Free-SQL Server Performance Monitoring Tool (Author: Sarjen Haque)

FREE: Real-time SQL Server Performance Monitor - Extreme visibility to SQL Server database engine

http://sqltouch.blogspot.com/2014/03/free-real-time-sql-server-performance.html

New Replacement Tool: https://sqltouch.blogspot.com/2019/11/sql-server-diagnostic-tool-for-sql-2012.html

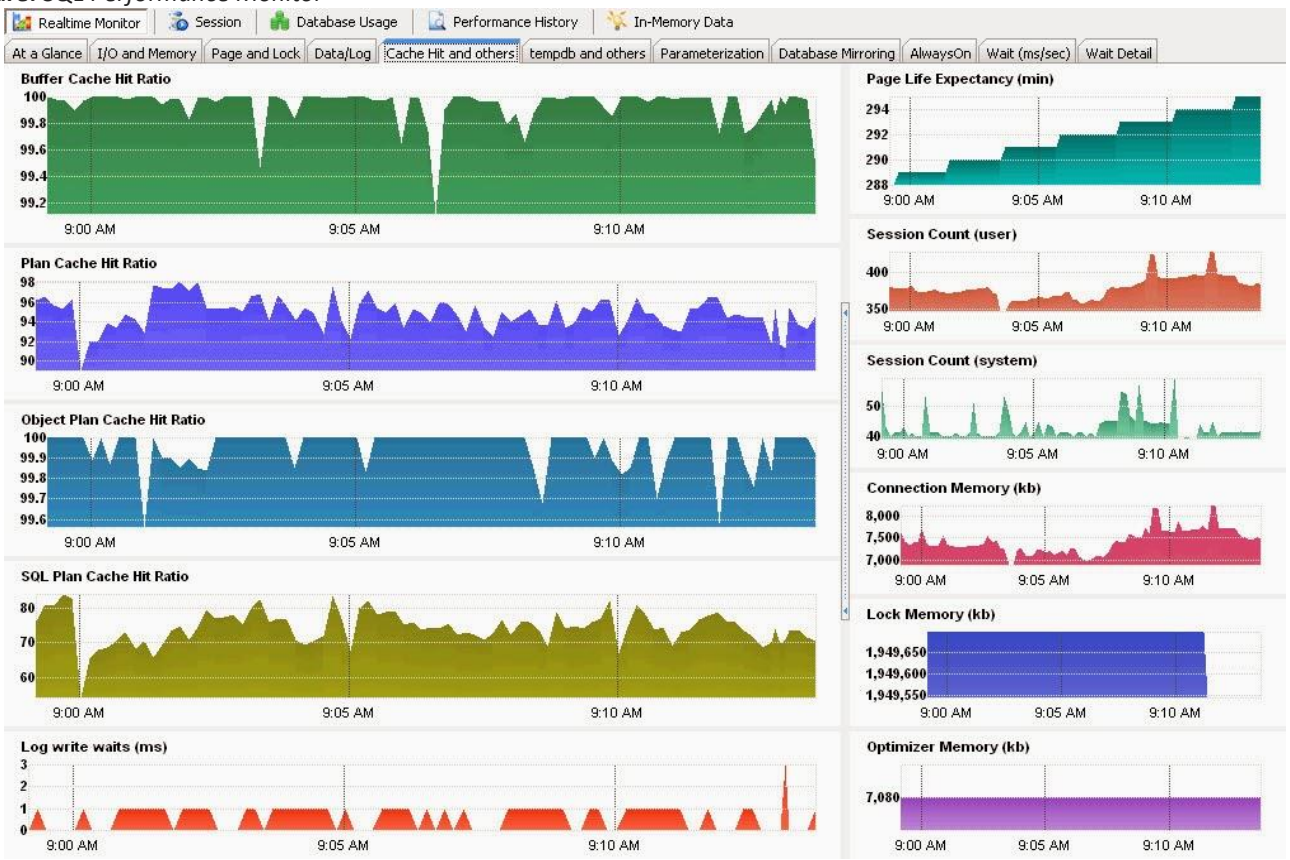
Background of the tool:

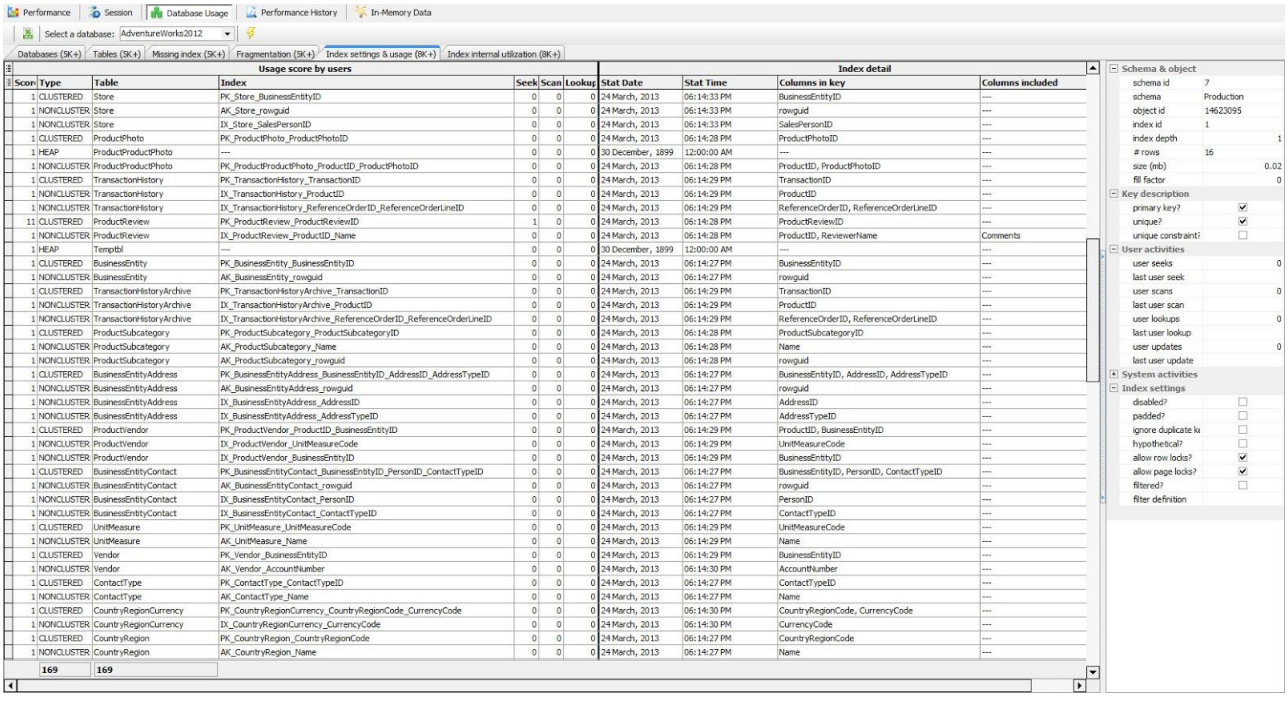
- As database administrators, we have to support different SQL Server environments.
- Often it becomes a challenge and obvious to understand the current server health status.
- To attain this goal based on my requirements, I have created this small tool just for fun with my limited development application skill.
- It is completely Free, Agent less, No Installation/Configuration is required, Single executable and portable, easy to use and only needs a couple of clicks to be up and running.

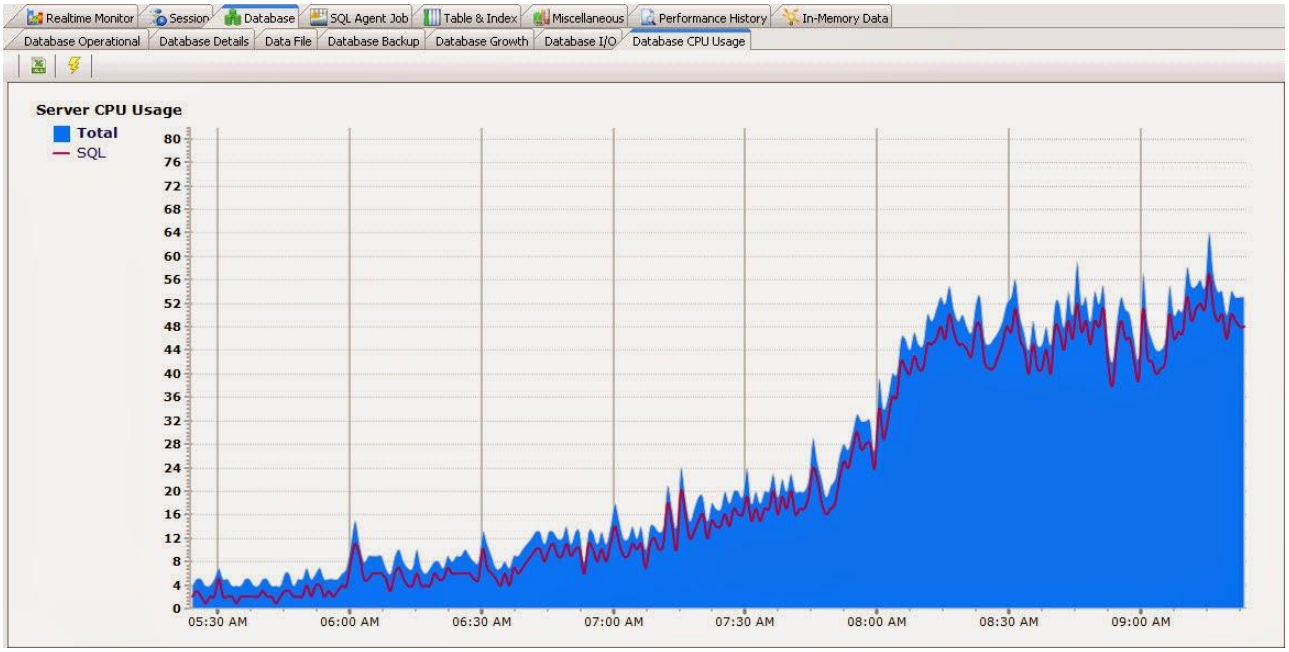
Agreement:

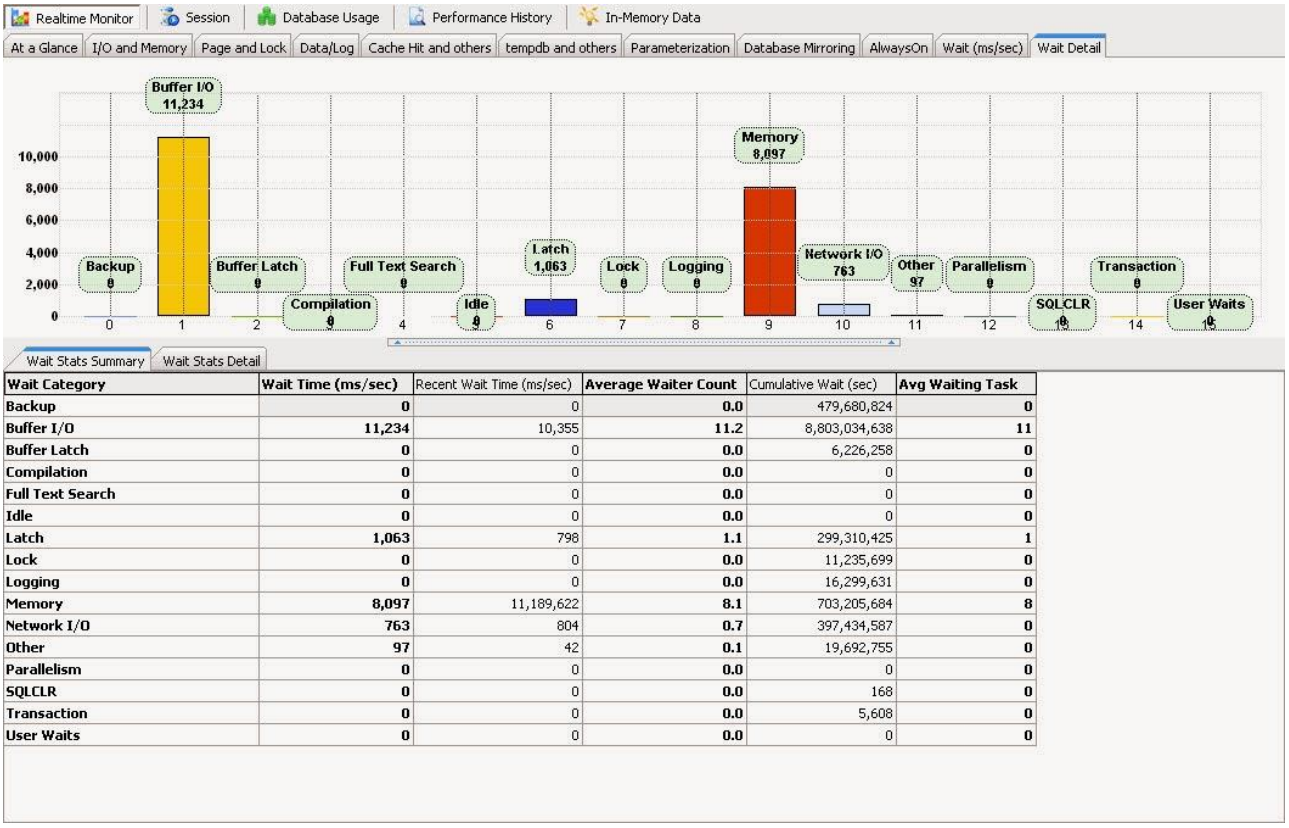
- This is a non-commercial, educational and learning purpose tool.
- It is not an alternative for any commercial grade application.
- **This tool is efficient and sharp like a blade**; however, I will not able to provide any warranty, guarantee or accuracy of this tool.
- Although it is a lightweight data collection and visualization tool, it should not cause any performance issues, **however you should test it yourself before running it against any database server.**

Figure: *SQL Performance Monitor*









Wait Stats Summary | Wait Stats Detail

Wait Category	Wait Time (ms/sec)	Recent Wait Time (ms/sec)	Average Waiter Count	Cumulative Wait (sec)	Avg Waiting Task
Backup	0	0	0.0	479,680,824	0
Buffer I/O	11,234	10,355	11.2	8,803,034,638	11
Buffer Latch	0	0	0.0	6,226,258	0
Compilation	0	0	0.0	0	0
Full Text Search	0	0	0.0	0	0
Idle	0	0	0.0	0	0
Latch	1,063	798	1.1	299,310,425	1
Lock	0	0	0.0	11,235,699	0
Logging	0	0	0.0	16,299,631	0
Memory	8,097	11,189,622	8.1	703,205,684	8
Network I/O	763	804	0.7	397,434,587	0
Other	97	42	0.1	19,692,755	0
Parallelism	0	0	0.0	0	0
SQLCLR	0	0	0.0	168	0
Transaction	0	0	0.0	5,608	0
User Waits	0	0	0.0	0	0

A challenge:

- Retrieving and visualizing the SQL Server performance data is always a challenge and a tedious task for SQL Server database professionals.
- Utilizing the Windows PerfMon application is the easiest way to perform this task as well as querying "*sys.dm_os_performance_counters"* and some other DMVs brings a lot of useful information.
- Starting from SQL Server 2005, Microsoft has introduced DMV to query various internal metadata directly to explore various health status data.
- Although collecting and analyzing SQL Server performance data in a regular basis provides trending ability, monitoring real-time performance data is critical to understand an ongoing performance condition that is occurring.
- We are all familiar with built-in "SQL Server Activity Monitor" and obviously it is a good starting point to troubleshoot some SQL Server issues. However, the capacity of this tool is limited as it does not provide other performance metrics which are important to understand the server health status. To extend this idea especially during a

performance condition, I have attempted to develop a "SQL Performance Monitor" desktop app by including some other interesting metrics which I believe might be helpful to troubleshoot or understand a problem.

This tool collects more than 50+ performance data directly from SQL Server in real-time and shows data in the chart continuously. Also, it does not require any installation and configuration.

Data collection:

SQL Scripts used in my tool are excerpted from SSMS and some are collected from various forums which are freely available. My understanding is that all the scripts that I have used are reliable however if any are not working, please let me know and I will attempt to fix the issue.

How does it work?

1. Has the ability to monitor only a single SQL instance at a time and can be used against **all editions of SQL Server from 2005 to SQL 2014**.
2. Charts and grids will be populated with collected performance data every 5 seconds by default (can be changed) for every 5 minutes (can be changed) moving forward.
3. Performance data will be saved automatically as they are collected in a **SQLite database (sqlmonitor.db3).**
4. All saved performance data can be queried, and then can be exported as a CSV format. As "sqlmonitor.db3" is not protected therefore it can be opened with any SQLite tool.

Limitations:

1. It has no notification system, such as email, alert, popup.
2. It is a desktop 32-bit application, cannot run as a service.
3. Chart colours have no special meaning.

Known Limitations:

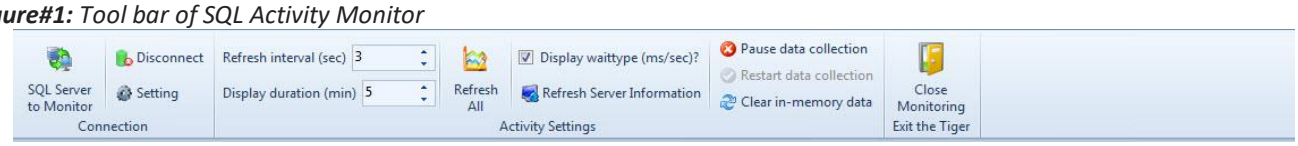
(a) SQL 2005 – in the "Server Info" tab the "Available Memory" will be zero.
(b) CPU utilization has been calculated from "Resource Pool" and @@CPU_BUSY. Due to the internal limitation of SQL Server, and feature limitation of Standard and Express editions, CPU value may show zero on the chart. In Enterprise edition, CPU utilization will not be zero.

How to run:

(a) Create a folder.
(b) Download the "SQLMonitor.exe" in that folder.
(c) Run the executable "SQLMonitor.exe"– that's it.
(d) There is no extra configuration or components required to run this tool.

Connect to a database server:
The tool bar of "SQL Performance Monitor"

Figure#1: *Tool bar of SQL Activity Monitor*



First time connection:

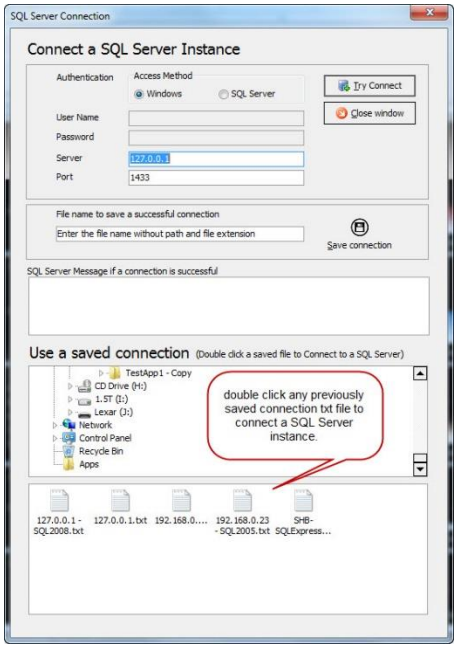
- To connect a SQL Server instance, click the **"SQL Server to Monitor"** button.
- Supply the required information and then click **"Try Connect"** in the connection dialog box.
- Once connected, close the connection dialog box or choose another server to connect to.
- All charts will be populated for an hour with blank data once a connection is made.
- It continues to collect and display data based on the duration configured on the tool bar.
- All collected data will be saved in a SQLite database (sqlmonitor.db) for later review and analysis.

Using a saved connection:

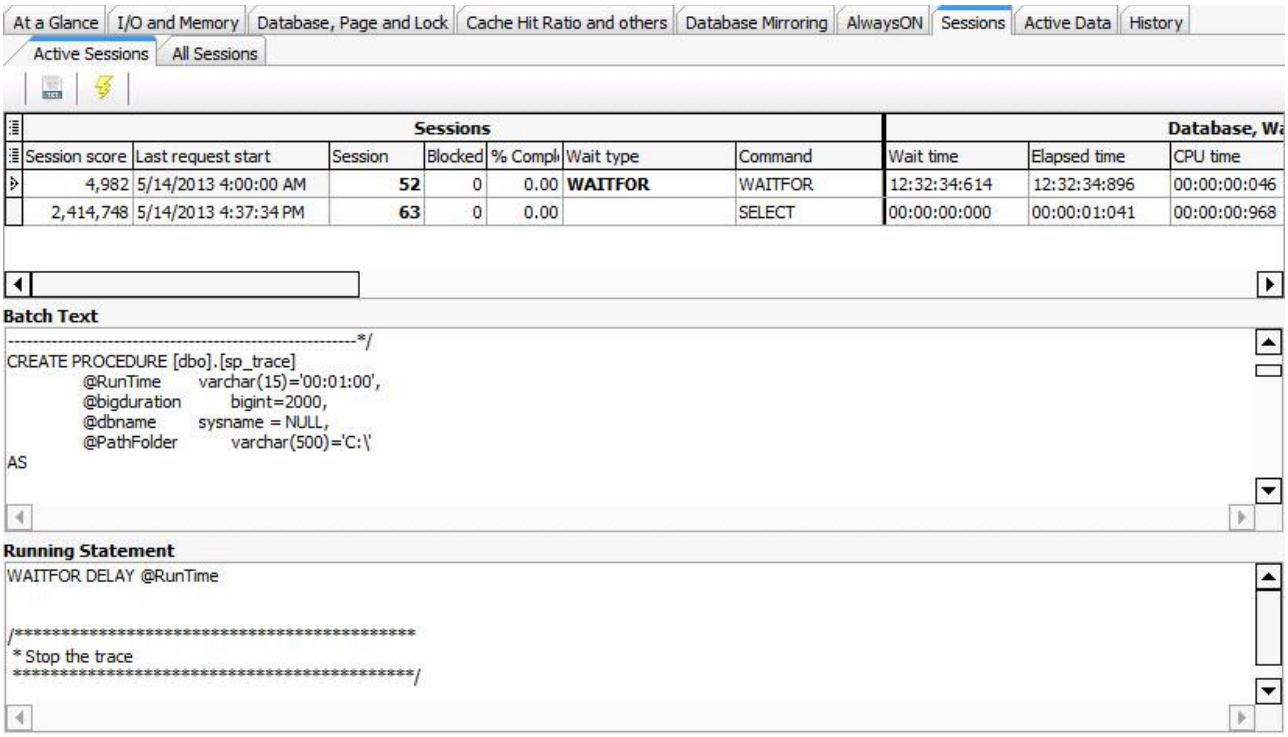
- A successful connection can be saved for later use.
- Once the tool successfully connects to a database server, click the “save connection” button to save the connection string.
- An encoded text file will be created in the same folder with the “.txt” extension where the “SQLMonitor.exe” resides.
- From the bottom list box of the “SQL Server Connection” (**figure#2)** dialog box, double click a previously saved file to connect to a SQL Server instance.

Couple of Screenshots from “SQL Performance Monitor”

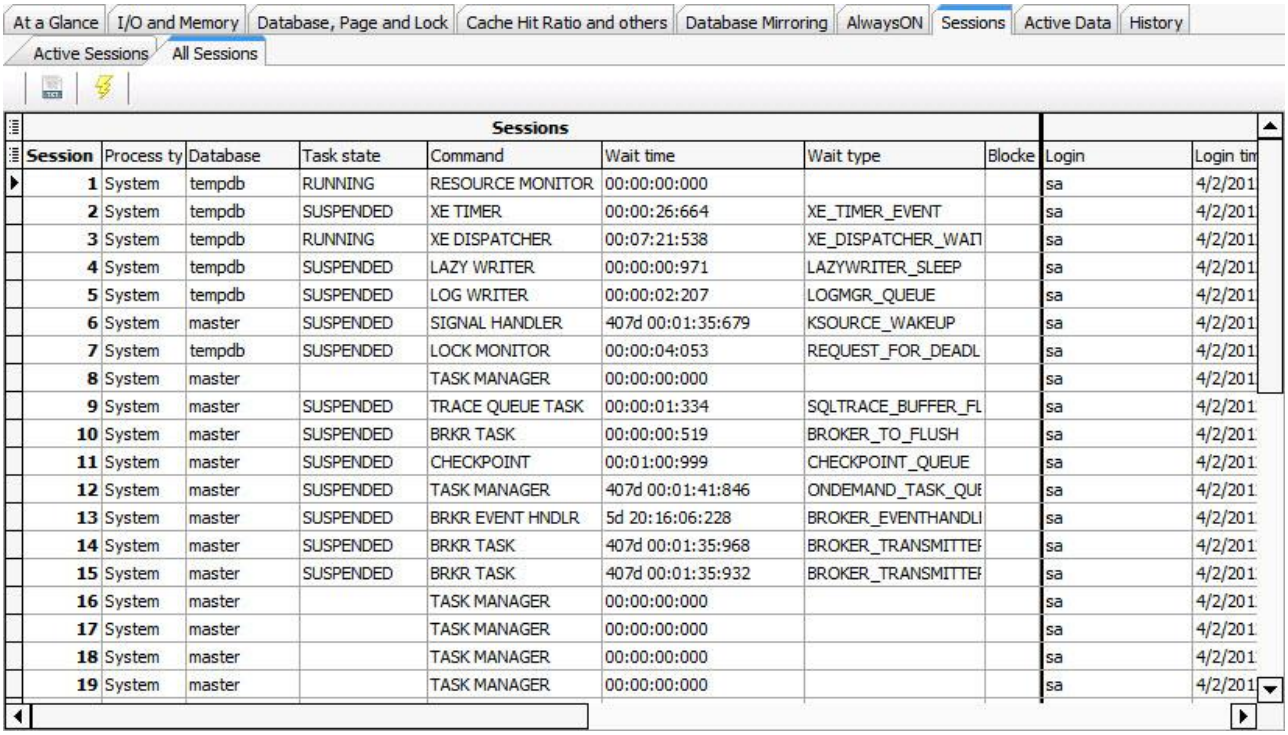
Figure#2: *SQL Server Connection dialog*



Figure#3A: Viewing all running sessions



Figure#3B: Viewing all sessions



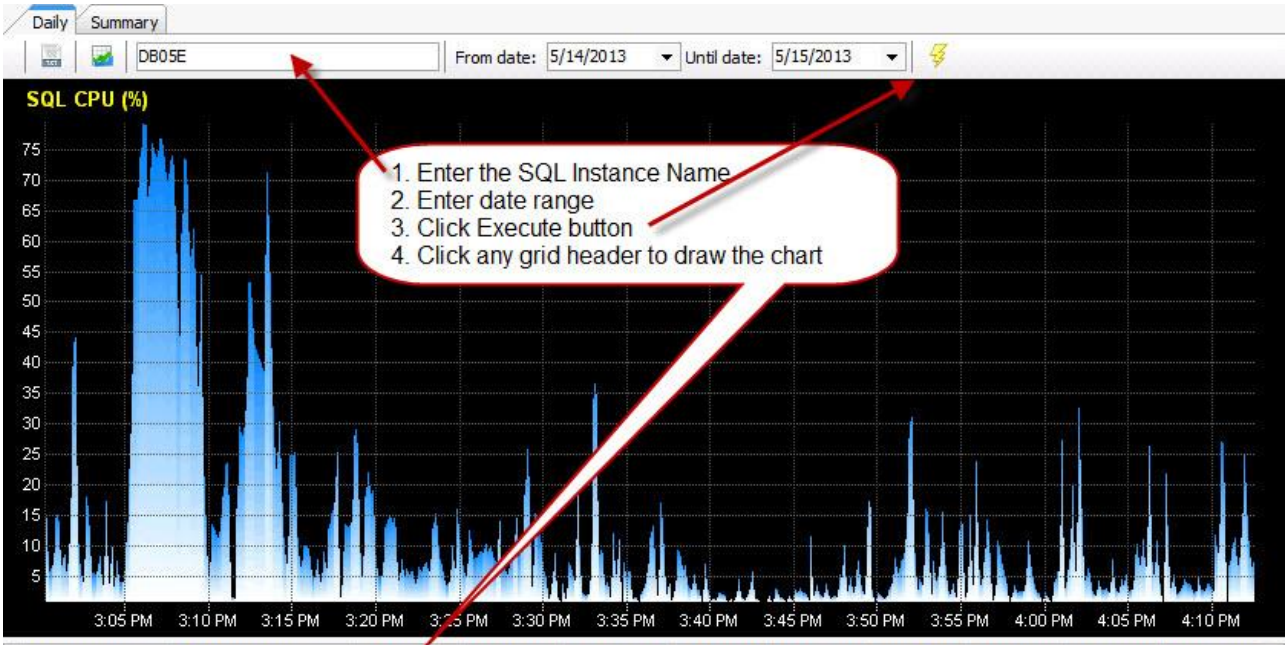
Sessions

Session	Process ty	Database	Task state	Command	Wait time	Wait type	Blocke	Login	Login tim
1	System	tempdb	RUNNING	RESOURCE MONITOR	00:00:00:000			sa	4/2/201
2	System	tempdb	SUSPENDED	XE TIMER	00:00:26:664	XE_TIMER_EVENT		sa	4/2/201
3	System	tempdb	RUNNING	XE DISPATCHER	00:07:21:538	XE_DISPATCHER_WAIT		sa	4/2/201
4	System	tempdb	SUSPENDED	LAZY WRITER	00:00:00:971	LAZYWRITER_SLEEP		sa	4/2/201
5	System	tempdb	SUSPENDED	LOG WRITER	00:00:02:207	LOGMGR_QUEUE		sa	4/2/201
6	System	master	SUSPENDED	SIGNAL HANDLER	407d 00:01:35:679	KSOURCE_WAKEUP		sa	4/2/201
7	System	tempdb	SUSPENDED	LOCK MONITOR	00:00:04:053	REQUEST_FOR_DEADL		sa	4/2/201
8	System	master		TASK MANAGER	00:00:00:000			sa	4/2/201
9	System	master	SUSPENDED	TRACE QUEUE TASK	00:00:01:334	SQLTRACE_BUFFER_FL		sa	4/2/201
10	System	master	SUSPENDED	BRKR TASK	00:00:00:519	BROKER_TO_FLUSH		sa	4/2/201
11	System	master	SUSPENDED	CHECKPOINT	00:01:00:999	CHECKPOINT_QUEUE		sa	4/2/201
12	System	master	SUSPENDED	TASK MANAGER	407d 00:01:41:846	ONDEMAND_TASK_QUE		sa	4/2/201
13	System	master	SUSPENDED	BRKR EVENT HNDLR	5d 20:16:06:228	BROKER_EVENTHANDLI		sa	4/2/201
14	System	master	SUSPENDED	BRKR TASK	407d 00:01:35:968	BROKER_TRANSMITTEF		sa	4/2/201
15	System	master	SUSPENDED	BRKR TASK	407d 00:01:35:932	BROKER_TRANSMITTEF		sa	4/2/201
16	System	master		TASK MANAGER	00:00:00:000			sa	4/2/201
17	System	master		TASK MANAGER	00:00:00:000			sa	4/2/201
18	System	master		TASK MANAGER	00:00:00:000			sa	4/2/201
19	System	master		TASK MANAGER	00:00:00:000			sa	4/2/201

Historical data:

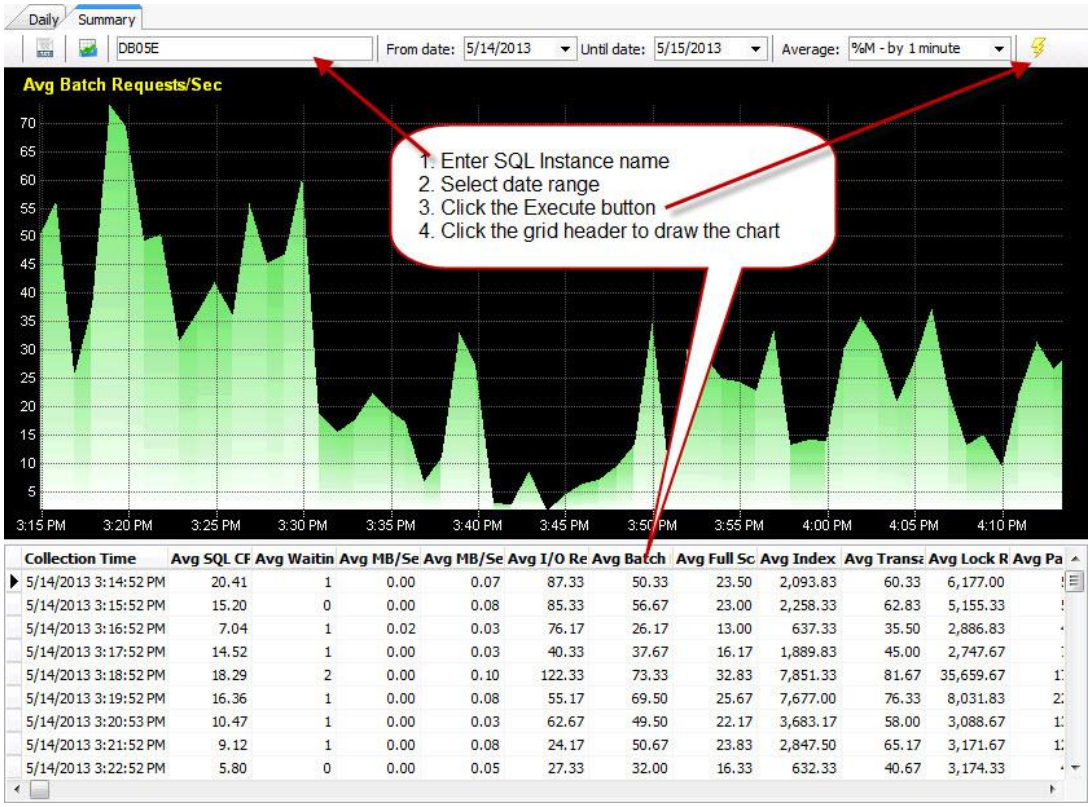
- In the history tab, put “SQL Instance Name” and “date” to query historical data.
- Click any column header to view data in the chart.
- All data and charts can be saved.

Figure#4: *Historical data browse*

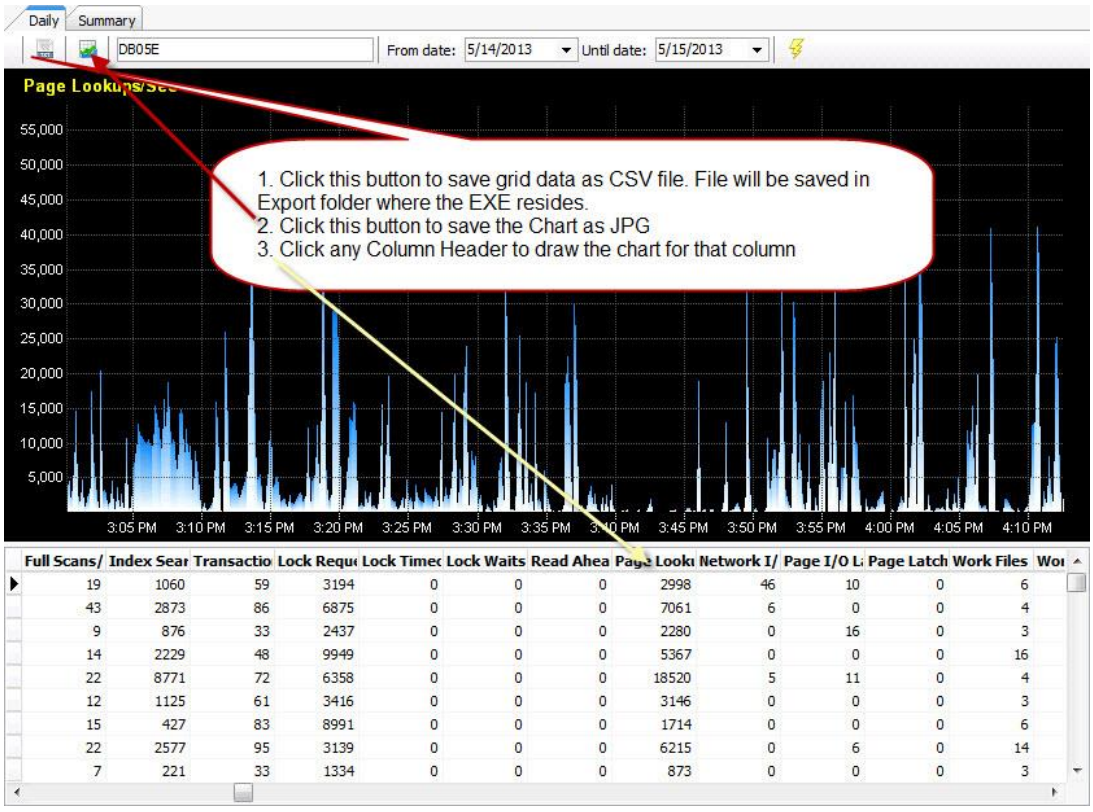


SQL Instance	Collection Time	SQL CPU (%)	Signal Wait	MB/Sec Re	MB/Sec Wr	I/O Respor	Page Read	Page Write	Full Scans/	Index Sea
DB05E	5/14/2013 3:00:12 PM	21.31	18.27	0	0.1	76	1	0	19	1060
DB05E	5/14/2013 3:00:21 PM	12.49	18.27	0	0.1	25	0	0	43	2873
DB05E	5/14/2013 3:00:31 PM	5.92	18.27	0	0	41	2	0	9	876
DB05E	5/14/2013 3:00:41 PM	6.96	18.27	0	0.2	43	0	17	14	2229
DB05E	5/14/2013 3:00:51 PM	14.63	18.27	0	0	72	2	1	22	8771
DB05E	5/14/2013 3:01:01 PM	16.48	18.27	0	0	59	0	1	12	1125
DB05E	5/14/2013 3:01:11 PM	6.98	18.27	0	0	35	0	0	15	427
DB05E	5/14/2013 3:01:21 PM	11.5	18.27	0	0.2	58	1	22	22	2577
DB05E	5/14/2013 3:01:31 PM	5.94	18.27	0	0	26	0	0	7	221

Figure#5: *Summarizing historical data*



Collection Time	Avg SQL CF	Avg Waitin	Avg MB/Se	Avg MB/Se	Avg I/O Re	Avg Batch	Avg Full Sc	Avg Index	Avg Transa	Avg Lock R	Avg Pa
5/14/2013 3:14:52 PM	20.41	1	0.00	0.07	87.33	50.33	23.50	2,093.83	60.33	6,177.00	
5/14/2013 3:15:52 PM	15.20	0	0.00	0.08	85.33	56.67	23.00	2,258.33	62.83	5,155.33	
5/14/2013 3:16:52 PM	7.04	1	0.02	0.03	76.17	26.17	13.00	637.33	35.50	2,886.83	
5/14/2013 3:17:52 PM	14.52	1	0.00	0.03	40.33	37.67	16.17	1,889.83	45.00	2,747.67	
5/14/2013 3:18:52 PM	18.29	2	0.00	0.10	122.33	73.33	32.83	7,851.33	81.67	35,659.67	
5/14/2013 3:19:52 PM	16.36	1	0.00	0.08	55.17	69.50	25.67	7,677.00	76.33	8,031.83	
5/14/2013 3:20:53 PM	10.47	1	0.00	0.03	62.67	49.50	22.17	3,683.17	58.00	3,088.67	
5/14/2013 3:21:52 PM	9.12	1	0.00	0.08	24.17	50.67	23.83	2,847.50	65.17	3,171.67	
5/14/2013 3:22:52 PM	5.80	0	0.00	0.05	27.33	32.00	16.33	632.33	40.67	3,174.33	



Full Scans/	Index Sear	Transactio	Lock Requ	Lock Timec	Lock Waits	Read Ahea	Page Looki	Network I/	Page I/O L	Page Latch	Work Files	Wo
19	1060	59	3194	0	0	0	2998	46	10	0	6	
43	2873	86	6875	0	0	0	7061	6	0	0	4	
9	876	33	2437	0	0	0	2280	0	16	0	3	
14	2229	48	9949	0	0	0	5367	0	0	0	16	
22	8771	72	6358	0	0	0	18520	5	11	0	4	
12	1125	61	3416	0	0	0	3146	0	0	0	3	
15	427	83	8991	0	0	0	1714	0	0	0	6	
22	2577	95	3139	0	0	0	6215	0	6	0	14	
7	221	33	1334	0	0	0	873	0	0	0	3	

Summary:

I use this tool on a regular basis and I hope someone may find it useful too. I will continue to add more features, so if you like it - check back often for updates.

Free: SQL Server Diagnostic Tool for SQL 2012 – SQL 2019

https://sqltouch.blogspot.com/2019/11/sql-server-diagnostic-tool-for-sql-2012.html

Portable Tool:..... SQL Digger

(SQLDigger.exe - Main executable

SQLDigger.txt - Encoded Configuration file)

Version Supports... SQL 2012 and above

Last Update:..... **2020.04.16 (removed some restrictions)**

Usage Type:.....Free, strictly non-commercial, and educational.

Download:..... http://bit.ly/2NTn6Zi

Purpose of this tool:

- As database administrator, I have to support different SQL Server environments.
- Often it becomes a challenge and obvious to understand the current server health status.
- To attain this goal based on my requirements, I have created this small tool just for fun with my limited development application skill.
- **It is Free, Agent-less, No Installation/Configuration is required, Single executable and portable, easy to use and only needs a couple of clicks to be up and running.**

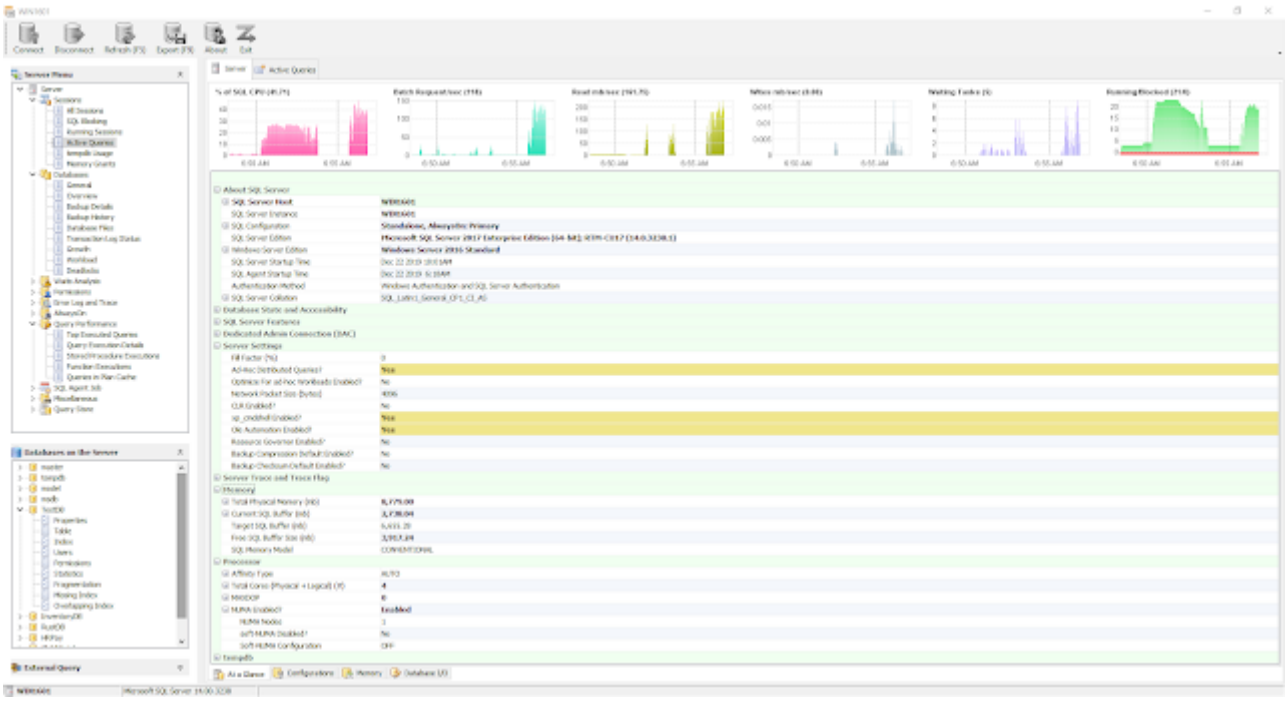
Background for the new development:

- The latest SQL Server release is bringing a number of new features and is changing the underlying functionality and behaviour of SQL Server.
- Thus, creating a new and updated diagnostic tool makes sense to help my DBA responsibilities and improve my regular workflow.
- The new tool is meant to be a replacement for my old “SQLMonitor.exe” which can be found at this link: http://sqltouch.blogspot.com/2014/03/free-real-time-sql-server-performance.html

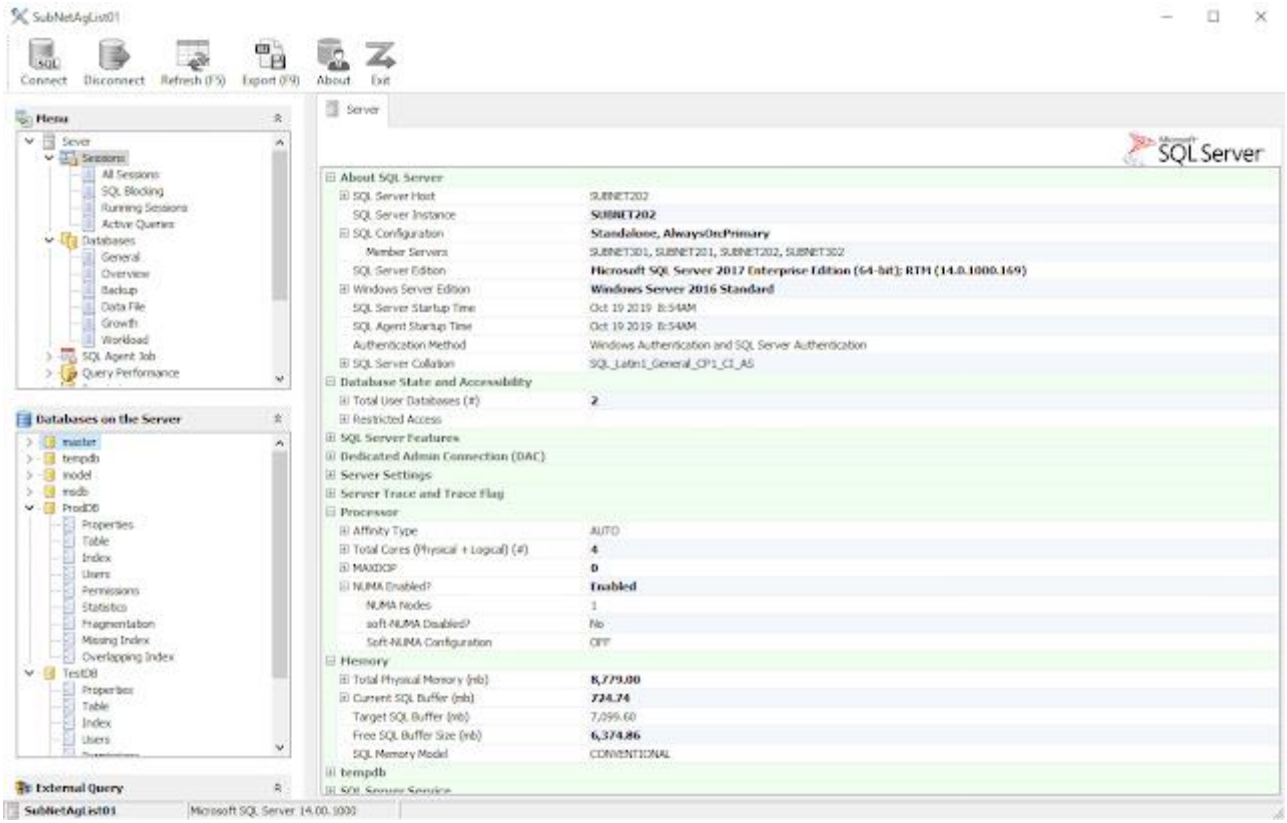
Development and distribution:

- **As I am developing this tool, new features will be added on an ongoing basis.**
- **This tool is free, non-commercial and only for educational purposes, so as always, use at your own risk.**
- If you’ve used (or are currently using) my previous tool, then you might be interested in testing this new tool as well.
- If you need an uncompressed version of this tool to avoid false positive alert from your Anti-Virus, then you may contact with me describing/providing the reason and I’ll try to forward you the **uncompressed version of this tool**.

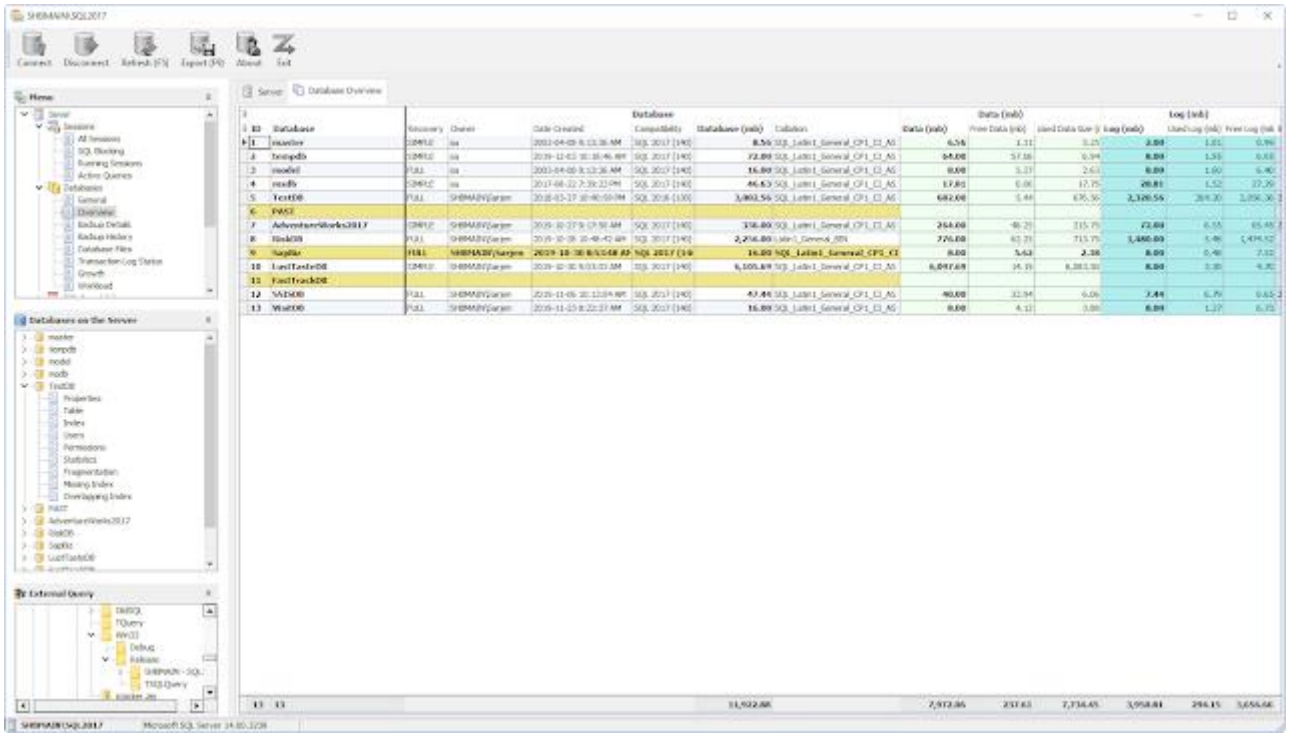
Screenshot #1: Connecting a SQL Server



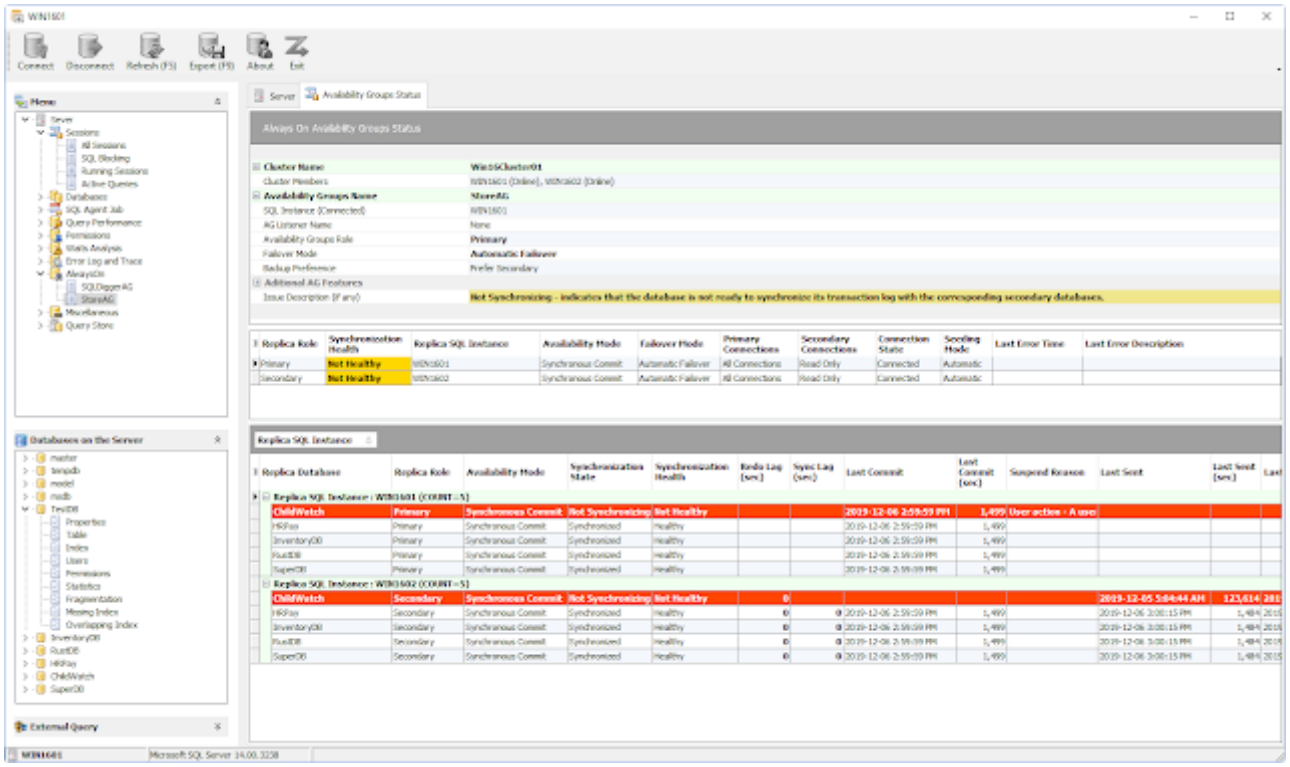
Screenshot #2: Connecting a multi-subnet AlwaysOn Availability Group



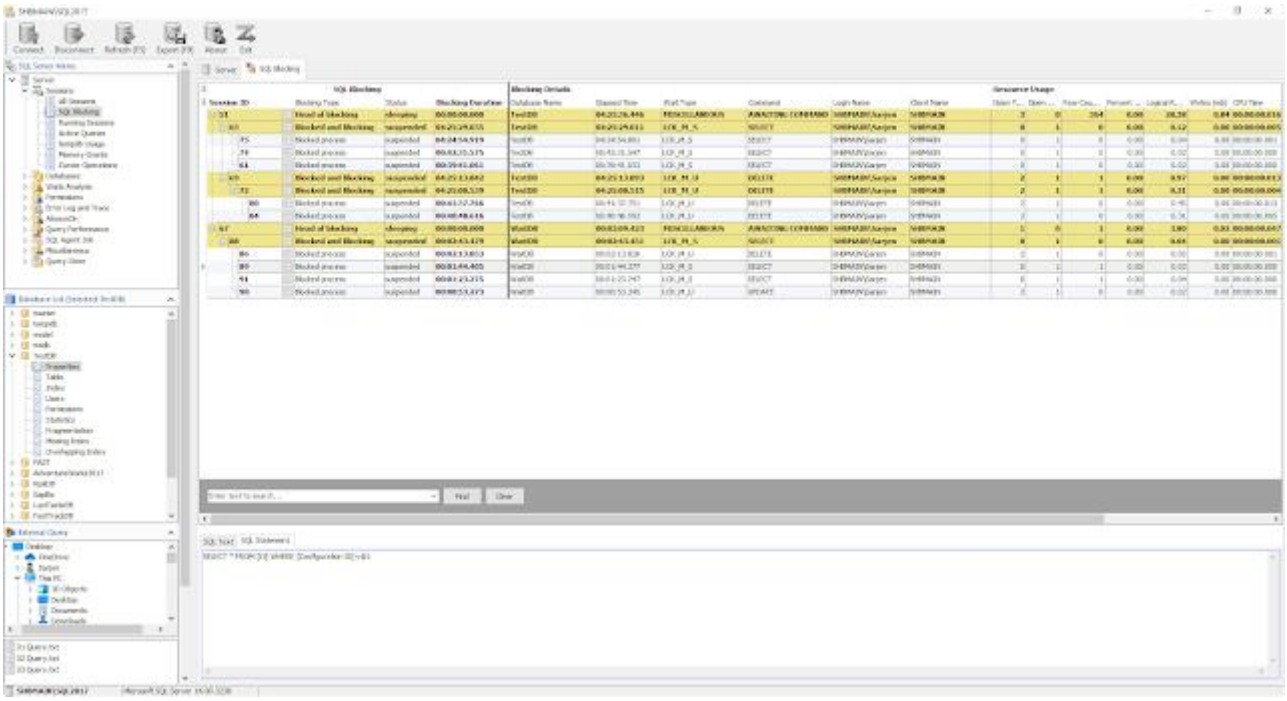
Screenshot #3: Detailed view of all databases on a server



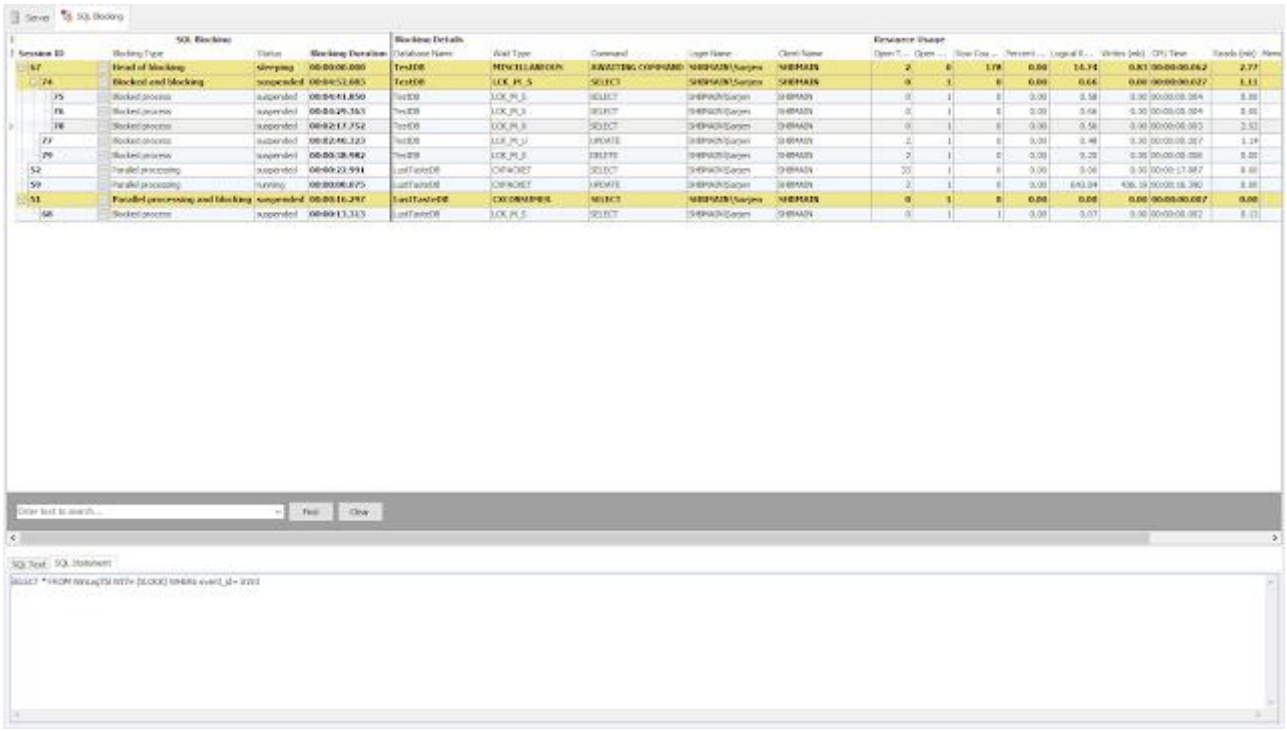
Screenshot #4: AlwaysOn Availability Group Health status



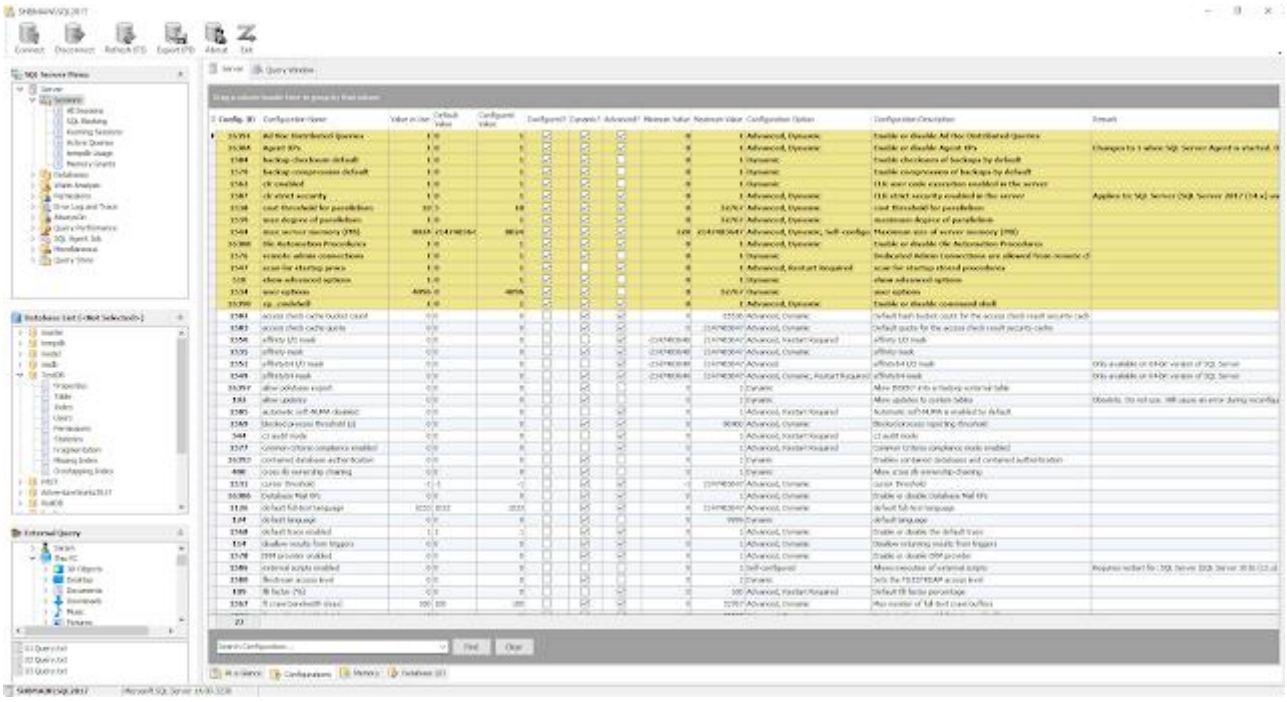
Screenshot #5: Details view of blocking



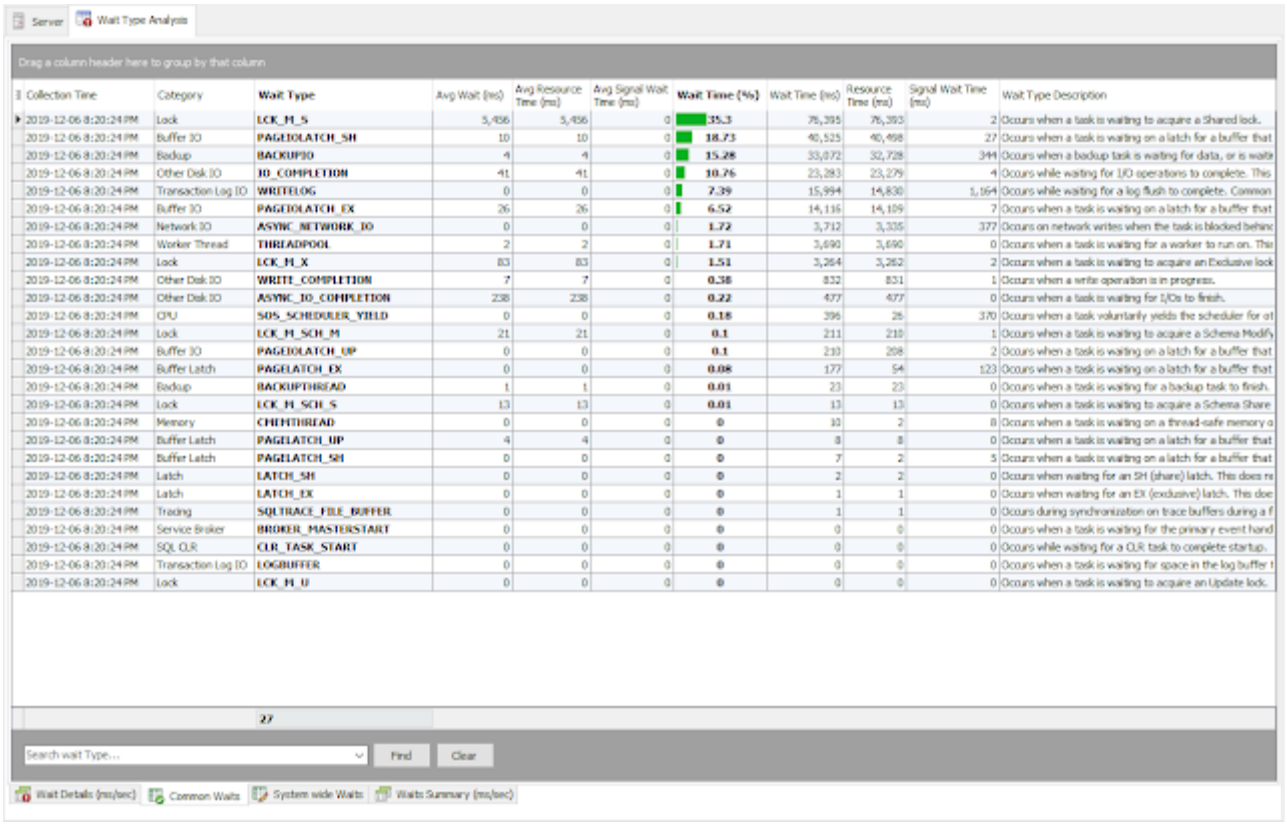
Screenshot #5(a): Details view of blocking including parallel processes



Screenshot #6: Server configurations.



Screenshot #7: Wait Type Analysis with wait type group and description



Server | Wait Type Analysis

Drag a column header here to group by that column

Collection Time	Category	Wait Type	Avg Wait (ms)	Avg Resource Time (ms)	Avg Signal Wait Time (ms)	Wait Time (%)	Wait Time (ms)	Resource Time (ms)	Signal Wait Time (ms)	Wait Type Description
2019-12-06 8:20:24 PM	Lock	LCK_M_S	5,456	5,456	0	35.3	76,395	76,393	2	Occurs when a task is waiting to acquire a Shared lock.
2019-12-06 8:20:24 PM	Buffer IO	PAGEIOLATCH_SH	10	10	0	18.73	40,525	40,498	27	Occurs when a task is waiting on a latch for a buffer that
2019-12-06 8:20:24 PM	Backup	BACKUPIO	4	4	0	15.28	33,072	32,728	344	Occurs when a backup task is waiting for data, or is waiti
2019-12-06 8:20:24 PM	Other Disk IO	IO_COMPLETION	41	41	0	10.76	23,283	23,279	4	Occurs while waiting for I/O operations to complete. This
2019-12-06 8:20:24 PM	Transaction Log IO	WRITELOG	0	0	0	7.39	15,994	14,830	1,164	Occurs while waiting for a log flush to complete. Common
2019-12-06 8:20:24 PM	Buffer IO	PAGEIOLATCH_EX	26	26	0	6.52	14,116	14,109	7	Occurs when a task is waiting on a latch for a buffer that
2019-12-06 8:20:24 PM	Network IO	ASYNC_NETWORK_IO	0	0	0	1.72	3,712	3,335	377	Occurs on network writes when the task is blocked behinc
2019-12-06 8:20:24 PM	Worker Thread	THREADPOOL	2	2	0	1.71	3,690	3,690	0	Occurs when a task is waiting for a worker to run on. This
2019-12-06 8:20:24 PM	Lock	LCK_M_X	83	83	0	1.51	3,264	3,262	2	Occurs when a task is waiting to acquire an Exclusive lock
2019-12-06 8:20:24 PM	Other Disk IO	WRITE_COMPLETION	7	7	0	0.38	832	831	1	Occurs when a write operation is in progress.
2019-12-06 8:20:24 PM	Other Disk IO	ASYNC_IO_COMPLETION	238	238	0	0.22	477	477	0	Occurs when a task is waiting for I/Os to finish.
2019-12-06 8:20:24 PM	CPU	SOS_SCHEDULER_YIELD	0	0	0	0.18	396	26	370	Occurs when a task voluntarily yields the scheduler for ot
2019-12-06 8:20:24 PM	Lock	LCK_M_SCH_M	21	21	0	0.1	211	210	1	Occurs when a task is waiting to acquire a Schema Modify
2019-12-06 8:20:24 PM	Buffer IO	PAGEIOLATCH_UP	0	0	0	0.1	210	208	2	Occurs when a task is waiting on a latch for a buffer that
2019-12-06 8:20:24 PM	Buffer Latch	PAGELATCH_EX	0	0	0	0.08	177	54	123	Occurs when a task is waiting on a latch for a buffer that
2019-12-06 8:20:24 PM	Backup	BACKUPTHREAD	1	1	0	0.01	23	23	0	Occurs when a task is waiting for a backup task to finish.
2019-12-06 8:20:24 PM	Lock	LCK_M_SCH_S	13	13	0	0.01	13	13	0	Occurs when a task is waiting to acquire a Schema Share
2019-12-06 8:20:24 PM	Memory	CMEMTHREAD	0	0	0	0	10	2	8	Occurs when a task is waiting on a thread-safe memory o
2019-12-06 8:20:24 PM	Buffer Latch	PAGELATCH_UP	4	4	0	0	8	8	0	Occurs when a task is waiting on a latch for a buffer that
2019-12-06 8:20:24 PM	Buffer Latch	PAGELATCH_SH	0	0	0	0	7	2	5	Occurs when a task is waiting on a latch for a buffer that
2019-12-06 8:20:24 PM	Latch	LATCH_SH	0	0	0	0	2	2	0	Occurs when waiting for an SH (share) latch. This does ne
2019-12-06 8:20:24 PM	Latch	LATCH_EX	0	0	0	0	1	1	0	Occurs when waiting for an EX (exclusive) latch. This doe
2019-12-06 8:20:24 PM	Tracing	SQLTRACE_FILE_BUFFER	0	0	0	0	1	1	0	Occurs during synchronization on trace buffers during a f
2019-12-06 8:20:24 PM	Service Broker	BROKER_MASTERSTART	0	0	0	0	0	0	0	Occurs when a task is waiting for the primary event hand
2019-12-06 8:20:24 PM	SQL CLR	CLR_TASK_START	0	0	0	0	0	0	0	Occurs while waiting for a CLR task to complete startup.
2019-12-06 8:20:24 PM	Transaction Log IO	LOGBUFFER	0	0	0	0	0	0	0	Occurs when a task is waiting for space in the log buffer t
2019-12-06 8:20:24 PM	Lock	LCK_M_U	0	0	0	0	0	0	0	Occurs when a task is waiting to acquire an Update lock.

27

Search wait Type... | Find | Clear

Wait Details (ms/sec) | Common Waits | System wide Waits | Waits Summary (ms/sec)