

(January 30<sup>th</sup>, 2017)

If you know someone who would benefit from being an Insider, feel free to forward this PDF to them so they can sign up [here](#).



## Quick Tips for our Insider friends!

Hey Insiders,

This newsletter is coming to you from Redmond, where we've been continuing our content generation and working hard with clients.

We're looking forward to seeing a bunch of you this year in our classes in Chicago in April/May, our classes in Bellevue in July/August, and our Spring SQLintersection conference in Orlando in May. See [here](#) for the class schedule and [here](#) for SQLintersection details. I'm particularly pleased that we have a **three-day class on using PowerShell** to administer SQL Server, taught by industry-expert, MCM, and MVP, Ben Miller.

Even though we're not teaching any Immersion Events in Europe this year, Kimberly and I will both be **presenting at SQLSaturday #620 in Dublin in June**. We're both doing full-day workshops beforehand too, details available by clicking the links below:

- June 15th: I'm presenting [Performance Troubleshooting Using Waits and Latches](#)
- June 16th: Kimberly's presenting [Queries Gone Wrong: Statistics, Cardinality, Solutions](#)

**And even if you can't join us in person**, we're still taking requests for remote sessions for this year. We have 44 scheduled so far; if you'd like one of us to present for your user group, check out my blog post [here](#).

I haven't finished any books since the last newsletter, so instead I'm going to tell you about the first in a three-part sci-fi series I read back in the summer of 2015, called [Nexus](#) by Ramez Naam. The trilogy is about a mind-altering drug (Nexus) that expands consciousness and allows minds to talk to each other. The protagonists have extended the concept to run a Linux-like OS in their heads, with all kinds of interesting apps. And of course the US govt. is against it so all kinds of clandestine ops result, with lots of mayhem. The final book sees a post-human A.I. almost bring about a world war, the downfall of China's leadership, and the world on the path to acceptance of the Nexus drug. All three books are page turners, and highly recommended!

Note: you can get all the prior Insider newsletters [here](#).

## The Curious Case of...

*This section of the newsletter explains problems we've found on client systems; they might be something you're experiencing too.*

I had a random email last week via my [waits library](#) from someone investigating the `LOGMGR_RESERVE_APPEND` wait being the highest one on their 2016 system. This is an unusual wait type to see and it turned out that they'd had encountered almost 10 million of them!

The wait basically means that there's a thread waiting to write a log record into the log, but there isn't space to do so and the log can't grow, so the thread sleeps with this wait type to give the log a chance to clear.

An example scenario would be a database in the simple recovery model (so automatic checkpoints clear the log), with autogrowth disabled and a large number of VLFs (so log truncation takes a while). If there are a lot of concurrent threads generating log records, this wait type will show up and your workload will slow down, as the threads will wait for a second each time.

They'd changed the autogrowth settings to try to avoid log growth waits and ended up with this problem instead. In this case it would have been better to grow the log manually to accommodate the workload's need, and leave autogrowth alone.

Bottom line: be careful what you change around the log file in production without testing it first, as it's pretty easy to cause workload delays with incorrect settings (i.e. the log can be pretty twitchy unfortunately).

### **Paul's Ponderings**

And continuing on the theme of wait statistics, I've had a lot of emails from people over the last six months who've been concerned about a particular wait type being the top of the list on their instance (using something like my query [here](#)) but without any knowledge of the workload or user experience.

In other words, these people have looked at a metric and then been concerned that it's too high without really understanding how that metric relates to what's happening on the server – i.e. is there actually a performance problem?

As you may have heard me say before, this is what I call knee-jerk performance troubleshooting, and it often leads to a lot of wasted time and effort chasing a problem that doesn't exist, or chasing what looks like a symptom of an actual problem, but turns out to be benign.

As far as waits are concerned, no matter what you do, you will *\*always\** see waits being reported by SQL Server, because that's how cooperative scheduling and asynchronous resource waiting works. The trick is in seeing what kinds of waits are the most common, how long the

average waits are, how the pattern of waits differs from normal, and what other symptoms can be observed.

Probably the most important of these points is in knowing how the pattern of waits differs from normal – i.e. the performance baseline. In my experience, it's uncommon to see a new client, or someone I'm talking to over email or at SQLintersection, actually have a performance baseline.

A baseline is a set of measurements of performance metrics when the overall system is performing normally (i.e. satisfactorily for users). Erin paints a comprehensive picture of what this involves in her Pluralsight course ([SQL Server: Benchmarking and Baselineing](#)), but some examples are:

- The I/O load during regular and peak operations, from the SQL Server and Windows perspectives
- Buffer pool health (e.g. Page Life Expectancy and lazy writer activity)
- Run-time characteristics of important queries
- Wait statistics

Having a baseline means that when performance appears to be suffering, measurements can be taken and compared against the baseline, to identify areas that have changed. This allows you to target further investigations rather than flailing around, where you have no idea where to start or just pick a random symptom and focus on that, wasting precious time.

One other thing I see quite commonly is that a client has no way to accurately measure how long an operation takes. How many of you have heard things like '*the web site seems slower today*' or '*my query seems to be taking longer sometimes*'? Just like having a baseline, it's very important that operations in your environment can be measured accurately so you can determine where a slowdown is occurring.

For instance, if you have a website front-end, connecting to a mid-tier caching server, connecting to a back-end database server and things start to run slowly then how do you ascertain which of the components in the overall data flow is contributing to the slow down? Only by having performance and timing metrics at each stage in the data flow can you do that, and only by comparing against your baseline.

**Call to action:** when using any performance metric for troubleshooting, it's imperative that you understand what that metric actually means, and how it relates to a known performance baseline. Getting a baseline isn't hard, and it does take some time, but it can save you so much time down the line when you're investigating a problem.

### **Video Demo**

In this video, Tim demonstrates how to use the updated DTU Calculator to determine a good starting point for which DTU pricing tier or eDTU Database Pool you will need for your Azure SQL Databases. Tim also shows how easy it is to configure and run the command line version of the DTU Calculator, and explains how the DTU Calculator works.

The video is about 6.5 minutes long and you can get it:

- In WMV format [here](#).
- In MOV format [here](#).

No demo code this time.

Enjoy!

### **SQLskills Offerings**

The classes for both Spring and Summer 2017 are available for registration!

To help your boss understand the importance of focused, technical training, we've also added a few items to help you justify spending your training dollars with us:

- [Letter to your boss explaining why SQLskills training is worthwhile](#)
- [Community blog posts about our classes](#)
- [Immersion Event FAQ](#)

### **Upcoming Immersion Events**

Chicago, IL, April/May 2017

- **IE0:** Immersion Event for Junior/Accidental DBAs
  - April 24-26
- **IEPTO1:** Immersion Event on Performance Tuning and Optimization – Part 1
  - April 24-28
- **IESSIS1:** Immersion Event on Learning SQL Server Integration Services
  - April 24-28
- **IEBI:** Immersion Event on Business Intelligence
  - May 1-5
- **IEPTO2:** Immersion Event on Performance Tuning and Optimization – Part 2
  - May 1-5
- **IEPS:** Immersion Event on Powershell
  - May 8-10 **\*\*NEW\*\* class**
- **IEPDS:** Immersion Event on Practical Data Science
  - May 8-12

- **IEHADR:** Immersion Event on High Availability and Disaster Recovery
  - May 8-12

Bellevue, WA, July/August 2017

- **IEPTO1:** Immersion Event on Performance Tuning and Optimization – Part 1
  - July 31-August 4
- **IEPTO2:** Immersion Event on Performance Tuning and Optimization – Part 2
  - August 7-11

Click [here](#) for the main Immersion Event Calendar page that allows you to drill through to each class for more details and registration links.

### **Summary**

We hope you've enjoyed this issue - we really enjoy putting these together.

If there is anything else you're interested in, we'd love to hear from you - [drop us a line](#).

Thanks,  
Paul and Kimberly

[Paul@SQLskills.com](mailto:Paul@SQLskills.com) and [Kimberly@SQLskills.com](mailto:Kimberly@SQLskills.com)