CTQ and what this technology allow either of those drives to is process multiple into loops and IRQs concurrently which you might want to do if you're accessing multiple file streams at once.

Greg Hughes: Right.

Richard Campbell: For the most part, what you're describing here is just a good idea for any database store.

Greg Hughes: Yeah, yeah.

Paul Randal: Yeah, exactly but what people wouldn't necessarily think about is that they're really just setting up another database and it they're going to do all the same kind of things that they would go through when they're provisioning hardware for any other database.

Richard Campbell: I've heard of NCQ, native command queuing, before but I haven't heard of CTQ.

Paul Randal: CTQ, common type queuing.

Richard Campbell: Okay.

Paul Randal: It's basically the same thing as NCQ.

Richard Campbell: So they're directly related, just one for iSCSI and one for SATA.

Paul Randal: Yup, yup. Let me think, what else? So separating out the data containers, so if you have multiple file stream file groups in the database which you can do quite happily, then don't stick them on the same drive. Just like you don't want to stick multiple data files or data files and log files on the same drive because if you try to access them concurrently, then you're going to close this kind of contention. It's going to slow down your transfer, right...

Richard Campbell: Any particular reason I would have more than one file stream space?

Paul Randal: So you might want to separate the space for different tables. If one table has known



critical base, you might want to have a different container on say RAID5...

Richard Campbell: Or lower liability configuration.

Paul Randal: Lower liability and another one on RAID10 which has triple mode RAID10.

Richard Campbell: Does it matter by size that if I would attend to one group, bigger items on one place, smaller items on another?

Paul Randal: Okay, so this is getting into one of the other set ups for NTFS. What do you want the cluster size of the NTFS to be when you format the drives?

Richard Campbell: Enormous.

Paul Randal: You want it to be as big as possible.

Richard Campbell: Yeah.

Paul Randal: Well, it can if you're storing lots of various small files which is not the design point of file stream...

Richard Campbell: No.

Paul Randal: Then you can leave the default 4k cluster size, but if you're storing blobs that are mega larger, then you probably want to have something like 64k.

Richard Campbell: Yeah, I mean if everything is mega bigger, use the biggest cluster size you lay your hands on.

Paul Randal: Absolutely. I mean that lets your volume size be larger as well.

Richard Campbell: Yeah and if you're wasting some disk space, that's the concern, but you're not wasting that much because your file is so big.

Paul Randal: Exactly. The percentage of wasted space for file is minimal. That's also going to help to reduce all system fragmentation as well.

Richard Campbell: Right.

Paul Randal: Now if you've got a workload that's been lots of inserts, deletes and updates, especially updates because they're created harder than your files, then you're going to end up with fragmentation at the NTFS level and so one of the things before you create a data container on a system that's already been used is defrag the drive and every so often when you're using file stream is defrag the

drive just to make sure that's fragmentation and having to bounce around on the disk, the disk has branch out on the disk to read a single file isn't slowing down your streaming.

Richard Campbell: So DBCC is not going to fix that for me?

Paul Randal: Nothing inside SQL server can affect operating system level fragmentation.

Richard Campbell: I bet if you were still there, it will be fixed.

Paul Randal: Not even if I was there. There's no way that you can -- I don't see how the DBCC can go out to the operating system and run a file system defrag.

Richard Campbell: Right.

Paul Randal: Yeah, that's your cooking fragmentation and then the two key things are turn off 8.3 name generation...

Richard Campbell: It's just not needed.

Paul Randal: And turn off last access time tracking. So what I mean by that, so whenever you create a file in NTFS volume, it generates an 8.3 name, 8 bytes, not 3 bytes name so that 16-bit apps can still use the files. The problem is that the algorithms to do that, when it generates a new name, it has to make sure there's no collisions so it goes and scans all the existing names to make sure that it doesn't collide.

Richard Campbell: Right.

Paul Randal: So as soon as you get a large number of files, the performance of inserting and updating drops down an awful lot. So you need to go out to your drive when you use the NTFS utility and you can turn off a point rename generation. In fact, I just did a blog post about high performance sponsoring tips and tricks on my blog yesterday that has all these kinds of stuff that I'm saying here and it's all covered in the whitepaper as well. To turn that off and then turn off last access time tracking as well.

Richard Campbell: Where do we do this?

Paul Randal: I'm sorry.

Richard Campbell: Where do we do this?

Paul Randal: I'm glad to apply prompts and use the FS utility.



Richard Campbell: Oh, okay. So this is all connected directly to the command prompt.

Paul Randal: Yeah.

Richard Campbell: Excellent.

Paul Randal: You can change, I mean there's a bunch of different things you can change. You can also use FS utility to coach the management as well, but FS utility has a couple of behavior options that you can set to change whether these are enabled or not. So the one caveat to that is if you do have an application that's got basis on that drive as well, if you're sharing the drive of the base container with something else, then turning off that a point rename generation affects the entire volume, the entire NTFS volume. So no 16 app is going to be able to touch that volume.

Richard Campbell: Right.

Paul Randal: So that can be a bummer for you.

Richard Campbell: It may or may not be a huge issue, I mean, it depends.

Paul Randal: Yeah.

Richard Campbell: But it's certainly nothing trivial either, but I mean we're talking about that substantial performance scheme to have these things switched?

Paul Randal: Absolutely. I mean it's actually documented. It's Microsoft documentation that says if you're going to have more than say 300,000 files in a directory, turn off a 3.9 generation.

Richard Campbell: Yeah, any way, not irrespective of ...

Paul Randal: Yeah, in any way, not as good as in file streams, not anyway.

Richard Campbell: This are just good practices for configuring systems as well. What's interesting to me is I've never thought about the fragmentation of files for SQL server because it hasn't matter before. There was only a file.

Paul Randal: Well, I wouldn't say it doesn't matter because I think in the last show that we did, I talked about how SQL server data files can actually get fragmented especially if you've got poorly configured *autogrow* turned on and you're letting autogrow grow the files rather than growing them yourself...

Greg Hughes: Right.

Paul Randal: And if you've got like very large tables inside very large fragmented files and you're doing an allocation order scan even a select star or you're doing a very large range scan, then again the disk heads, to read an over chunk of the data file are going to be dancing around on the disk so you're going to be slowing down your scan performance. So actually it does matter, it doesn't matter as much as an index fragmentation inside data files, but it does have an effect on performance if the data files themselves are fragmented.

Richard Campbell: That's huge. All right, so that's our sort of basic configuration, things we want to do. Anything else we need to know?

Paul Randal: I would say you need to be careful of some of the limitations of file stream, okay, limitation in terms of unsupported feature combinations. So here is a massive unsupported feature combination for me. If you have database mirroring in school, you cannot use file stream. They would just not let you do it.

Richard Campbell: Wow.

Paul Randal: Yeah, if you have file stream, if you configure a system for file stream and then you want to add database mirroring afterwards, you can't do it.

Richard Campbell: Not an option.

Paul Randal: They are not compatible.

Richard Campbell: So you have to use the different back-up strategy.

Paul Randal: You have to use some other kind of strategy.

Richard Campbell: Right.

Paul Randal: And a log shift for instance, or application but you cannot use database or clustering for instance.

Richard Campbell: Right.

Paul Randal: You cannot use database mirroring. I personally think that's going to be a big barrier of adoption for many customers, for V1 because filestream isn't V1 but I have been promised that would be fixed for the next version of SQL server.

Richard Campbell: And you can see why, I mean mirroring means that they will have to mirror all of the file stream files as well.



Paul Randal: Oh yeah, I mean it's a lot of extra space if it's presented across so I'm sure they'll come up with some smart way of to compress things although even in 2008, log stream that they send over is compressed. They had a new feature for 2008. The mirroring log stream that gets send between the principle and mirror is actually now compressed by default. It's about the only different feature that's on by default when you upgrade because it's turned on. Well, the team likes to operate on what they've come across the principle of least surprise.

Richard Campbell: Yes.

Paul Randal: It's when you upgrade, nothing changes.

Richard Campbell: Yeah, it should be the same.

Paul Randal: But this is one that you could at least turn on. You can actually turn it off because as in any kind of compression, the actual amount of compression you get, in other words what's your compression ratio depends on what the data that's you compressing.

Richard Campbell: Right.

Paul Randal: So if your log records are doing lots of little tiny updates, then it's probably not going to compress very well. If your log records are inserting large amounts of data and large amounts of, say, blog data then it probably will compress but if you're going to 2008 and you find that your using more CPU because of the compression database mirroring, you can't turn it off as it documented trace like 1462, it turns off log stream compression.

Richard Campbell: I'm always disturbed that you rip off those numbers out just like that, Mr. Randal.

Paul Randal: Yeah, Kimberly knows the -- she knows the KB articles, although I'm starting to learn them as well. 3551, that's concerning trace like 1118 in 2000, I would say she needs to get out more. She's off in a class.

Richard Campbell: So walk us through a transaction with the file stream. So we literally can roll back, I mean all of those effects will work right down on these files as well?

Paul Randal: Absolutely. There's no limitation.

Richard Campbell: That's why it's making copies of the files so it can always roll back.

Paul Randal: Yup.

Richard Campbell: Or roll forward.

Paul Randal: That's pretty cool. Let me tell you some more interactions with other features, okay.

Richard Campbell: Sure.

Paul Randal: Database snapshot, so you can create a database snapshot on a database, the house file stream in it but the file stream data won't be snapped, right.

Richard Campbell: Okay.

Paul Randal: Database snapshot is only at the SQL level or at the file system level which means that you can get in and play around with your database snapshot or you're not going to be able to access to file stream data. What it also means is you're not going to be able to do one of the cool things you can do in snapshot which is revert to snapshot. Database snapshot is like you take this point in time image of the data space and if something goes wrong with your database, you can say, okay, I want to revert back to that point in time and it will essentially restore the database from the snapshot. If you have file stream data, then you're database can't do that anymore.

Richard Campbell: So it won't work for the regular data either.

Paul Randal: It won't work for regular, yeah, you just kind of revert to snapshot if your database has file stream in it.

Richard Campbell: So disable the feature but that's an interesting consequence. Again, you know, the price of using file stream, if you log your database snapshot feature, it's broken.

Paul Randal: Yup. So I mentioned clustering earlier, if you're going to use failover clustering then make sure that wherever you put the file stream data container is on shared storage so that when the file over happens, the other node can still see it.

Richard Campbell: Just the same as irregular database.

Paul Randal: Yup.

Richard Campbell: So you're under the same constraint.

Paul Randal: That's about it. Oh, here's a really cool one. SQL Express has a four-gigabyte data space-like limit...

Richard Campbell: Yes.



Paul Randal Talks File Streams in SQL Server 2008! September 10, 2008

Paul Randal: But that doesn't include file stream.

Richard Campbell: Oh, okay.

Paul Randal: This is going to be good for developers. You can have this little application go on top of SQL Express and ship as much file stream data as you want or store as much file stream data as you want.

Greg Hughes: Interesting.

Richard Campbell: Yes, it is your choice.

Paul Randal: Yeah.

Greg Hughes: Very cool.

Paul Randal: Apart from those things, basically it works with everything else. It doesn't play balls with partitioning, doesn't play balls with replication, but they're pretty well documented inside books and line.

Richard Campbell: Yeah I know. I was thinking about replication will be an interesting challenge under the scenario too.

Paul Randal: You can actually replicate file stream. You can choose to replicate the data with or without the file stream attribute because, I guess one thing I have to mention is how do you define a file stream column. A file stream column is basically a bar binary max column that you add the word file stream on to.

Richard Campbell: So there's no data type called file stream.

Paul Randal: It's kind of funky. There's no column type called file stream. It's a bar binary max file stream column.

Richard Campbell: But you add the word file stream on to the name of the column.

Paul Randal: Yeah.

Richard Campbell: That's it.

Paul Randal: Yeah. It's kind of funky.

Richard Campbell: Kind of.

Paul Randal: They actually do a create table with table name and your column name, space bar binary max space file stream. So it depended on the end of the data file.

Richard Campbell: That and SQL server watching for that and treats it differently.

Paul Randal: Yup.

Greg Hughes: Interesting.

Paul Randal: One thing I didn't say about assessing up a file stream is there's a funky process where actually installing file stream on your system and it has changed to the various CTPs in SQL server 2008. What I have in RTM is as follows: There's a split between installing on the box itself and installing in SQL server because using file stream involves finding a share name and installing a file system filter driver, that is not something that somebody who's only a SQL admin should be able to do to it to the box.

Greg Hughes: Sure.

Paul Randal: That is what the box admin has to do. So there's kind of a separation of duties thing between the box admin and the SQL admin. So the box admin at the time of installing SQL server or after the fact using the configuration manager has to go and say allow us this instance to use file stream, in other words stick the filthy driver in there and then inside SQL server, the SQL server admin has to enable file stream as well with an SP configuration. So there's kind of a two-stage process.

Richard Campbell: Okay, yeah because it requires different sets of permissions really.

Paul Randal: Different sets of permissions, yeah.

Richard Campbell: And we talk about the NTFS configuration. I can also see in a large organization at least the DBA going to a server admin and saying, "Oh, while you're at it, I need you to set up these drives this way.

Paul Randal: Absolutely.

Richard Campbell: So that is a whole set of steps independent of what we typically, a database guy, need to do.

Paul Randal: I said I register 30 whitepaper and a full one third of the whitepaper is about setting up your system for file stream.

Richard Campbell: Right.

Paul Randal: So it's all the different things that I said in the show today. It has a bunch more information about -- it's very important that when you



choose to do file stream, that you set your system up properly otherwise you are not going to see the performance that you're going to expect.

Richard Campbell: Not that anyone is really going to know what to expect out of file stream anyway but we are shooting for as fast as if the SQL server wasn't involved at all.

Paul Randal: That's the idea.

Richard Campbell: Right.

Greg Hughes: All the power of SQL server without the overhead, yeah.

Paul Randal: Without the overhead, yeah, but unless you do things like disabling a point rename generation and setting up NTFS properly, you're going to start to have these scalability limits of NTFS that hasn't been configured for a high performance, high scalability.

Greg Hughes: Paul, you mentioned PASS, I think you talked about the PASS communities somewhere that's coming up pretty soon. I want you to go ahead and talk a little bit about that and what PASS is also because people who are interested in this topic certainly could probably used to...

Paul Randal: PASS is a voluntary run organization. It's kind of an umbrella professional association over a bunch of user groups for SQL server. They're a bunch of user groups or affiliates of the PASS and PASS is probably most well-known having this yearly conference called the PASS Community Summit, and there's usually one in Europe and the main one is in U.S. They move around from year-to-year and this year happens to be in Seattle. So they do a bunch of pre-conferences and then a few days of sessions and longer spotlight sessions, so this year Kimberly and I are doing a pre-con on database maintenance for The Involuntary DBA which we talked about in the last show.

Greg Hughes: Right, this was before.

Paul Randal: I'm going to do a, it's called light session on recovering from corruption. So there's a session a little bit TechEd where I create all kinds of nasty corrupt databases and then show what tech DB's going to show you in terms of the errors, how you go about recovering from those reattaching suspect databases. So we're going to do all kinds of nasty things and demo them and talk about them. You know, a whole bunch of people are going to be alongside MVPs, lots of Microsoft people, it's the place to be.

Greg Hughes: Looks like this is really for anybody who touches SQL server whether it's a DBA or if you write applications that use SQL server or all of the above.

Paul Randal: It is for developers and IT Pro. There's also another one in Vegas the week before, the Connections Conference which Kimberly and I co-chair which is again a large SQL server conference but also has pre-located developer conference as well. So that's very cool for developers who want to learn more about SQL server and managing SQL server.

Richard Campbell: I expect you will be there as well.

Paul Randal: We're going to be as well but we're always there and we're going to be doing a -- we're going to be doing two pre-cons, one on, again, The Involuntary DBA, and also another one on using new features in 2005 and 2008 to manage a relational data warehouse.

Richard Campbell: Cool.

Greg Hughes: I like this concept of talking around the topic of being the involuntary DBA. There's an offload of those out there.

Paul Randal: As I said last time, there are so many people out there. The cool thing is there is so much for you to learn and as you may know, I love blogging, I'm a prolific blogger so I love blogging out all kinds of tips and tricks and has been working stuff to help these guys as well as the people that are more advance and know about internals.

Richard Campbell: Right.

Greg Hughes: I guess finally, toss out the blog address, go ahead and say it so we can start typing.

Paul Randal: sqlskills.com/blogs/paul or they search for Paul Randal on live search for Google. My blog is called In Recovery.

Richard Campbell: Yes, you are.

Paul Randal: The joke is I'm in recovery from being in Microsoft.

Greg Hughes: The 12 steps of something involved, I'm sure.

Paul Randal: Twelve steps to something, yes. That's right.

Paul Randal: Kimberly blogs as well, in fact Kimberly has just turned over a new leaf she says and



she has been blogging like a crazy person since the weekend.

Richard Campbell: That's awesome.

Paul Randal: Yup. She doesn't blog as much as I do but she blogs long, intense, involved blog posts that just have as much information. Very cool actually, the amount of free information there is on our blogs and just blogs in general that MVPs and other folks do.

Richard Campbell: For sure. Paul Randal, thanks so much for coming on the show.

Paul Randal: You are so welcome.

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