

(February 2nd, 2015)

If you know someone who you think 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,

It's the 100th newsletter!! I've been producing these newsletters since March 2011 – wow!

To celebrate this milestone, I've put together some cool stuff for you:

- **A Pluralsight giveaway.** The first 100 people to respond to the newsletter email asking for a Pluralsight code will get 30-days of access to all 120+ hours of SQLskills online training for free. No catches, no credit cards. Go!
- **My top-ten books of all time** as well as my regular book review
- **100 SQL Server (and career) hints, tips, and tricks** from the SQLskills team in my Paul's Ponderings section.

And the usual demo video, by me for a change. ☺ This newsletter is *loooooong* this time! Grab a cup of coffee and settle in for a great ride.

We've done eleven remote and in-person user group presentations this year already, to a combined audience of more than 1,100 people in the SQL community, and Glenn's presenting remotely in Israel as this newsletter is hitting your inbox. Go community!

I recently won the "Person You'd Most Like to be Mentored By" community award in the annual Redgate Tribal Awards. Thanks! To celebrate, I'm offering to mentor 3 men and 3 women through March and April. See [this post](#) for how to enter to be considered.

I made it to the fantastic Living Computer Museum in Seattle over the weekend, which has a ton of cool old computers, many from DEC where I worked before Microsoft. Check out some photos I took (with permission) [here](#).

The most recent book I've read is Tracy Kidder's excellent [The Soul of a New Machine](#). The book follows the design and building of the Eagle 'supermini' computer by Data General in the late 1970s (I was really hoping to see one at the Museum but they'd moved it to make way for a Control Data 6500 supercomputer). As well as some cool technical discussions around the design and debugging of the various components of the computer, it talks about all the people involved and their experiences during the process. It reminded me in many ways about the last year of SQL Server 2005 development at Microsoft – late nights, weekends, evil, subtle bugs, and the exclusion of much of the rest of life. Highly recommended!

As well as my regular book review, I thought I'd figure out the top-100 books I've ever read. Then I came to my senses, realizing that would be an almost impossible task, and decided to give you a list of my top-10 books (well, some are individual books) of all time, in no particular order, with Amazon links.

1. [The Road](#) by Cormac McCarthy (my top book in 2009)
2. [Shantaram](#) by Gregory David Roberts (my top book in 2010)
3. [The Lord Of The Rings](#) trilogy (rereading right now)
4. [The Hitch Hiker's Guide to the Galaxy](#) 'trilogy' (there are actually five of them) by Douglas Adams (last read in 2000 – time to reread!)
5. The Aubrey/Maturin series by Patrick O'Brien, such as [Master and Commander](#) (last listened to in 2002-3 driving to and from work at Microsoft – time for a reread! – and of course, the utterly fabulous movie)
6. The Culture series of sci-fi books by Iain M. Banks, especially [Excession](#) and [The Hydrogen Sonata](#) (my top book in 2013)
7. [The Horatio Hornblower series](#) by C. S. Forester (last read in 2001 – time to reread! – also love the A&E television series)
8. [The Harry Potter series](#) by J. K. Rowling (last read in 2007 – must have seen the movies 20 times each – they're a winter favorite in our house!)
9. Travel books by Paul Theroux, especially [The Great Railway Bazaar](#) and [Riding The Iron Rooster](#) (last read some in 2011)
10. [Snow Crash](#) by Neal Stephenson (last read in 2012)

Special mentions have to go to the Calvin and Hobbes cartoons, anything by Irvine Welsh (e.g. Trainspotting), and anything involving David Attenborough (one of my few lifelong heroes).

If you absolutely forced me to pick my top book ever (if I had to pick one to be stuck on a desert island with, for instance) it would have to be Lord of the Rings, closely followed by the Iain M. Banks sci-fi books.

Thank you all for signing up to get our newsletter – hope you're still with us when we hit newsletter #200 sometime in 2019!

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

Paul's Ponderings

I asked the team to each come up with 15 hints, tips, and tricks to help me on my goal towards providing you 100 – one for each newsletter. Hmmm, that means that in 2019 we'll need to give you *200* of them. Maybe this wasn't such a smart idea... ☺

In no particular order, here they are!

Glenn

1. You can use the free [CPU-Z tool](#) available from [cpuid.com](#) to verify the details about the processors in your database server. This is especially important for checking the current clock speed of your processors, to determine whether or not any type of power management is in effect on your database server.
2. All recent Microsoft server operating systems (since Windows Server 2008) use the Windows Balanced power plan by default. [You should make sure to change the power plan](#) to use the High Performance plan, and then use the [CPU-Z tool](#) to check the clock speed of your processors to ensure that BIOS-level power management is not in effect.
3. Microsoft recently published a very useful Knowledge Base article called [Recommended updates and configuration options for SQL Server 2012 and SQL Server 2014 used with high-performance workloads](#) that has a lot of valuable information on how to configure and update your SQL Server 2012 and 2014 instances for better performance and scalability.
4. You can use global trace flag 3226 to keep SQL Server from writing messages to the SQL Server error log whenever a successful backup occurs. This will reduce the size of your error log and will make it easier to find relevant error messages.
5. You can use the free script [in this blog post](#) to create some important SQL Server Agent Alerts to help detect many critical SQL Server and hardware related errors in a more timely fashion.
6. With SQL Server 2012 and newer, you can run tempdb on local storage when you are using traditional, failover clustering (which requires shared storage for user databases). If you do this you will probably want to use fast, flash-based storage if possible, for better performance.
7. Intel has an affordable, [high-performance line of PCIe NVMe flash-storage cards](#) called the DC P3700 family that is suitable for many SQL Server workloads. Some good reviews of these cards are available [here](#) and [here](#).
8. If you do purchase an [NVMe](#) flash-storage card, make sure to use the vendor's native NVMe drivers rather than the generic Microsoft NVMe drivers. The [performance difference](#) with the correct drivers can be pretty dramatic.
9. The TPC-E OLTP benchmark is a useful tool for comparing relative processor performance for OLTP workloads, since the benchmark is ultimately CPU-bound when you have adequate storage performance. A list of the best performing processors on this benchmark [can be found here](#).
10. The 22nm Intel Xeon E5-2600 v3 "Haswell-EP" processor family currently has the best single-threaded processor performance available for two-socket servers. You can find out more details about which processor models from this family are the best choice for SQL Server 2014 from this [blog post](#).

11. For standard maintenance tasks in SQL Server, Ola Hallengren's [scripts](#) provide a free and easy to setup solution. We recommend them to our clients and in all our classes.
12. One good way to help justify buying new server hardware if your SQL Server is running on really old hardware is to run [Geekbench 3](#) on your laptop and on the server, and then dramatically announce that your laptop is as fast or faster than the server.
13. Sometimes there are simple configuration issues that cause storage performance problems, such as having a RAID controller, HBA, or PCI-E flash card installed in a low-bandwidth PCI-E slot, or using a RAID-5 configuration for a write-intensive workload. Look for the simple things first once you know that the I/O subsystem is where the problem is.
14. When you're asking for more storage space, ask for the best possible storage (e.g. flash storage or RAID-10). If you don't specify what kind of storage you need, you'll likely be given RAID-5.
15. If you're on Enterprise Edition, the Resource Governor feature can be really useful for separating workloads and constraining them so they don't interfere with each other. This is particularly helpful in consolidation scenarios, and in SQL Server 2014, Resource Governor can even be used to limit I/O resources.

Erin

1. When reviewing the actual query plan for a stored procedure, you can see which values were used when compiling the plan, and the values used when executing it. Within the graphical plan in SSMS, right-click on the SELECT operator and select Properties. Within the Properties pane on the right, expand Parameter List to see the compiled and runtime values. You can also search the XML version of the plan for ParameterCompiledValue and ParameterRuntimeValue.
2. In SQL Server 2008 and higher, SQL Trace offers the ability to capture information 180 events. In SQL Server 2012, Extended Events offer the ability to capture over 600 different events (including all 180 available in Trace), and in the current release of 2014 (CU5), there are over 870 different events. If you're looking to troubleshoot or better understand any new features such as Availability Groups, In-Memory OLTP, or Columnstore Indexes, look to use Extended Events. Query to see what events exist:

```
SELECT [name], [object_type], [description] FROM [sys].[dm_xe_objects]
```

```
WHERE [object_type] = N'event'
```

```
ORDER BY [name];
```

```
GO
```

For more information on getting started with Extended Events see [here](#).

3. The default isolation level for Management Studio is READ COMMITTED. With this isolation level, queries which read large amounts of data can block other queries. If you're querying the production database for investigative purposes, and are not concerned with reading committed versus uncommitted data, consider setting your isolation level to READ UNCOMMITTED for the connection so you don't accidentally block user and production queries. While this is something I recommend when you're running random queries against production, this is not something I recommend as a regular practice in your code or application.
4. The sys.dm_db_stats_properties DMF was added in SQL Server 2008R2 SP2 and SQL Server 2012 SP1 to provide the same information DBCC SHOW_STATISTICS in a much easier manner, and will information about modifications. This is an easier way to track changes to a statistic to determine when it might need to be updated manually, rather than letting the statistic auto-update. More info [here](#).
5. Having issues with Kerberos or SPN configuration and SQL Server? Download the [Microsoft Kerberos Configuration Manager for SQL Server](#) to help determine the issue. Also, if you're not familiar with the different options for setting and removing SPNs, check out the [TechNet article](#) on Setspn.exe.
6. If you're still using sp_updatestats in your job that manually manages statistics, understand that statistics will be updated even if only one row of data has changed in a statistic. For statistics where less than 5% of the data has changed, consider whether the benefit of the update outweighs the resource cost of updating the statistic. See [here](#) and [here](#) for more details. It might be better to programmatically update statistics based on modifications.
7. DBCC CHECKFILEGROUP is a useful option for breaking up integrity checks for large databases. The DBCC CHECKFILEGROUP command runs DBCC CHECKALLOC for the filegroup, then runs DBCC CHECKTABLE for each table in the filegroup. However, any of the table's nonclustered indexes that do NOT exist in the same filegroup as the table will not be checked. Further, if nonclustered indexes for a table which exists in a different filegroup are in the filegroup being checked, they will not be checked either. The only way to comprehensively check a table and its nonclustered indexes when the table and nonclustered indexes reside in different filegroups is to run DBCC CHECKTABLE for that table, or DBCC CHECKDB. More info [here](#).
8. Linked server queries are often known for not performing well. In some cases this is not the result of a poorly written query, or the connection between the two instances, but due to the lack of statistics information. Prior to SQL Server 2012 SP1, in order to obtain statistics information (run DBCC SHOW_STATISTICS), the executing user had to own the table, or be a member of the sysadmin fixed server role, or be a member of either the db_owner or db_ddladmin fixed database role. Users who did not meet this criteria could not obtain statistics information about the object. In 2012 SP1 the permissions were changed so that as long as the user has SELECT permission, DBCC

SHOW_STATISTICS can be executed. Remember that different security options exist for linked servers, so understand which context the connection is using to determine whether the executing user has appropriate rights. More info [here](#).

9. If you use identity columns, you might have to use DBCC CHECKIDENT to check the current value for the identity column. But did you know that if you don't use the right options with DBCC CHECKIDENT, you can actually reset the identity seed unintentionally? To avoid this, always run DBCC CHECKIDENT with the NORESEED option:

```
DBCC CHECKIDENT ('dbo.table_name', NORESEED);
```

More info on this [here](#).

10. There are numerous tools available for monitoring SQL Server performance, but I still use my old friend PerfMon on a regular basis. It's great for capturing data over an extended time for analysis later on, and I also use it to look at performance data live when there's a performance issue. But the default view only includes a couple of counters, and you can waste valuable minutes adding the counters you want to see...unless you use the /sys option! Running perfmon /sys from a Run command allows you to add relevant counters and then save the configuration as a .PerfmonCfg file. The next time you want to look at PerfMon data live, just open up that file and you're off and running. More info [here](#).
11. The [latest version of RML Utilities](#) (9.04.0013) is the first free tool that will process .xel output files from Extended Events for analysis. While the Extended Events UI in Management Studio offers many enhancements for data analysis of captured data, compared to Profiler, RML Utilities goes a little further with its production of graphical output, and its ability to compare two sets of output. Bonus: RML Utilities also includes OStress which provides a free option for stress-testing your environment.
12. If you're looking to upgrade to SQL Server 2014, make sure you understand (and also test!) the supported upgrade path. For example, you can upgrade from SQL Server 2005 to SQL Server 2014, but you need to be running SP4 of 2005 (or later). For exact details of the supported paths, covering version and edition, see MSDN [here](#).
13. The Merge Extended Events File feature (File > Open > Merge Extended Events Files) can be used to merge event_files from multiple servers to correlate events across time, or to merge Extended Events and Server Audit files together.
14. To open a deadlock graph generated by the xml_deadlock_report Extended Event graphically in Management Studio, save the deadlock XML to a file with an extension of .xdl and then close the file and reopen it inside Management Studio. Management Studio checks the file extension when opening the file and if it is .xdl it attempts to load the XML as a graphical deadlock layout instead of as XML.
15. In SQL Server 2012 and higher, the system_health event session captures the sp_server_diagnostics_component_output event and also writes the events to an

event_file target. This can give you a history of server performance in five minute intervals for a few days, weeks, and even months on some servers.

Tim

1. Enable the remote admin connection. By default the dedicated administrator connection (DAC) is enabled from the host only. By enabling the remote admin connection, you will be able to connect using the dedicated admin connection remotely.

```
EXEC sp_configure 'remote admin connections', 1;
```

```
GO
```

```
RECONFIGURE;
```

```
GO
```

2. For SQL Server 2008 Enterprise and above, consider enabling backup compression for the instance. If you do that, make sure that for any databases where backup compression is not useful, you specify `WITH NOCOMPRESSION` on the backup commands.
3. Have a restore validation process in place to regularly restore your backups. Simply having backup jobs run successfully does not ensure that you have good backups. The only way to truly validate a backup is by restoring it. Having a restore validation process in place will not only help you validate backups, it will give you a good idea how long a restore would take in production.
4. Consider enabling instant file initialization to speed up the creation of tempdb, data file creation, and data file expansions. More info [here](#).
5. Increase the default value for “cost threshold for parallelism” from its default value of 5. Perform an analysis of parallelism use based on cost using the plan cache, and then alter the setting accordingly. A typical starting value is anywhere between 25 and 50, but this may vary for your environment. More info [here](#).
6. By default the SQL Server error log retains seven logs and only rolls over to a new log upon restart of the server. This can lead to having very large error logs which makes it slow to load and difficult to manage. We recommend scheduling a job to exec `sp_cycle_errorlog` every day (e.g. at midnight) to cause the log file to rollover. You should also increase the number of log files to at least 30, the max is 99. You want to ensure you maintain enough log files to help with troubleshooting and identifying trends.
7. Implement backup checksums on all backups. This option checks for page checksum failures as the backup is taken, and also allows better verification of backup integrity. See the ‘Backup Integrity’ section at the end of the [feature article](#) Paul wrote for TechNet Magazine.

8. Backup and restore history needs to be regularly purged from msdb to keep msdb lean. If msdb job history is not removed on a regular basis, it can affect overall backup speed and the performance of queries that reference the maintenance history tables. To purge backup and restore history configure a job to execute `sp_delete_backuphistory`.
9. A major part of recovering a SQL Server instance is being able to recover your users and permissions. Consider scripting out to file the users and permissions using the `sp_help_revlogin` script. Details of the process can be found in [KB 918992](#).
10. In order to restore the master database from one server to another, both installs of SQL must be at the same build version. If the original server is inaccessible, you can determine the build version by restoring the header only from the most recent backup of the master database. Sample syntax = `RESTORE HEADERONLY FROM DISK = 'DISK:\PATH\BACKUP_FILE.BAK'`.
11. If your backup testing strategy is to restore the databases on another server and run `DBCC CHECKDB` on them, be aware that when you restore the master database (as a user database), `DBCC CHECKDB` will report some errors, as there are certain structures that are only allowed in database ID 1.
12. Talking about the master database, make sure you practice recovering when the master database is damaged or destroyed. Paul has a good blog post on this [here](#).
13. Which database is the most used on my instance? You can easily find this out by using `sys.dm_io_virtual_file_stats`. This DMV provides how many I/Os have occurred for each database file on the instance. With this information, you can determine which databases for focus your attention on first for optimizations.
14. If you must run antivirus on your SQL Servers ensure that you have proper exclusions for the SQL Server files. This includes the SQL binaries as well as `.mdf`, `.ldf`, `.ndf`, `.bak`. For a full list of recommendations from Microsoft for anti-virus on SQL Servers you can read [KB 309422](#).
15. Furthermore, if you're using FILESTREAM storage, make sure that the FILESTREAM data containers are excluded from anti-virus scanning. If you want to check the data in the FILESTREAM files for possible viruses, you'll need to do that before entering the data into SQL Server.

Jonathan

1. If you encounter a deadlock on SQL Server 2008 or higher, check the `system_health` extended event session for the `xml_deadlock_report`, which is collected by default.
2. If you drop or modify the `system_health` extended event session, it can be recreated from the `u_tables.sql` script file in the instance Install folder (typically `C:\Program Files\Microsoft SQL Server\<Instance>\MSSQL\Install`)
3. Benchmarking a new I/O subsystem with `SQLIO` or `Iometer` can provide nominal throughput numbers for the configuration, but replaying a SQL Server workload using Distributed Replay, will give a better measurement of the configuration's performance.

4. To use more than one replay client with Distributed Replay, the controller must be installed using Enterprise Edition.
5. For stress testing a hardware configuration with Distributed Replay, the configuration file should be set to stress mode, and the ConnectTime and ThinkTime options can be reduced to drive the workload against the target server.
6. Use a custom color in Management Studio for production instances to reduce the risk of mistakes. Click the Options button on a new connection, then the Connection Properties Tab and check the box to Use custom color.
7. Instead of typing all the columns in a table, drag and drop the Columns folder from Object Explorer in Management Studio to the query editor window.
8. Use regular expressions to find or replace text patterns in Management Studio. See [here](#) for more info.
9. When troubleshooting performance problems on a server, remember to check for active running traces in sys.traces and active running Extended Event sessions in sys.dm_xe_sessions which might be impacting performance.
10. The ring_buffer target in Extended Events stores binary data that is serialized as XML output through sys.dm_xe_session_targets. The XML output is limited to 4MB through the DMV, so it is possible for the ring_buffer to hold event data that is not able to be queried. More info [here](#).
11. The lock_escalation Extended Event in SQL Server 2012 and higher, can be very useful in troubleshooting the root cause of certain blocking issues. While this event doesn't exist in SQL Server 2008, the lock_acquired event can be used to identify object level locks (resource_type=5) to identify object locks that are non-Intent based.
12. Synchronizing uncontained objects in Availability Groups (logins, SQL Agent Jobs, server roles, etc.) is critical to having a solid implementation. This manual task can be simplified through the use of the SQLskills 2012 Availability Group Addin for Management Studio. See [here](#) for more info.
13. The only place you can find information about the compile memory gateways for the optimizer is the output of DBCC MEMORYSTATUS. The information is not available through any of the DMVs and can be very useful in troubleshooting RESOURCE_SEMAPHORE_QUERY_COMPILE waits in the system.
14. If you see the RESOURCE_SEMAPHORE wait type on actively waiting sessions, query sys.dm_exec_query_memory_grants to look at the active memory grants and all waiters in the queue to determine the root cause of the wait.
15. Execution memory grant information is only available from the actual execution plan. The memory grant information was expanded in SQL Server 2012 to include the serial required memory, serial desired memory, required memory, desired memory, requested memory, grant wait time, granted memory and the max used memory for the queries execution.

<MemoryGrantInfo

SerialRequiredMemory="6144"

SerialDesiredMemory="6760"

RequiredMemory="51464"

DesiredMemory="52104"

RequestedMemory="52104"

GrantWaitTime="0"

GrantedMemory="52104"

MaxUsedMemory="4680" />

This information can be beneficial in diagnosing RESOURCE_SEMAPHORE waits and how execution memory grant memory is being consumed.

Kimberly

1. Always have baseline metrics. With them, when things go wrong, you'll have better information for troubleshooting. Check out Erin's [Accidental DBA post](#) on baselines: and then her entire [category on baselines](#).
2. Statistics are only as good as you design them and update them. Very large tables might need more statistics to accurately describe them. Check out my Accidental DBA series post on [Statistics Maintenance](#) and then my entire [category on statistics](#). Paul and Erin also have some great posts on statistics as well. Check out their statistics categories: [Paul's statistics category](#) and [Erin's statistics category](#).
3. The columns that make up a clustered index are often referred to as the "clustering key." The data types of the columns that define a good clustering key have these properties: fixed-width data types, none of the columns allow nulls, each column is as narrow as possible, the combination of columns is unique (and defined as such), the columns do not change after insert (static), and ideally, the pattern for inserts is ever increasing. A great choice is a composite clustering key made up of a date data type combined with a row identifier, in that order. This works especially well in very large tables that are also partitioned by a date-related pattern. Check out my [category on the clustering key](#).
4. Define and use foreign keys in the database. Make sure that your foreign keys are indexed. As your indexing strategies improve, you might consolidate and drop a narrow index in favor of a wider index that handles more scenarios. However, starting with a good clustering key and using/indexing foreign keys – are critical for better performance and scalability. Check out my [category on foreign keys](#).

5. There are three important things to check to be sure that your indexing strategies are working. Firstly, remove duplicate or redundant nonclustered indexes – see [this blog post](#). Secondly, make sure that your indexes are healthy, with a good maintenance plan (see Glenn’s tip above where he references Ola Hallengren’s scripts). And lastly, create additionally needed indexes only after thorough testing and evaluation of the pros and cons with respect to performance.
6. Stored procedures should be tested thoroughly and you should have good code coverage – at an absolute minimum. However, stored procedure testing cannot stop there if you also want good performance. You need to verify a stored procedure’s “plan stability” with different parameter combinations. By testing many combinations of execution parameters to see if the optimal plans vary across the executions you can determine whether or not the plan needs to change. If so, then different coding practices must be used to prevent parameter sensitivity problems. Check out my post on [Building High Performance Stored Procedures](#) and then check out my Pluralsight courses on [SQL Server: Optimizing Ad Hoc Statement Performance](#) and [SQL Server: Optimizing Stored Procedure Performance](#).
7. Be sure to use TRY... CATCH elements and always use good error handling techniques in stored procedures and any functional code.
8. Ad hoc statements can pollute your plan cache because SQL Server adds every statements plan into the even if it's only ever executed once. To change this default behavior, turn on the server-option: optimize for ad hoc workloads. This will reduce the amount that's initially put in the cache (only a “compiled plan stub” is placed in the cache – not the entire plan). Only if the statement is executed a second time will SQL Server place the plan in cache. Check out my blog post [Plan cache and optimizing for adhoc workloads](#).
9. It's a good idea to periodically check the amount of plan cache allocated to only once-executed statements. Because you can define the amount of memory that SQL Server allocates to the plan cache, you might be allocating a large amount for statements that have only executed once. As a result, periodically clearing the cache is required to allow the cache to be used for something more useful – like data. This is a good follow-on post for this – as well as my prior – tip: [Plan cache, adhoc workloads and clearing the single-use plan cache bloat](#).
10. Have a good understanding of locks and blocking, but remember that it’s usually a symptom of another problem, and not the root cause itself. Things to consider are inadequate indexing, poor coding practices, and improper statement hints.
11. Have a good strategy for creating your databases so that you don’t end up with too many or too few VLFs. Check out my [blog post](#) about this (and be sure to read: [8 Steps to better transaction log throughput](#)) as well as Paul’s post about the [Important change to VLF creation algorithm in SQL Server 2014](#).

12. Don't believe in absolutes. SQL Server is a multipurpose relational database engine and with it – you can do ANYTHING! There's an amazing amount of power and flexibility available to you and it's really rare that there's only one way to do something. Often, there are many ways to handle a problem – each having different pros and cons. Always thoroughly evaluate your options and prototype your solutions. Check out Paul's excellent post titled: [It depends. It really, really does.](#)
13. In my wish for “15 #SQLFamily Wishes for 2015” I quoted Yogi BhaJan: If you want to master something, teach it. I want to remind you that one of the best ways to learn something is to teach it. Create an “education time” for your team. Using 2 hours on Friday mornings (someone brings bagels ☺), two or three of you present something to the team. This will keep you actively learning, actively presenting, and it will get more eyes looking at important topics. Pick blogs posts, tweets, presentations attended at conferences, any interesting topic – and run with it. Read the rest of the SQLFamily wishes [here](#).
14. Read the Accidental DBA Series from start to finish... this is useful to do periodically just to make sure that you're staying fresh on each of those ever-important topics: <https://www.sqlskills.com/help/accidental-dba/>
15. The most important tip I can give – have fun. Do something you love. Life's too short to waste it on something uninteresting to you.

Me

1. Make sure you're not wasting buffer pool memory. It's the most precious resource on your server and if you have rampant index fragmentation causing low average page densities, you'll have a lot of your buffer pool storing empty space, which will mean less data and more physical I/O for your workload. More info in [this post](#).
2. Make sure you don't have tempdb allocation bitmap contention. This is a common problem with many concurrent connections all creating and dropping small temp tables causing access bottlenecks on some of the allocation bitmaps (PFS and SGAM pages). Enabling trace flag 1118 (all systems in the world should have this enabled) and creating multiple tempdb data files can fix this. More info in [this post](#) and [this post](#).
3. Make sure you don't have tempdb or log I/O performance problems. Sometimes basic I/O performance is one of the major things dragging down the throughput of your workload, so it's important that you check up in them periodically. [This post](#) has all the details as well as some very interesting survey results.
4. Use wait statistics to help with performance troubleshooting. SQL Server keeps track of why threads have to wait (the resource required and the length of the wait time) and this is one of the easiest and best ways to begin figuring out what's going wrong with the workload. [This post](#) (the most popular post on our website) is the starting point for you, as well as my [wait statistics category](#), and [Pluralsight course](#).

5. Understand what CXPACKET waits are. They're not always bad. The control thread in a parallel operation always accrues a CXPACKET wait, so unless you don't want parallelism, that's not a big deal. When non-control threads also accrue CXPACKET waits, that's a sign of a skewed distribution of work from poor statistics. Don't blindly lower server maxdop to try to remove them without understanding what they mean. More info in the links above and in [this newsletter](#).
6. Understand what ASYNC_NETWORK_IO waits are. Rarely are they anything to do with the network. They're generally from poorly-written applications doing RBAR (Row-By-Agonizing-Row) processing of data and nothing to do with SQL Server. More info in the link in #4 and in [this newsletter](#).
7. Keep snapshots of I/O latencies and wait stats. One of the easiest ways to make sure you're monitoring what's going on in your server is to keep history of metrics. If you have a third-party monitoring too, that will probably do it for you, but it's simple to do yourself too. [This post](#) gives a script to capture 30 minutes of I/O latencies, and [this post](#) gives a script to capture 30 minutes of wait statistics.
8. Check your disk partition alignment to make sure that your I/O subsystem doesn't have to read double the number of RAID stripes for each read and write SQL Server does. More info [here](#).
9. Don't scrimp on index maintenance when using mirroring or AGs. With these technologies you have no choice but to use the full recovery model, which means the index rebuilds are fully logged. Often this generates so much transaction log that people stop removing fragmentation, to the great detriment of the workload's performance. In these cases, using index reorganize is the way to go, even performing a little each day in a staggered fashion. I describe this method [here](#) on our old SQL Server Magazine blog.
10. Don't take anyone's advice without an explanation. Quite often I'll hear from clients that a prior consultant told them to do X, which they did, and it caused another problem and the consultant didn't explain why X was being recommended. Sometimes it's even a Microsoft support engineer who gives the advice, without any explanation. Don't stand for it! Always make sure that someone can explain the advice they're giving you, so *you* know that *they* know what they're talking about.
11. Don't misuse temp tables. Sometimes temp tables become a design pattern for an organization and they get used everywhere, when often a query will execute faster without using a temp table at all. And when temp tables are used, they're often abused by being populated and indexed incorrectly. In [this newsletter](#) from 2011 I explain further.
12. Make sure you know how to regain access to SQL Server when there's no sysadmin role or everyone's forgotten what the sysadmin password is. There's no need to reinstall SQL Server! Erin demonstrates how in [this newsletter](#) and in [this one](#).
13. Don't knee jerk. This applies to many facets of life and work, but in particular it applies to performance tuning. When you see a symptom (e.g. a particular wait type), don't

immediately assume that it's the root cause of the problem. Jumping to conclusions without understanding and making sure is a waste of time and resources.

14. Make sure you understand and (in most cases) enable configuration options like 'lock page in memory' and 'optimize for ad hoc workloads', which can have a profound effect on the performance of your workload. Jonathan discusses the former [here](#), and Kimberly discusses the latter [here](#).
15. Make sure you understand the potential problems you could run into from using the bulk_logged recovery model. Although it does allow some operations to generate less transaction log, there are two potentially disastrous limitations you may hit if a crash and corruption occur at the wrong time. More info in [this newsletter](#).
16. Understand the potential problems you could run into when adding more memory to your server. Most people think that it's just a simple thing to add more memory and the workload will perform better. But there are a bunch of considerations you need to think about that could cause you problems – like increased shut down and start up times. More details in [this post](#).
17. Know how to perform consistency checks correctly. There's no substitute for regular runs of DBCC CHECKDB to make sure that your database is free of corruptions, and that includes read-only and system databases. If you're not doing it properly, you run the risk of data loss and downtime. [This newsletter](#) has more details.
18. Know how to perform EMERGENCY-mode repair and re-attach a suspect database. The [second](#) and [fourth](#) most popular posts on my blog discuss these in more depth, respectively. As part of any disaster recovery plan, and testing, you should know how to recover from a disaster when you don't have backups available, or after someone else has messed up the DR process.
19. Use multiple filegroups. They have so many good uses: allowing targeted disaster recovery, allowing part of the database to be made read-only, allowing placement of parts of the database on different portions of the I/O subsystem, helping partitioning, and so on. More details in [this newsletter](#).
20. Make sure the company you work for values you and takes care you. Nothing is more demoralizing than knowing that your boss and/or your company doesn't value your hard work, and nothing motivates someone to want to leave a company than when they're not being compensated or treated well. It's a seller's market for good DBAs and developers right now, so if you're not being appreciated and treated well, look around for a better gig. A manager who doesn't look after his/her team members is a very poor manager. More in [this post](#).
21. Don't equate ignorance with stupidity. Read [this post](#).
22. Be polite. Read [this post](#), [this post](#), and especially [this post](#).
23. Get involved with the community. There are so many ways to be involved with the community, and by reading this newsletter, you're already passively involved. Passive involvement is totally fine – not everyone has the time, skills, or inclination to want to

organize a user group, speak in public, or start a blog, and those of us that do, do it for all of you out there who want to read and listen to what we have to say. If you do want to get actively involved, there are tons of ways, including blogging, answering #sqlhelp questions, answering forum questions (like MSDN, Stack Overflow, Ask SQL Server Central), speaking at conferences, SQL Saturdays, and user groups, and ultimately organizing and running a conference, SQL Saturday, or user group.

24. Get a mentor. No matter who you are, how experienced you are, and what your position is in the community and your life, you would likely benefit from having a mentor. There's always someone who has more experience or who has different experience that you can benefit from. And then the ultimate is become a mentor yourself. It's one of the most rewarding things you can do – helping another human being grow in some way. I've been mentoring people for the last 16 years since I joined Microsoft and I still enjoy it. See the start of this newsletter for details of how you can be mentored by me.
25. Read more. Reading is my #1 passion in life apart from Kimberly and the kids, as you probably already know. Reading broadens your horizons and keeps your brain healthy. I don't think people read enough – mostly through lack of time – and I think it's something that everyone should make time for. You can find plenty of recommendations in my [books blog category](#).

Call to action: Nothing specific this time. We hope these hints, tips, and tricks help you out!

Video Demo

SQL Server 2014 introduced a cool new feature called delayed durability, which alters how log flushing works. In the right circumstances it can give your workload a huge performance boost, with a trade off against the durability of the transaction when it commits. In this video I show an example of using delayed durability – pretty spectacular! You can read more about this feature in [this blog post](#) from November 2014.

The video is just over 7 minutes long and you can get it:

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

And you can get the demo code [here](#).

Enjoy!

SQLskills Offerings

We've released most of our classes for 2015. We might add one more domestic delivery of IEPTO1 in the 2nd half of the year and possibly IEPTO2 / IEBI but these will be partially based

on instructor and conference schedules; it's important to note that 2nd half courses will be very limited.

Finally, to help your boss understand the importance of focused, technical training, we've added a few new 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

Sydney, NSW, Australia

- February 23-27, 2015: **IEPTO2**: Immersion Event on Performance Tuning and Optimization – Part 2 (formerly IE2)

Chicago, IL

- April 27-29, 2015: **IE0**: Immersion Event for the Accidental/Junior DBA
- April 27-May 1, 2015: **IEPTO1**: Immersion Event on Performance Tuning and Optimization – Part 1 (formerly IE1)
- May 4-8, 2015: **IEPTO2**: Immersion Event on Performance Tuning and Optimization – Part 2 (formerly IE2)
- May 4-8, 2015: **IEBI**: Immersion Event on Business Intelligence
- May 11-15, 2015: **IEHADR**: Immersion Event on High Availability and Disaster Recovery (formerly IE3)

Bellevue, WA

- June 8-12, 2015: **IEPTO1**: Immersion Event on Performance Tuning and Optimization – Part 1 (formerly IE1)
- June 15-19, 2015: **IEPTO2**: Immersion Event on Performance Tuning and Optimization – Part 2 (formerly IE2)

London, UK

- August 24-28, 2015: **IEPTO1**: Immersion Event on Performance Tuning and Optimization – Part 1 (formerly IE1)

Dublin, Ireland

- October 12-16, 2015: **IEPTO2**: Immersion Event on Performance Tuning and Optimization – Part 2 (formerly IE2)

See [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 and Kimberly@SQLskills.com