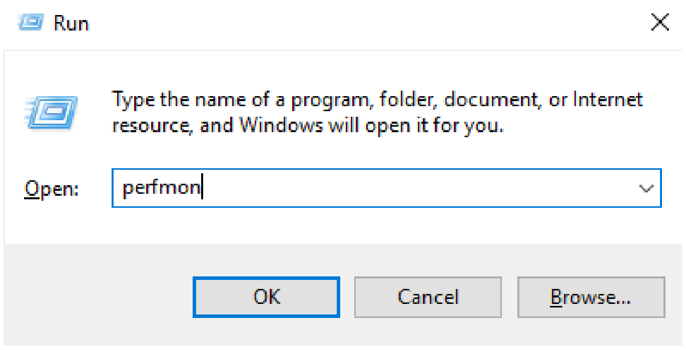
Performance Monitor

Windows Performance Monitor or PerfMon is another great tool to capture and graph many aspects for the Windows server. There are counters for .NET, Disks, Memory, Processors, Network, etc... As well several counters related to each instance of SQL Server on the box.

Start > Control Panel > Administrative Tools > Performance Monitor or you can launch **PerfMon.exe**.

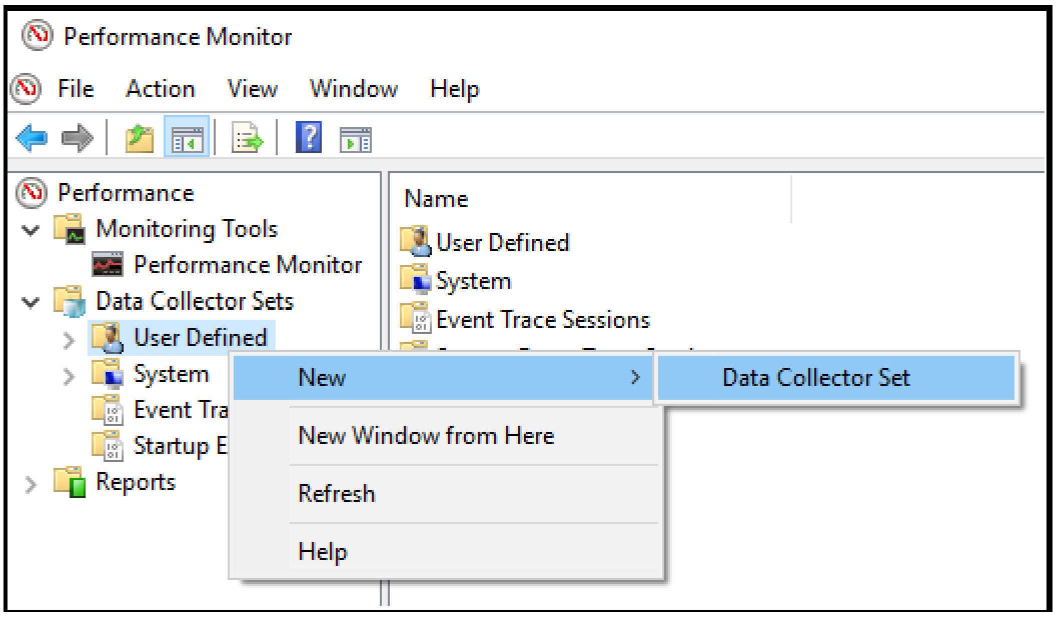
OR

Use the Windows key + R keyboard shortcut to open the Run command, type "**perfmon**", and click **OK** to open:

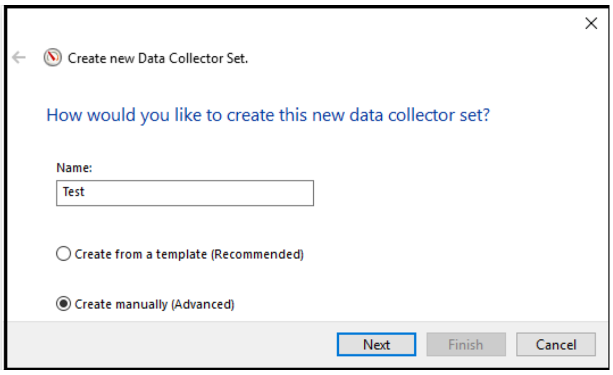

Run
Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.
Open: perfmon
OK
Cancel
Browse...

When you first open this **SQL Server monitoring tool**, it'll open in the main page with a classic overview where we can watch counters. What I like to do is to create a custom Data Collector Set which is how we can create custom sets holding performance counters and alerts based on specific criteria.

Expand **Data Collector Sets**, right-click **User Defined**, select **New**, and then click **Data Collector Set**:



Give this new data collector a name, we don’t want to create it from a template, so change to option to **Create manually (Advanced)** and click **Next**:



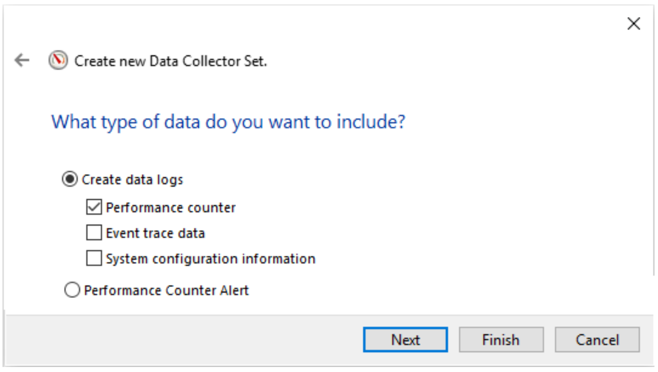
Create new Data Collector Set.

How would you like to create this new data collector set?

Name: Test

Create from a template (Recommended) [✓ Create manually (Advanced)]

As for the data type, we want to track performance counters so check the **Performance counter** option and click **Next**:



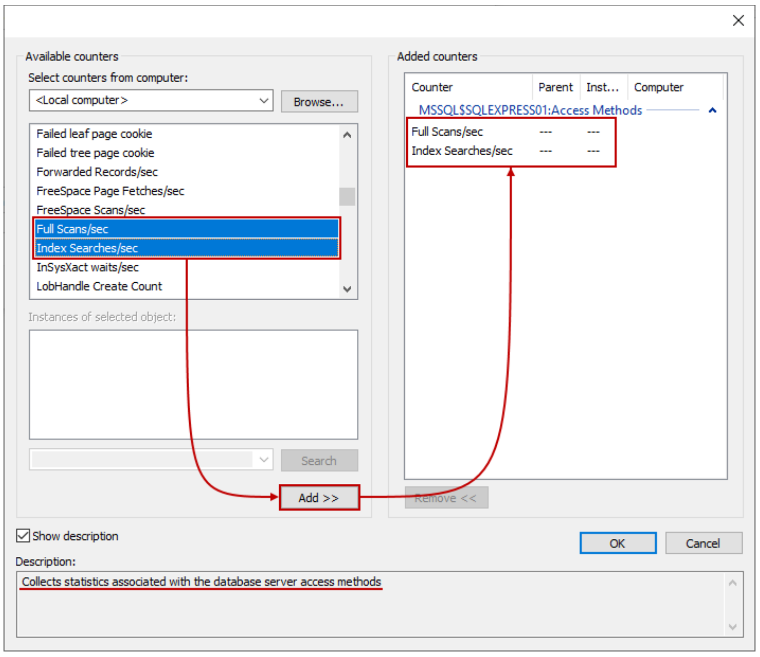
Create new Data Collector Set.

What type of data do you want to include?

[✓ Create data logs]
- [✓ Performance counter]
- Event trace data
- System configuration information

Performance Counter Alert

Next, the wizard will ask you which performance counters you would like to log. Counters can be added by simply clicking the **Add…** option and locating them in the list. For example, we can locate **SQLServer:Access Methods**, expand it to find **Full Scans/sec** and **Index Searches/sec** counters.



Available counters

Select counters from computer: <Local computer>

Failed leaf page cookie
Failed tree page cookie
Forwarded Records/sec
FreeSpace Page Fetches/sec
FreeSpace Scans/sec
Full Scans/sec
Index Searches/sec
InSysXact waits/sec
LobHandle Create Count

Instances of selected object:

Added counters

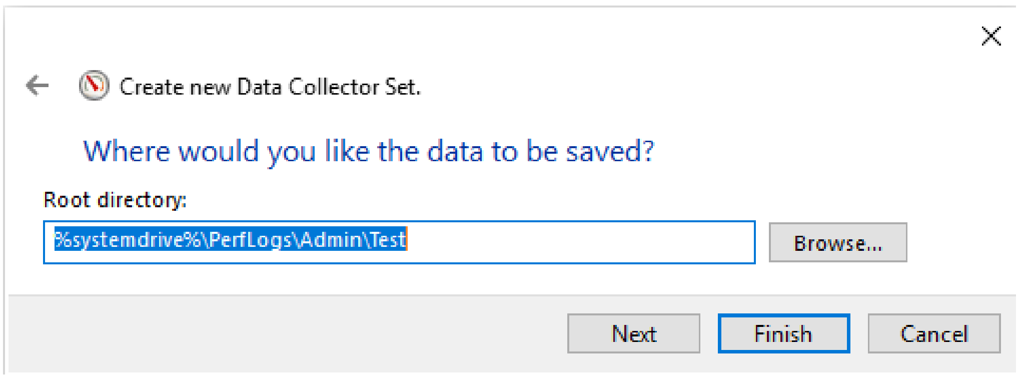
Counter	Parent	Inst...	Computer
MSSQL$SQLEXPRESS01:Access Methods			
Full Scans/sec	---	---	
Index Searches/sec	---	---	

[✓ Show description]

Description:
Collects statistics associated with the database server access methods

MOHSIN KHAN
SQL NOTES

Hit **Next** two times, then **Finish** and the wizard will create the custom data collector set:

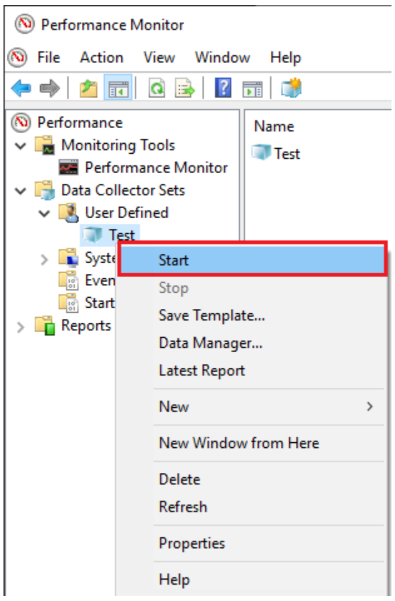

Create new Data Collector Set.

Where would you like the data to be saved?

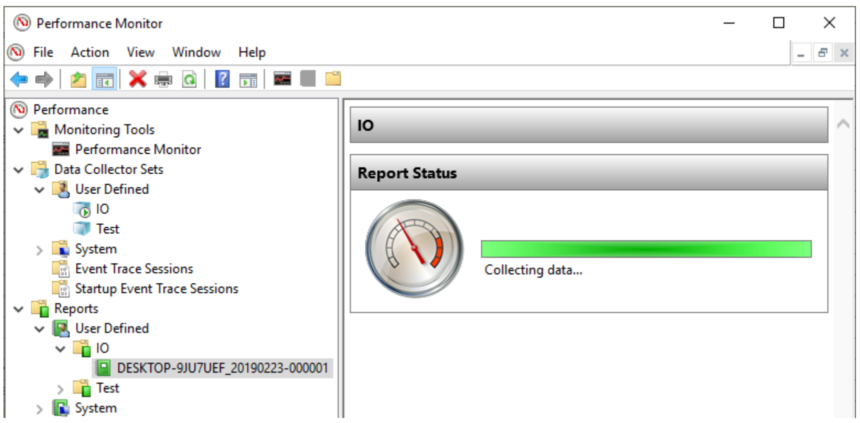
Root directory:

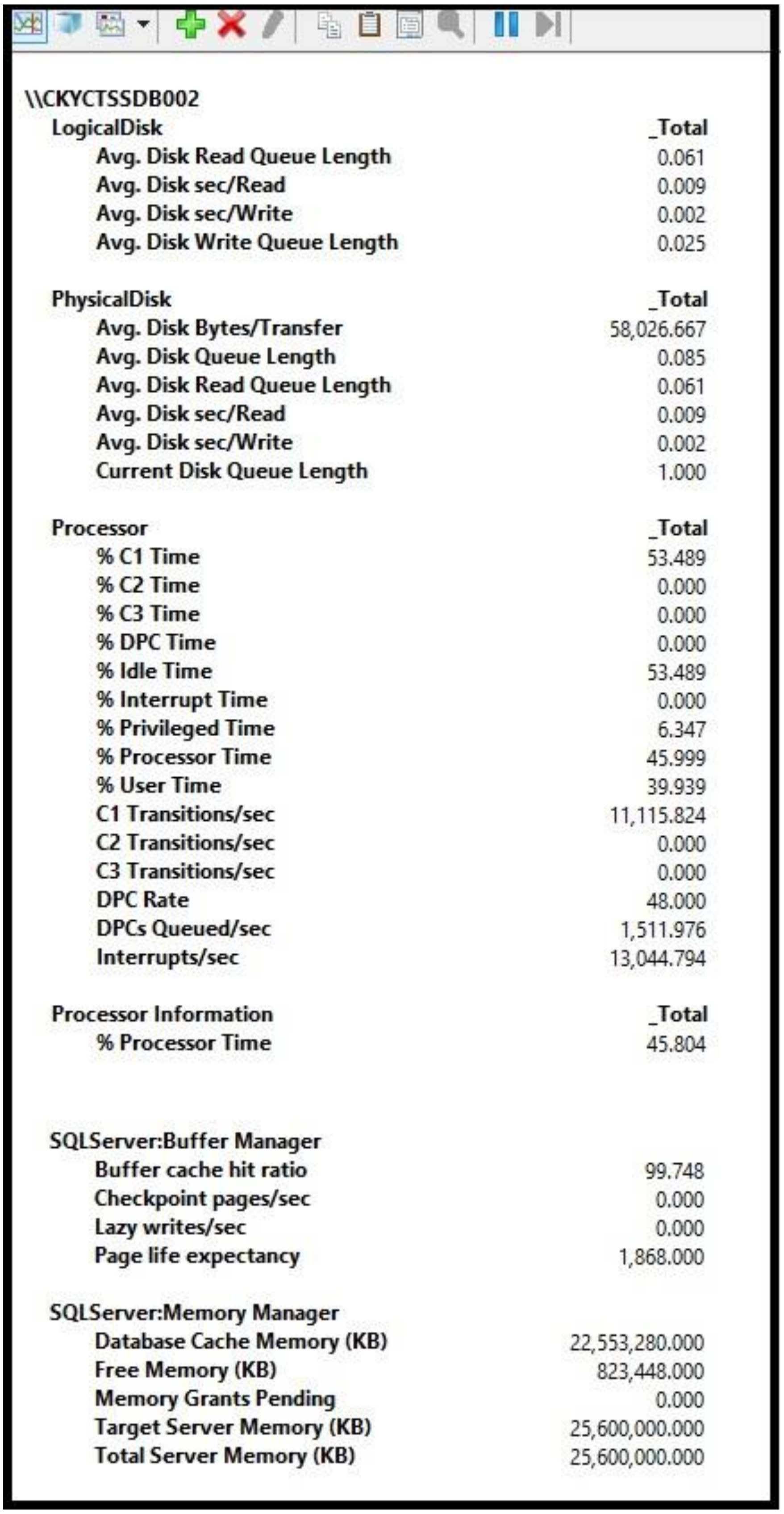
%systemdrive%\PerfLogs\Admin\Test

Once you completed all these steps, you can right-click newly created data collector set under **User Defined** category, and click **Start** to run it or **Stop** to shut it down:



I've already preconfigured one for monitoring disks I/O activity in real-time. I've also started it and let it collect data for a while to collect data. This can be seen under **Reports** folder, **User Defined** category:





```
\\CKYCTSSDB002
   LogicalDisk                                        _Total
        Avg. Disk Read Queue Length                    0.061
        Avg. Disk sec/Read                             0.009
        Avg. Disk sec/Write                            0.002
        Avg. Disk Write Queue Length                   0.025

   PhysicalDisk                                       _Total
        Avg. Disk Bytes/Transfer                  58,026.667
        Avg. Disk Queue Length                         0.085
        Avg. Disk Read Queue Length                    0.061
        Avg. Disk sec/Read                             0.009
        Avg. Disk sec/Write                            0.002
        Current Disk Queue Length                      1.000

   Processor                                          _Total
        % C1 Time                                     53.489
        % C2 Time                                      0.000
        % C3 Time                                      0.000
        % DPC Time                                     0.000
        % Idle Time                                   53.489
        % Interrupt Time                               0.000
        % Privileged Time                              6.347
        % Processor Time                              45.999
        % User Time                                   39.939
        C1 Transitions/sec                        11,115.824
        C2 Transitions/sec                             0.000
        C3 Transitions/sec                             0.000
        DPC Rate                                      48.000
        DPCs Queued/sec                            1,511.976
        Interrupts/sec                            13,044.794

   Processor Information                              _Total
        % Processor Time                              45.804


   SQLServer:Buffer Manager
        Buffer cache hit ratio                        99.748
        Checkpoint pages/sec                           0.000
        Lazy writes/sec                                0.000
        Page life expectancy                       1,868.000

   SQLServer:Memory Manager
        Database Cache Memory (KB)            22,553,280.000
        Free Memory (KB)                         823,448.000
        Memory Grants Pending                          0.000
        Target Server Memory (KB)             25,600,000.000
        Total Server Memory (KB)              25,600,000.000
```

OS Memory & Paging Performance Counters			
Object	**Counter**	**You Want**	**Description**
Memory	Available Mbytes	> 100MB	Unused physical memory (not page file).
Memory	Pages Input/Sec	< 10	Reads from hard disk per second to resolve hard pages.
Memory	Pages/Sec	See Description	Often referenced in older documentation. Useful only in combination with Pages Input/Sec, %Usage, %Usage Peak.
Paging File	%Usage	< 70%	Amount of Page File in use, which indicates the server is substituting disk space for memory.
Paging File	%Usage Peak	< 70%	Highest %Usage metric since the last time the server was restarted.

OS Disk & Miscellaneous Counters			
Object	**Counter**	**You Want**	**Description**
Physical Disk	Avg. Disk Sec/Read	< 8ms	A key measure of disk latency represent- ing the average time, in milliseconds, of each read to disk where > 20 is poor, <20 is good/fair, <12 is better, <8 is best
Physical Disk	Avg. Disk sec/Write	< 8ms (non cached) < 1ms (cached)	A key measure of disk latency representing the average time, in milliseconds, of each write to disk, where non-cached writes (> 20 poor, <20 fair, <12 better, <8 best) differ significantly from cached writes (> 4 poor, <4 fair, <2 better, <1 best). For OLTP databases, the lower this number the better, especially for disks holding the transaction log.
Network Interface	Bytes Total/sec	See Descr iptio n	The number of bytes sent and received over a specific network adapter, includ- ing framing characters. Be sure to record the throughput of your SQL Server's NIC card(s). Watch for this value pos- sibly exceeding the NIC's specifications, especially when conducting large and/ or multiple backups or copies to network drives. A high-speed network and/or a NIC dedicated to admin processes often allevi- ates this bottleneck. This counter is a sum of "Network Interface\\Bytes Received/ sec" and "Network Interface\\Bytes Sent/ sec". In some situations, you may wish to determine both inbound and outbound network traffic separately.This counter is particularly useful in iSCSI environments where it can help to measure disk I/O when the NIC is dedicated to storage.

MSSQL Buffer Manager & Memory Performance Counters			
Object	**Counter**	**You Want**	**Description**
SQL Server: Buffer Manager	Free List Stalls/sec	< 2	Monitors the number of requests per second where data requests stall because no buffers are available. Any value above 2 means SQL Server needs more memory.
SQL Server: Buffer Manager	Lazy Writes/Sec	< 20	Monitors the number of times per second that the Lazy Writer process moves dirty pages from the buffer to disk as it frees up buffer space. Lower is better with zero being ideal. When greater than 20, this counter indicates a need for more memory.
SQL Server: Buffer Manager	Checkpoint Pages/ sec	See Description	Monitors the number of dirty pages, per second, that are flushed to disk when SQL Server invokes the checkpoint process. Checkpoint frequency is influenced by the recovery interval setting in sp_configure. High values for this counter may indicate insufficient memory or that the recovery interval is too high.
SQL Server: Buffer Manager	Page Life Expecta ncy	> 300	Tells, on average, how many seconds SQL Server expects a data page to stay in cache. The target on an OLTP system should be at least 300 (5 min). When under 300, this may indicate poor index design (leading to increased disk I/O and less effective use of memory) or, simply, a potential shortage of memory.
SQL Server: Buffer Manager	Page Lookups/sec	(Page lookups/ sec) / (Batch Requests/ sec) < 100	The number of requests to find a page in the buffer pool. When the ratio of batch requests to page lookups crests 100, you may have inefficient execution plans or too many adhoc queries.
SQL Server: Buffer Manager	Page Reads/sec	< 90	Number of physical database page reads issued per second. Normal OLTP workloads support 80 – 90 per second, but higher values may be a yellow flag for poor indexing or insufficient memory.
SQL Server: Buffer Manager	Page Writes/sec	< 90	Number of database pages physically written to disk per second. Normal OLTP workloads support 80 – 90 per second. Values over 90 should be crossed checked with “lazy writer/sec” and “checkpoint” counters. If the other counters are also high, then it may indicate insufficient memory.
SQL Server: Buffer Manager	Read ahead/sec	< 20% of Page Reads/ sec	Number of data pages read per second in anticipation of their use. If this value is makes up even a sizeable minority of total Page Reads/sec (say, greater than 20% of total page reads), you may have too many physical reads occurring.
Memory	Free System Page Table Entries	>10,000 > 24,000 on boot	Shows the number of page table entries (PTE) not in use on the server. PTEs are used to map virtual to physical memory addresses and are affected by the /PAE and /3GB Windows boot switches.

MSSQL Workload Performance Counters			
Object	**Counter**	**You Want**	**Description**
SQL Server: SQL Statistics	Batch Requests/Secv	See Description	Number of batch requests received per second, and is a good general indicator for the activity level of the SQL Server. This counter is highly dependent on the hardware and quality of code running on the server. The more powerful the hardware, the higher this number can be, even on poorly coded applications. A value of 1000 batch requests/sec is easily attainable though a typical 100Mbs NIC can only handle about 3000 batch requests/sec.Many other counter thresh- olds depend upon batch requests/sec while, in some cases, a low (or high) number does not point to poor processing power. You should frequently use this counter in combination with other counters, such as processor utilization or user connections.In version 2000, "Transactions/ sec" was the counter most often used to measure overall activity, while versions 2005 and later use "Batch Requests/sec". Versions 2005 prior to SP2, measure this counter differently and may lead to some misunderstandings. Read the footnote for more details.
SQL Server: SQL Statistics	SQL Compila - tions/sec	< 10% of the number of Batch Re- quests/Sec	Number of times that Transact-SQL compilations occurred, per second (including recompiles). The lower this value is the better. High values often indicate excessive adhoc querying and should be as low as possible. If excessive adhoc querying is happening, try rewriting the queries as procedures or invoke the queries using sp_ex- ecuteSQL. When rewriting isn't possible, consider using a plan guide or setting the database to parameterization forced mode.
SQL Server: SQL Statistics	SQL Re- Compila- tions/sec	< 10% of the number of SQL Compila- tions/sec	Number of times, per second, that Transact-SQL objects attempted to be executed but had to be recompiled before completion. This number should be at or near zero, since recompiles can cause deadlocks and exclusive compile locks. This counter's value should follow in proportion to "Batch Requests/sec" and "SQL Compilations/ sec". This needs to be nil in your system as much as possible.
SQL Server: Deprecated Features	Usage	~0	Number of cancel and query timeouts per second or features used that are considered "dep- recated"; that is, those features and commands that Microsoft will not support in a release or two. Run this counter when considering an upgrade to a newer version of SQL Server, then update the application accordingly.
SQL Server: SQL Statistics	SQL Attention Rate/sec	~0	Number of cancels and query timeouts occurring per second. This number should be as low as possible. A high sustained number indicates frequent query timeout or end-user cancellation of queries.
SQL Server: Cursor Manager by Type	Active Cursors	See Description	Number of active cursors when polled. Monitor cursor counters to see if there may be heavy use of server cursors since improper use can result in performance issues.
SQL Server: SQL Errors	Errors/sec \\ DB Offline Errors and Errors/sec \\ Kill Connection Errors	~0	Number of errors per second which takes a data- base offline or kills a user connection, respec- tively. Since these are severe errors, they should occur very infrequently.

MSSQL Users & Locks Performance Counters			
Object	**Counter**	**You Want**	**Description**
SQLServer: General Statistics	Logins/sec and Logouts/sec	< 2	The number of user logins per second. Any value over 2 may indicate insufficient connection pooling.
SQLServer: General Statistics	User Connections	See Description	The number of users currently connected to the SQL Server. This counter should roughly track with “Batch Requests/Sec”. They should generally rise and fall together. For example, blocking problems could be revealed by rising user connections, lock waits and lock wait time coupled with declining batch requests/sec.
SQL Server: Latches	Latch Waits/sec	(Total Latch Wait Time) / (Latch Waits/ Sec) < 10	The number of latches in the last second that had to wait. Latches are lightweight means of holding a very transient server resource, such as an address in memory.
SQL Server: Latches	Avg Latch Wait Time (ms)	See Description	The average latch wait time, in millisec- onds, for any latch requests that had to wait. This value should generally correlate to “Latch Waits/sec” and move up or down with it accordingly.
SQL Server: Latches	Total Latch Wait Time (ms)	(Total Latch Wait Time) / (Latch Waits/ Sec) < 10	The total latch wait time in milliseconds spent waiting for a latch in the last second. This value should stay stable compared to the number of latch waits per second.
SQL Server: Locks	Lock Wait Time (ms)	See Description	The total time spent waiting across all transactions, in milliseconds, to acquire a lock in the last second. Because SQL Server records a lock at the end of a lock- ing event, remember that an application with huge transactions may have inflated lock wait times while still performing as expected. For example, an applica- tion that issues multi-million record updates might have very long lock wait times while performing exactly as it was designed.
SQL Server: Locks	Lock Waits/sec	0	How many times users waited to acquire a lock over the past second. Values greater than zero indicate at least some blocking is occurring, while a value of zero can quickly eliminate blocking as a potential root-cause problem. As with “Lock Wait Time”, lock waits are not recorded by Perf- Mon until after the lock event completes.
SQL Server: Locks	Avg Wait Time (ms)	<500	The average wait time, in milliseconds, for each lock request that had to wait. An average wait time longer than 500ms may indicate excessive blocking. This value should generally correlate to “Lock Waits/sec” and move up or down with it accordingly.
SQL Server: Wait Statistics	See Description	See Description	Reveals a variety of areas in which SQL Server might be waiting. Worth examin- ing when other more obvious avenues, such as locks and latches, have been exhausted.
SQL Server: Locks	Lock Requests/ sec	<1000	The number of new locks and locks converted per second. This metric’s value should generally correspond to “Batch Re- quests/sec”. Values > 1000 may indicate queries are accessing very large numbers of rows and may benefit from tuning.

SQL Server: Locks	Lock Timeouts/ sec	<1	Shows the number of lock requests per second that timed out, including internal requests for NOWAIT locks. A value greater than zero might indicate that user queries are not completing. The lower this value is, the better.
SQL Server: Locks	Number of Deadlocks/s ec	< 1	Number of lock requests, per second, which resulted in a deadlock. Since only a COMMIT, ROLLBACK, or deadlock can terminate a transaction (excluding failures or errors), this is an important value to track. Excessive deadlocking indicates a table or index design error or bad applica- tion design.
SQL Server: Access Methods	Worktables From Cache Ratio	>90%	Percentage of work tables created whose initial two pages were immediately avail- able from the worktable cache. A value less than 90% may indicate insufficient memory, since execution plans are be- ing dropped, or may indicate, on 32-bit systems, the need for an upgrade to a 64-bit system.
SQL Server: Access Methods	Table Lock Escalations/ sec	See Description	Number of times that SQL Server esca- lated locks from page- or row-level to table-level. This number should, generally, be low. Frequent or even occasional spiking in this value may indicate poorly coded transactions.
SQL Server: Transac- tions	Longest Running Transaction Time	See Description	The time, in seconds, of the longest run- ning transaction. When blocking is high, check this counter to see if transactions are open for long periods of time.

OS CPU & Processor Counters			
Object	**Counter**	**You Want**	**Description**
Process (sqlservr)	%Processor Time	< 80%	Percentage of processor time spent on SQL Server process threads. You may also wish to investigate other Process (sqlservr) such as Private Bytes, Virtual Bytes, Working Set, etc to get a fuller understanding of how SQL Server allocates certain segments of memory. Usually, these auxiliary counters provide contextual information and are not necessary for troubleshooting.
Process (msmdsrv)	%Processor Time	< 80%	Percentage of processor time spent on SSAS process threads.
Processor	%Processor Time	< 30% of total % Processor Time	Percentage of elapsed time that the process threads spent executing code in privileged mode, like SQL Server I/O requests. Poor performance here may be caused by an old or inefficient hardware driver.
Processor	%Processor Time	< 80%	Percentage of elapsed time the processor spends executing non-idle threads.
System	Processor Queue Length	< 4 per CPU	Number of threads waiting for CPU cycles, where < 12 per CPU is good/fair, < 8 is bet- ter, < 4 is best.
System	Context Switches/sec	< 4 per CPU	Number of execution contexts switched in the last second, where >6000 is poor, <3000 is good, and <1500 is excellent.

MSSQL User Database Performance Counters
Monitor these counters to determine general benchmark levels set by your user databases and for TempDB.

Object	Counter	You Want	Description
SQL Server: Databases	Data File(s) Size (KB)	See Description	Cumulative size (KB) of all the data files in the database including any automatic growth. Monitoring this counter is useful, for example, for determining the correct size of TempDB.
SQL Server: Databases	Log Bytes Flushed/ sec	See Description	Total number of log bytes flushed per second. Useful for determining trends and utilization of the transaction log
SQL Server: Databases	Log File(s) Size (KB)	See Description	Cumulative size, in (KB), of all the transaction log files for the specific database. Useful for deter- mining trends and utilization of the transaction log.
SQL Server: Databases	Log File(s) Used Size (KB)	See Description	The cumulative used size of all the log files in the database.
SQL Server: Databases	Log Flush Wait Time	~0	Total wait time, in milliseconds, to write all trans- action log pages.
SQL Server: Databases	Log Flush Waits/sec	~0	Effectively, the number of times per second that SQL Server must wait for pages to be written to the transaction log.
SQL Server: Databases	Log Flushes/sec	See Description	Technically, the number of log pages flushed to the transaction log per second.
SQL Server: Databases	Log Growths	~0	Total number of times the transaction log for the database has been expanded. Each time the transaction log grows, all user activity must halt until the log growth completes. Therefore, you want log growths to occur during predefined maintenance windows rather than during gen- eral working hours.
SQL Server: Databases	Log Shrinks	~0	Total number of times the transaction log for the database has been shrunk.
SQL Server: Databases	Log Truncations	See Description	Total number of times the transaction log has been truncated for the database specified. Truncations should happen during log backups or, on databases in simple recovery mode, at checkpoint or the time period specified by recovery interval.
SQL Server: Databases	Percent Log Used	<80%	Percentage of space in the log that is in use. Since all work in an OLTP database stops until writes can occur to the transaction log, it’s a very good idea to ensure that the log never fills completely. Hence, the recommendation to keep the log under 80% full.

SQL Server: Memory Manager Counters	
Counter	**Description**
Granted Workspace Memory (KB)	Total amount of memory currently granted to executing processes such as hash, sort, bulk copy, and index creation operations.
Maximum Workspace Memory (KB)	Maximum amount of memory available for executing processes such as hash, sort, bulk copy, and index creation operations.
Memory Grants Outstandin g	Total number of processes per second that have successfully acquired a workspace memory grant.
Memory Grants Pending[vii]	Total number of processes per second waiting for a workspace memory grant. Numbers higher than 0 indicate a lack of memory.
Target Server Memory (KB)	Total amount of dynamic memory the server can consume.

SQL Server SQL Statistics Counters	
Counter	**Description**
Auto-Param Attempts/sec	Number of auto-parameterization attempts per second. Total should be the sum of the failed, safe, and unsafe auto-parameterizations. Auto-parameterization occurs when an instance of SQL Server attempts to reuse a cached plan for a previously executed query that is similar to, but not the same as, the current query. For more information, see Auto- parameterization in the SQL Server Books On-Line (BOL).
Failed Auto-Params/sec	Number of failed auto-parameterization attempts per second. This should be small.
Safe Auto-Params/sec	Number of safe auto-parameterization attempts per second.
Unsafe Auto-Params/sec	A query is designated as unsafe when it has characteristics that prevent its cached plan from being shared.

SQL Server : Plan Cache : Cache Manager Instance Instances can be:**_total** (all instances in total), **SQL Plans** (cached query plans from ad-hoc Transact-SQL statements), **Object Plans** (query plans for stored procedures, user-defined functions and triggers), **Bound Trees** (normalized trees for views, rules, computed columns and check constraints), **Extended stored procedures** (number of these objects referenced in cache), **Temporary tables & table variables** (number of temp tables and table variables referenced in cache).	
Counter	**Description**
Ad hoc SQL Plans	Query plans produced from an ad hoc Transact-SQL query, including auto-parameterized queries. SQL Server caches the plans for ad hoc SQL statements for later reuse if the identical Transact-SQL statement is later executed.
Prepared SQL Plans	Query plans that correspond to Transact-SQL statements prepared using sp_prepare, sp_cursorprepare, or auto-parameterization. User-parameterized queries (even if not explicitly prepared) are also monitored as Prepared SQL Plans.
Procedure Plans	Query plans generated by creating a stored procedure.

MSSQL Data Access Performance Counters			
Object	**Counter**	**You Want**	**Description**
SQL Server: Access Methods	Forwarded Records/sec	< 10 per 100 Batch Requests/Sec	Identifies use of a pointer which has been created when variable length columns have caused a row to move to a new page in a heap.
SQL Server: Access Methods	Full Scans / sec	See Description	Monitors the number of full scans on tables or indexes. Ignore unless high CPU coincides with high scan rates. High scan rates may be caused by missing indexes, very small tables, or requests for too many records. A sudden increase in this value may indicate a statistics threshold has been reached, resulting in an index no longer being used.
SQL Server: Access Methods	Index Searches/sec	1 Full Scan/sec per 1000 Index Searches/sec	Monitors the number of index searches when doing range scans, single index record fetches, and repositioning within an index. The threshold recommendation is strictly for OLTP workloads.
SQL Server: Access Methods	Page Splits/sec	< 20 per 100 Batch Requests/Sec	Monitors the number of page splits per second which occur due to overflowing index pages and should be as low as possible. To avoid page splits, review table and index design to reduce non-sequential inserts or implement fillfactor and pad_index to leave more empty space per page. **NOTE:** A high value for this counter is not bad in situations where many new pages are being created, since it includes new page allocations.
SQL Server: Access Methods	Workfiles Created/sec	<20	Number of work files created per second, usually as a part of **tempdb** processing when working with hashing joins and other hashing operations. High values can indicate thrash in **tempdb** and poorly coded queries.
SQL Server: Access Methods	Worktables Created/sec	<20	Number of work tables created per second, usually as a part of **tempdb** processing when working with spools such as table spools, index spools, etc.

Red Herring Counters

Microsoft SQL Server has a long and storied history. Some PerfMon counters, once among the most useful available, are now much less useful to the point of being counter-productive. In many cases, improvements to the hardware or internal processes within SQL Server mean these PerfMon counters, while once informative, no longer add value.

Counter	**Explanation**
Physical Disk: % Disk Time	This counter is deceptive because it makes no accommodation for multiple spindles. Thus, the more spindles (i.e. physical hard disks) you have, the higher the percentile values can go. Conversely, if these spindles are shared across LUNs or other services, you may have high numbers on this counter without any correlation to SQL Server activity. In short, there are better ways to find out SQL Server's I/O performance.
Physical Disk: Avg Disk Queue Lengths	The way in which Windows measures disk queues, combined with the amount of cache that storage vendors provide with hard disk controllers, SANs, and hard disks themselves means that Windows might perceive that data is written all the way to disk, when in fact the data is actually sitting in a hardware-level cache somewhere.

Physical Disk: Transfer/sec	Using this counter, by itself, is not useful since you cannot tell how many reads or writes, separately, are happening. In addition, there's not really a useable "You Want" sort of threshold for this counter. This counter is still occasionally useful for a "stalled I/O" type of issue, but only in correlation with the other I/O counters mentioned earlier.
Processor: % Processor Time	Only useful on physical machines. It's value drops dramatically on virtual machines since you cannot know if CPU issues are due to CPU scheduling, a throttle set by the hypervi- so, or a limited amount of CPU available due to other guest activity.
SQL Server: Buffer Manager : Buffer Cache Hit Ratio	Long a stalwart counter used by SQL Server DBAs, this counter is no longer very useful. It monitors the percentage of data requests answer from the buffer cache since the last reboot. However, other counters are much better for showing current memory pressure that this one because it blows the curve. For example, PLE (page life expectancy) might suddenly drop from 2000 to 70, while buffer cache hit ration moves only from 98.2 to 98.1. Only be concerned by this counter if it's value is regularly below 90 (for OLTP) or 80 (for very large OLAP).

MSSQL "How is My Memory Being Used" Performance Counters[iv]			
Object	**Counter**	**You Want**	**Description**
SQL Server: Buffer Manager	Database Pages	See Description	Number of database pages in the buffer pool, as opposed to other usages for memory such as free pages, procedure cache, etc.
SQL Server: Buffer Manager	Procedure Cache Pages	See Description	Number of pages used to store compiled que- ries and objects.
SQL Server: Buffer Manager	Target Pages	See Description	The ideal number of pages in the buffer pool according the maximum memory granted to SQL Server in sp_configure.
SQL Server: Buffer Manager	Target Pages	See Description	Total number of pages in the buffer pool (including database, free, and stolen pages).
SQL Server: Buffer Manager	Free Pages	> 640	Total number of pages available across all free lists. A value less than 640 (5MB) indicates physical memory pressure.
SQL Server: Buffer Manager	Stolen Pages/sec	See Description	Tells how many pages were "stolen" from the buffer pool to satisfy other memory needs, such as plan cache and workspace memory. This number is a good metric to determine how much data is flowing into SQL Server caches and should remain proportionate to "Batch Requests/sec". Also remember to look for where these stolen pages might be stolen from – optimizer memory, lock memory, and so forth.
SQL Server: Memory Manager	Total Server Memory(KB)	See Description	Shows the amount of memory that SQL Server is currently using. This value should grow until its equal to Target Server Memory, as it popu- lates its caches and loads pages into memory. When it has finished, SQL Server is said to be in a "steady-state". Until it is in steady-state, per- formance may be slow and IO may be higher.
SQL Server: Memory Manager	Target Server Memory(KB)	See Description	Shows the amount of memory that SQL Server wants to use based on the configured Max Server Memory.