

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:

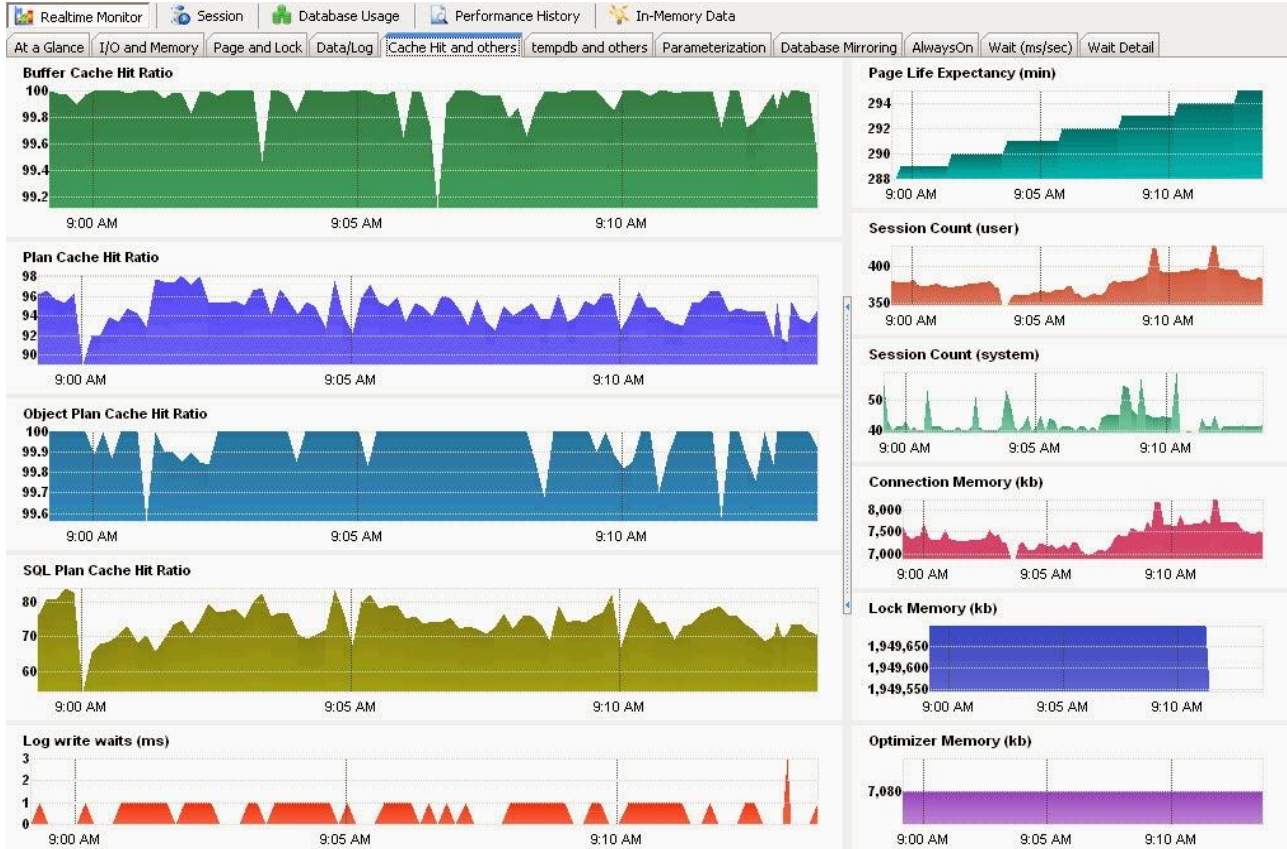
- As database administrators, we have to support different SQL Server environments.
- Often it becomes a challenge and obvious to understand the current server health status.
- To attain this goal based on my requirements, I have created this small tool just for fun with my limited development application 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.



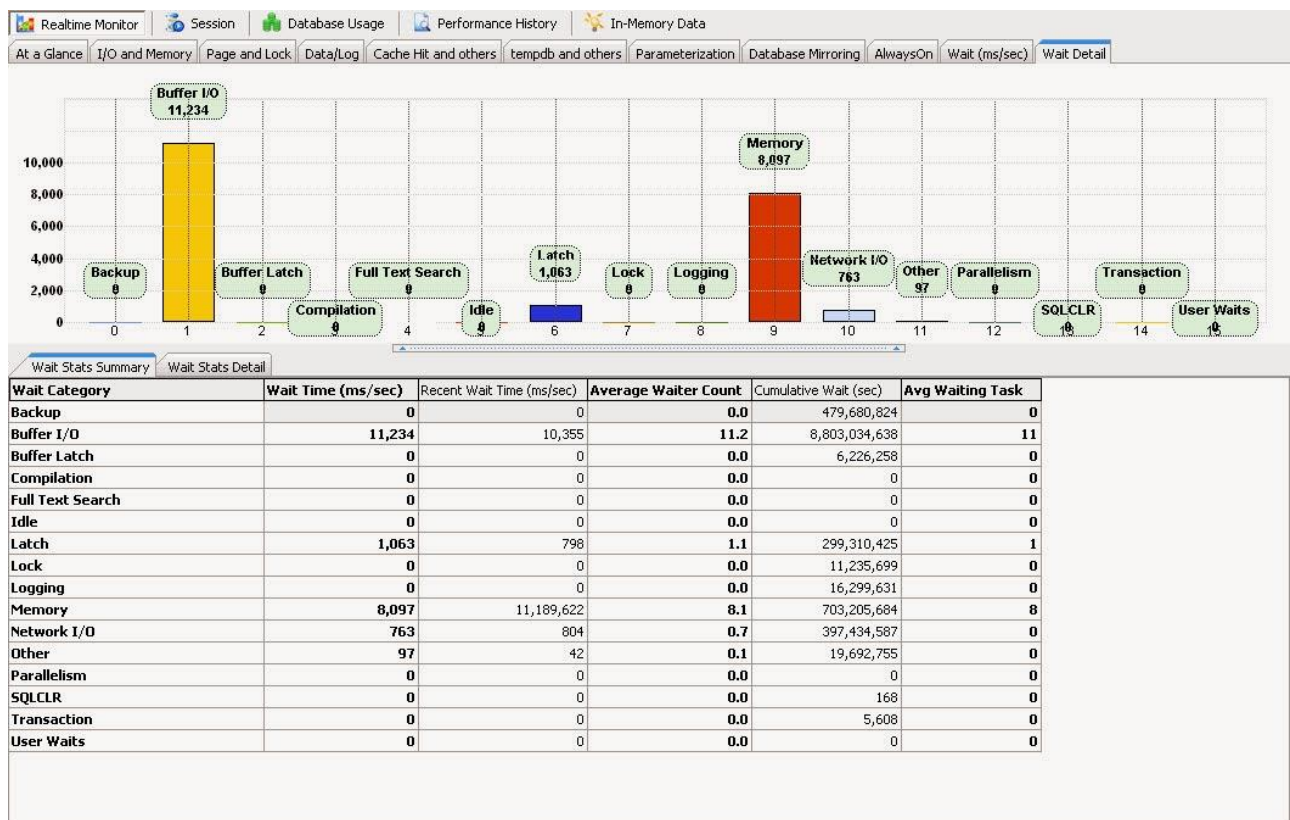
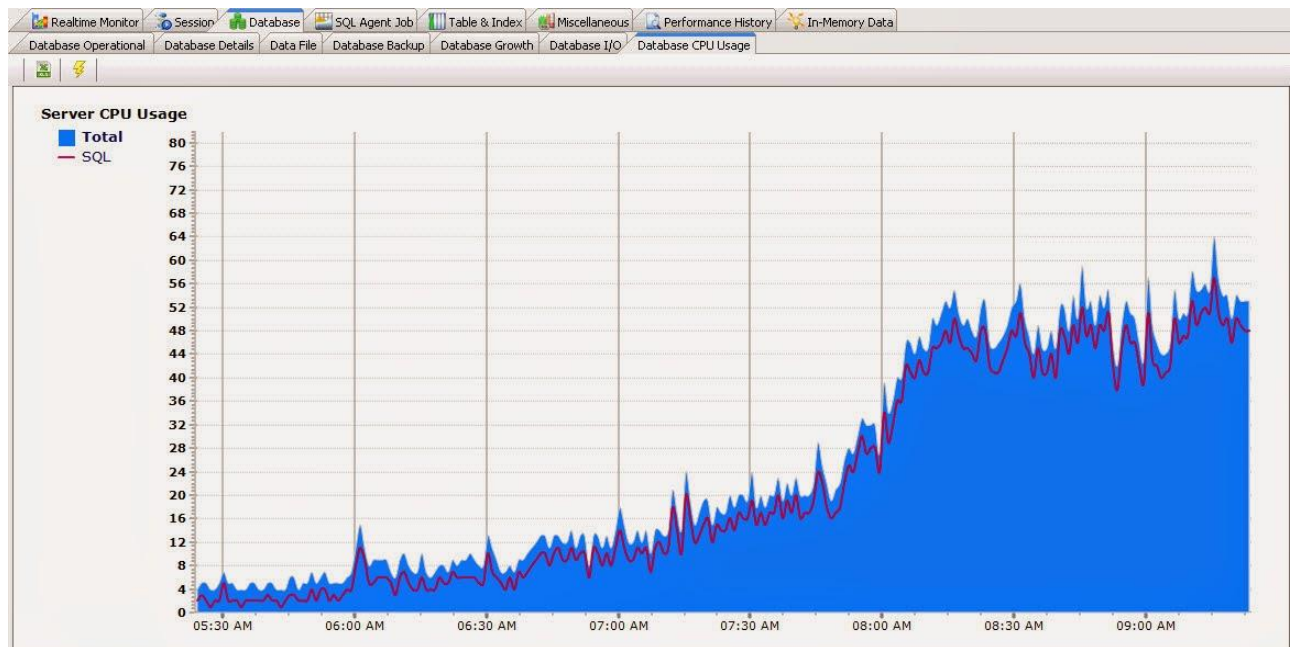
Agreement:

- 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 be 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



Performance									
Select a database: AdventureWorks2012									
Database (3K+) Tables (3K+) Missing index (3K+) Index settings & usage (8K+) Index internal utilization (8K+)									
Usage score by users									
Score	Type	Table	Index	Seek	Scan	Lookup	Stat Date	Stat Time	Index detail
1	CLUSTERED	Store	PK_Store_BusinessEntityID	0	0	0	24 March, 2013	06:14:33 PM	BusinessEntityID
1	NONCLUSTER	Store	AK_Store_rowguid	0	0	0	24 March, 2013	06:14:33 PM	rowguid
1	NONCLUSTER	Store	IX_Store_SalespersonID	0	0	0	24 March, 2013	06:14:33 PM	SalespersonID
1	CLUSTERED	ProductPhoto	PK_ProductPhoto_ProductPhotoID	0	0	0	24 March, 2013	06:14:28 PM	ProductPhotoID
1	HEAP	ProductProductPhoto	---	0	0	0	30 December, 1899	12:00:00 AM	---
1	NONCLUSTER	ProductProductPhoto	PK_ProductPhoto_ProductID_ProductPhotoID	0	0	0	24 March, 2013	06:14:28 PM	ProductID, ProductPhotoID
1	CLUSTERED	TransactionHistory	PK_TransactionHistory_TransactionID	0	0	0	24 March, 2013	06:14:29 PM	TransactionID
1	NONCLUSTER	TransactionHistory	IX_TransactionHistory_ProductID	0	0	0	24 March, 2013	06:14:29 PM	ProductID
1	NONCLUSTER	TransactionHistory	IX_TransactionHistory_ReferenceOrderID_ReferenceOrderLineID	0	0	0	24 March, 2013	06:14:29 PM	ReferenceOrderID, ReferenceOrderLineID
11	CLUSTERED	ProductReview	PK_ProductReview_ProductReviewID	1	0	0	24 March, 2013	06:14:28 PM	ProductReviewID
1	NONCLUSTER	ProductReview	IX_ProductReview_ProductID_Name	0	0	0	24 March, 2013	06:14:28 PM	ProductID, ReviewerName
1	HEAP	Tempdb	---	0	0	0	30 December, 1899	12:00:00 AM	---
1	CLUSTERED	BusinessEntity	PK_BusinessEntity_BusinessEntityID	0	0	0	24 March, 2013	06:14:27 PM	BusinessEntityID
1	NONCLUSTER	BusinessEntity	AK_BusinessEntity_rowguid	0	0	0	24 March, 2013	06:14:27 PM	rowguid
1	CLUSTERED	TransactionHistoryArchive	PK_TransactionHistoryArchive_TransactionID	0	0	0	24 March, 2013	06:14:29 PM	TransactionID
1	NONCLUSTER	TransactionHistoryArchive	IX_TransactionHistoryArchive_ProductID	0	0	0	24 March, 2013	06:14:29 PM	ProductID
1	NONCLUSTER	TransactionHistoryArchive	IX_TransactionHistoryArchive_ReferenceOrderID_ReferenceOrderLineID	0	0	0	24 March, 2013	06:14:29 PM	ReferenceOrderID, ReferenceOrderLineID
1	CLUSTERED	ProductSubcategory	PK_ProductSubcategory_ProductSubcategoryID	0	0	0	24 March, 2013	06:14:28 PM	ProductSubcategoryID
1	NONCLUSTER	ProductSubcategory	AK_ProductSubcategory_Name	0	0	0	24 March, 2013	06:14:28 PM	Name
1	NONCLUSTER	ProductSubcategory	AK_ProductSubcategory_rowguid	0	0	0	24 March, 2013	06:14:28 PM	rowguid
1	CLUSTERED	BusinessEntityAddress	PK_BusinessEntityAddress_BusinessEntityID_AddressID_AddressTypeID	0	0	0	24 March, 2013	06:14:27 PM	BusinessEntityID, AddressID, AddressTypeID
1	NONCLUSTER	BusinessEntityAddress	AK_BusinessEntityAddress_rowguid	0	0	0	24 March, 2013	06:14:27 PM	rowguid
1	NONCLUSTER	BusinessEntityAddress	IX_BusinessEntityAddress_AddressID	0	0	0	24 March, 2013	06:14:27 PM	AddressID
1	NONCLUSTER	BusinessEntityAddress	IX_BusinessEntityAddress_AddressTypeID	0	0	0	24 March, 2013	06:14:27 PM	AddressTypeID
1	CLUSTERED	ProductVendor	PK_ProductVendor_ProductID_BusinessEntityID	0	0	0	24 March, 2013	06:14:29 PM	ProductID, BusinessEntityID
1	NONCLUSTER	ProductVendor	IX_ProductVendor_UnitMeasureCode	0	0	0	24 March, 2013	06:14:29 PM	UnitMeasureCode
1	NONCLUSTER	ProductVendor	IX_ProductVendor_BusinessEntityID	0	0	0	24 March, 2013	06:14:29 PM	BusinessEntityID
1	CLUSTERED	BusinessEntityContact	PK_BusinessEntityContact_BusinessEntityID_PersonID_ContactTypeID	0	0	0	24 March, 2013	06:14:27 PM	BusinessEntityID, PersonID, ContactTypeID
1	NONCLUSTER	BusinessEntityContact	AK_BusinessEntityContact_rowguid	0	0	0	24 March, 2013	06:14:27 PM	rowguid
1	NONCLUSTER	BusinessEntityContact	IX_BusinessEntityContact_PersonID	0	0	0	24 March, 2013	06:14:27 PM	PersonID
1	NONCLUSTER	BusinessEntityContact	IX_BusinessEntityContact_ContactTypeID	0	0	0	24 March, 2013	06:14:27 PM	ContactTypeID
1	CLUSTERED	UnitMeasure	PK_UnitMeasure_UnitMeasureCode	0	0	0	24 March, 2013	06:14:29 PM	UnitMeasureCode
1	NONCLUSTER	UnitMeasure	AK_UnitMeasure_Name	0	0	0	24 March, 2013	06:14:29 PM	Name
1	CLUSTERED	Vendor	PK_Vendor_BusinessEntityID	0	0	0	24 March, 2013	06:14:29 PM	BusinessEntityID
1	NONCLUSTER	Vendor	AK_Vendor_AccountNumber	0	0	0	24 March, 2013	06:14:30 PM	AccountNumber
1	CLUSTERED	ContactType	PK_ContactType_ContactTypeID	0	0	0	24 March, 2013	06:14:27 PM	ContactTypeID
1	NONCLUSTER	ContactType	AK_ContactType_Name	0	0	0	24 March, 2013	06:14:27 PM	Name
1	CLUSTERED	CountryRegionCurrency	PK_CountryRegionCurrency_CountryRegionCode_CurrencyCode	0	0	0	24 March, 2013	06:14:30 PM	CountryRegionCode, CurrencyCode
1	NONCLUSTER	CountryRegionCurrency	IX_CountryRegionCurrency_CurrencyCode	0	0	0	24 March, 2013	06:14:30 PM	CurrencyCode
1	CLUSTERED	CountryRegion	PK_CountryRegion_CountryRegionCode	0	0	0	24 March, 2013	06:14:27 PM	CountryRegionCode
1	NONCLUSTER	CountryRegion	AK_CountryRegion_Name	0	0	0	24 March, 2013	06:14:27 PM	Name



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.

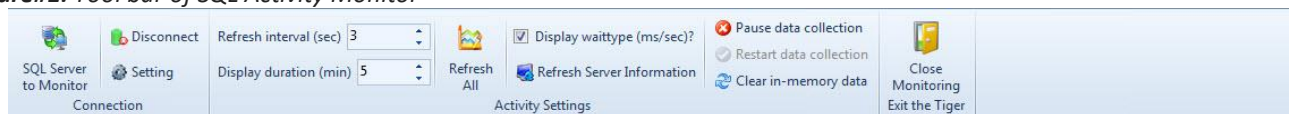
How to run:

- (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:

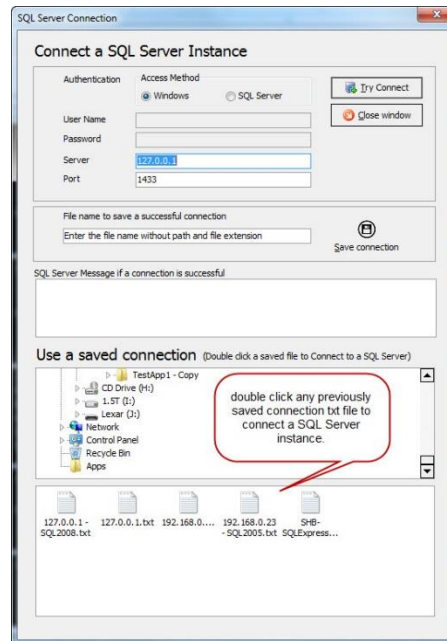
- To connect a SQL Server instance, click the “**SQL Server to Monitor**” button.
- Supply the required information and then click “**Try Connect**” in the connection dialog box.
- Once connected, close the connection dialog box or choose another server to connect to.
- All charts will be populated for 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

At a Glance | I/O and Memory | Database, Page and Lock | Cache Hit Ratio and others | Database Mirroring | AlwaysON | **Sessions** | Active Data | History

Active Sessions | All Sessions

Sessions								Database, W		
Session score	Last request start	Session	Blocked	% Compl	Wait type	Command	Wait time	Elapsed time	CPU time	
4,982	5/14/2013 4:00:00 AM	52	0	0.00	WAITFOR	WAITFOR	12:32:34:614	12:32:34:896	00:00:00:046	
2,414,748	5/14/2013 4:37:34 PM	63	0	0.00		SELECT	00:00:00:000	00:00:01:041	00:00:00:968	

Batch Text

```
CREATE PROCEDURE [dbo].[sp_trace]
    @RunTime varchar(15)='00:01:00',
    @bigduration bigint=2000,
    @dbname sysname = NULL,
    @PathFolder varchar(500)='C:\'
AS
```

Running Statement

```
WAITFOR DELAY @RunTime

/* Stop the trace
*****
```

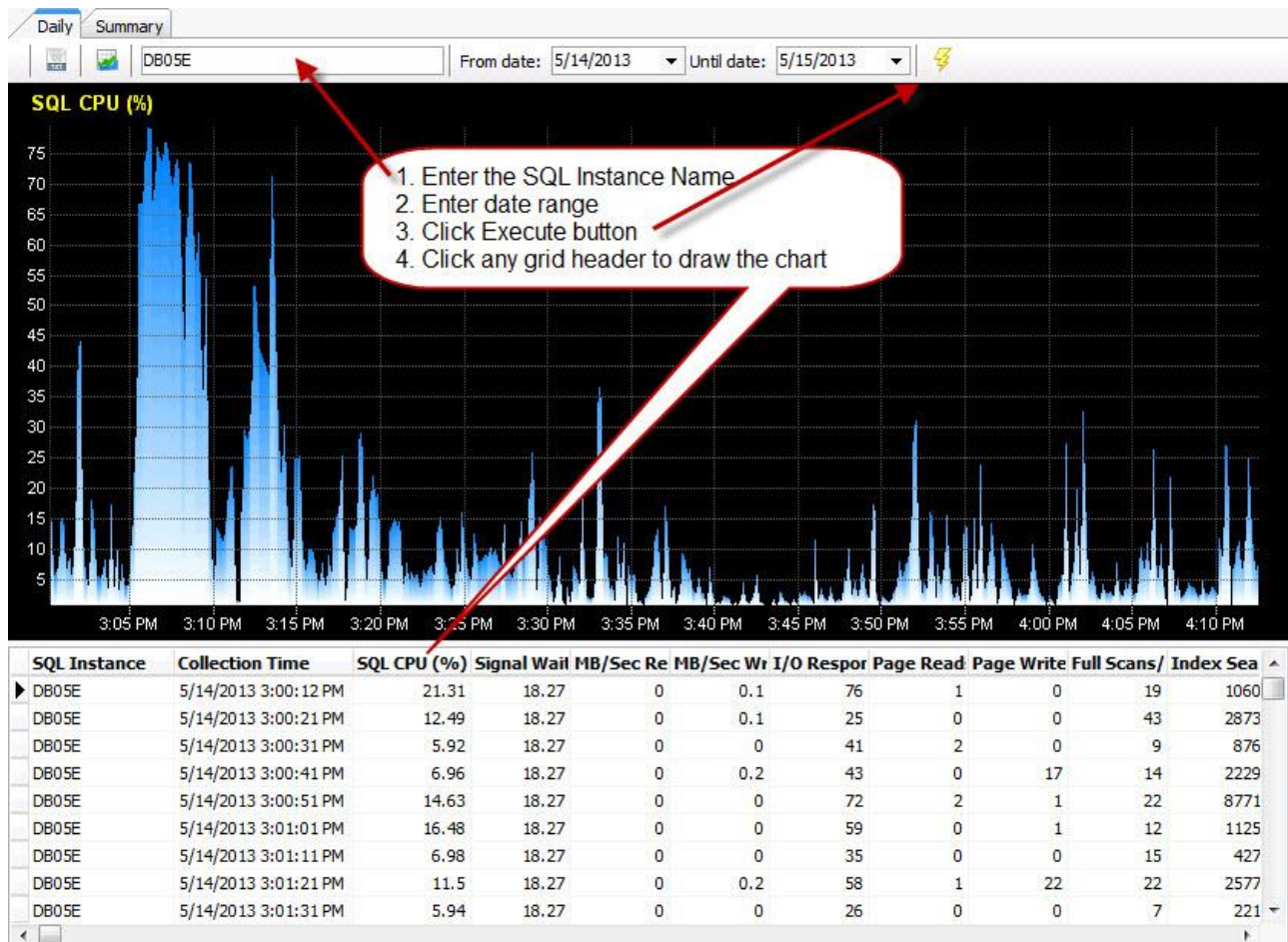

Figure#3B: Viewing all sessions

<div> <div>At a Glance</div> <div>I/O and Memory</div> <div>Database, Page and Lock</div> <div>Cache Hit Ratio and others</div> <div>Database Mirroring</div> <div>AlwaysON</div> <div>Sessions</div> <div>Active Data</div> <div>History</div> </div>									
<div> <div>Active Sessions</div> <div>All Sessions</div> </div>									
Sessions									
Session	Process ty	Database	Task state	Command	Wait time	Wait type	Blocke	Login	Login tin
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_EVENTHANDL		sa	4/2/201
14	System	master	SUSPENDED	BRKR TASK	407d 00:01:35:968	BROKER_TRANSMITTE		sa	4/2/201
15	System	master	SUSPENDED	BRKR TASK	407d 00:01:35:932	BROKER_TRANSMITTE		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

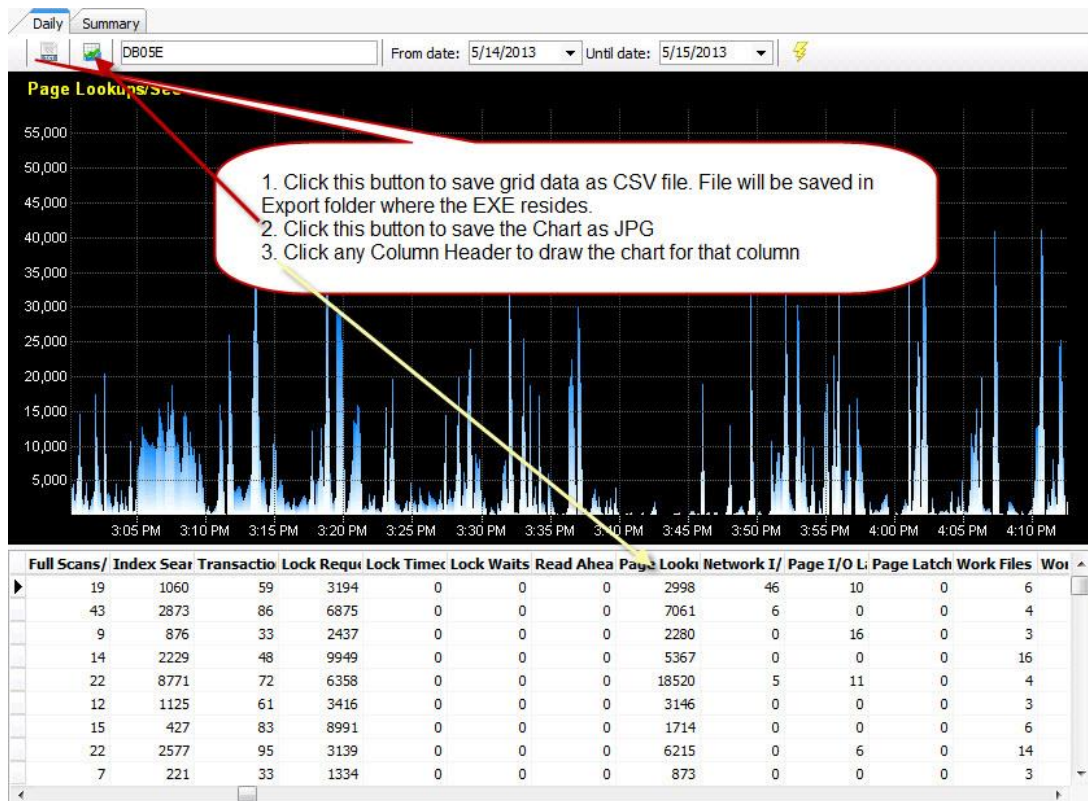
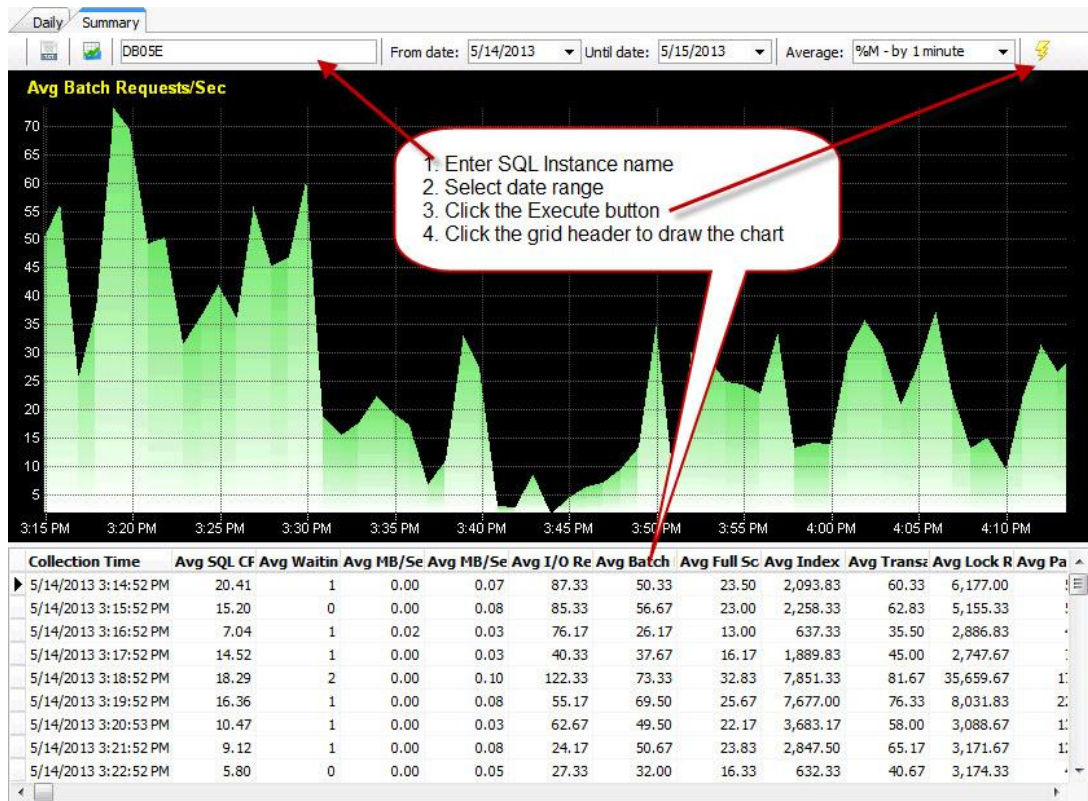
Historical data:

- 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



Figure#5: Summarizing historical data



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.**

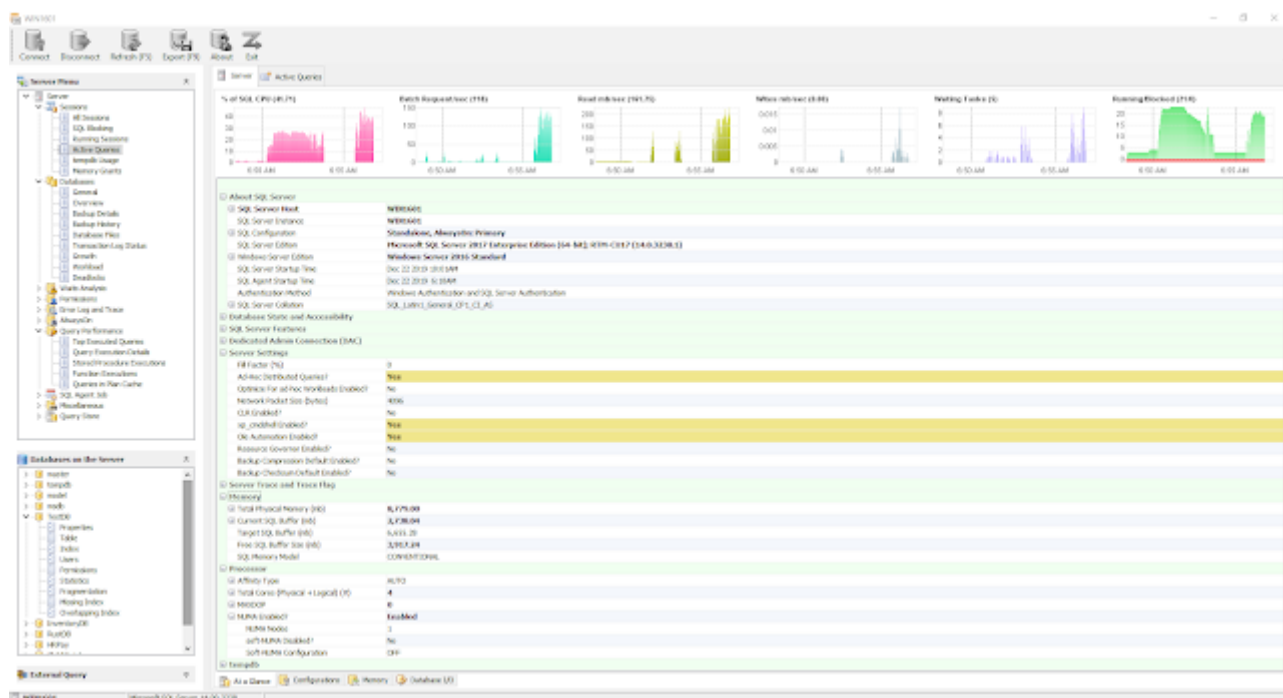
Background for the new development:

- 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>

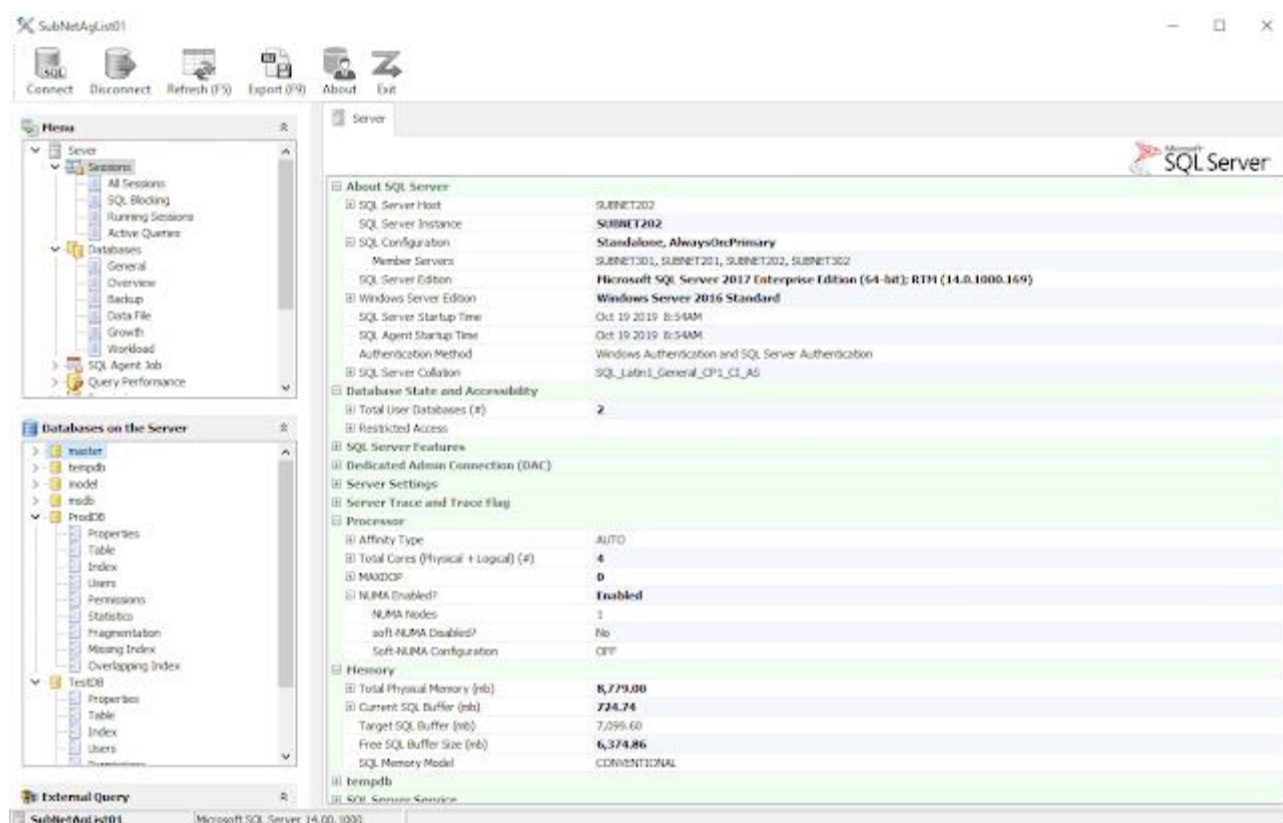
Development and distribution:

- **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

ID	Database	Recovery	Collation	Date Created	Compatibility	Database (mb)	Log (mb)	Data (mb)	Free Data (mb)	Used Data Size (b)	Log (mb)	Free Log (mb)
1	master	Simple	SQL_Latin1_General_CI_AS	2017-04-09 9:12:36 AM	100	8,556	1,131	6,258	1,131	6,258	1,131	6,258
2	tempdb	Simple	SQL_Latin1_General_CI_AS	2017-04-09 9:12:36 AM	100	22,899	1,481	21,418	1,481	21,418	1,481	21,418
3	model	Simple	SQL_Latin1_General_CI_AS	2017-04-09 9:12:36 AM	100	16,899	1,481	15,418	1,481	15,418	1,481	15,418
4	msdb	Simple	SQL_Latin1_General_CI_AS	2017-04-09 9:12:36 AM	100	46,833	1,481	45,352	1,481	45,352	1,481	45,352
5	TestDB	Simple	SQL_Latin1_General_CI_AS	2017-04-09 9:12:36 AM	100	3,903,545	1,481	3,902,064	1,481	3,902,064	1,481	3,902,064
6	AdventureWorks2017	Simple	SQL_Latin1_General_CI_AS	2017-04-09 9:12:36 AM	100	276,800	1,481	275,319	1,481	275,319	1,481	275,319
7	AdventureWorks2017	Simple	SQL_Latin1_General_CI_AS	2017-04-09 9:12:36 AM	100	2,216,800	1,481	2,215,319	1,481	2,215,319	1,481	2,215,319
8	AdventureWorks2017	Simple	SQL_Latin1_General_CI_AS	2017-04-09 9:12:36 AM	100	16,899	1,481	15,418	1,481	15,418	1,481	15,418
9	AdventureWorks2017	Simple	SQL_Latin1_General_CI_AS	2017-04-09 9:12:36 AM	100	16,899	1,481	15,418	1,481	15,418	1,481	15,418
10	AdventureWorks2017	Simple	SQL_Latin1_General_CI_AS	2017-04-09 9:12:36 AM	100	16,899	1,481	15,418	1,481	15,418	1,481	15,418
11	AdventureWorks2017	Simple	SQL_Latin1_General_CI_AS	2017-04-09 9:12:36 AM	100	16,899	1,481	15,418	1,481	15,418	1,481	15,418
12	AdventureWorks2017	Simple	SQL_Latin1_General_CI_AS	2017-04-09 9:12:36 AM	100	16,899	1,481	15,418	1,481	15,418	1,481	15,418
13	AdventureWorks2017	Simple	SQL_Latin1_General_CI_AS	2017-04-09 9:12:36 AM	100	16,899	1,481	15,418	1,481	15,418	1,481	15,418

Screenshot #4: AlwaysOn Availability Group Health status

Cluster Name	Cluster Members	Availability Group Name	SQL Instance (Connected)	AG Listener Name	Availability Group Role	Failover Mode	Backup Preference	Additional AG Features
WIN1601-Cluster01	WIN1601 (Primary), WIN1602 (Secondary)	ShareAG	WIN1601	None	Primary	Automatic Failover	Prefer Secondary	Not Synchronizing - Indicates that the database is not ready to synchronize its transaction log with the corresponding secondary databases.

Replica Role	Synchronization Health	Replica SQL Instance	Availability Mode	Failover Mode	Primary Connections	Secondary Connections	Connection State	Seeding Mode	Last Error Time	Last Error Description
Primary	Not Healthy	WIN1601	Synchronous Commit	Automatic Failover	All Connections	Read Only	Connected	Automatic		
Secondary	Not Healthy	WIN1602	Synchronous Commit	Automatic Failover	All Connections	Read Only	Connected	Automatic		

Replica SQL Instance	Replica Role	Availability Mode	Synchronization State	Synchronization Health	Ends Lag (sec)	Sync Lag (sec)	Last Commit	Last Commit (sec)	Suspend Reason	Last Sent	Last Sent (sec)	Last
WIN1601 (COPIES-5)	Primary	Synchronous Commit	Not Synchronizing	Not Healthy			2019-12-06 2:58:59 PM	1,499	User action - A user			
WIN1602 (COPIES-5)	Secondary	Synchronous Commit	Synchronized	Healthy			2019-12-06 2:58:59 PM	1,499				
WIN1603 (COPIES-5)	Secondary	Synchronous Commit	Synchronized	Healthy			2019-12-06 2:58:59 PM	1,499				
WIN1604 (COPIES-5)	Secondary	Synchronous Commit	Synchronized	Healthy			2019-12-06 2:58:59 PM	1,499				
WIN1605 (COPIES-5)	Secondary	Synchronous Commit	Synchronized	Healthy			2019-12-06 2:58:59 PM	1,499				
WIN1606 (COPIES-5)	Secondary	Synchronous Commit	Synchronized	Healthy			2019-12-06 2:58:59 PM	1,499				
WIN1607 (COPIES-5)	Secondary	Synchronous Commit	Synchronized	Healthy			2019-12-06 2:58:59 PM	1,499				
WIN1608 (COPIES-5)	Secondary	Synchronous Commit	Synchronized	Healthy			2019-12-06 2:58:59 PM	1,499				
WIN1609 (COPIES-5)	Secondary	Synchronous Commit	Synchronized	Healthy			2019-12-06 2:58:59 PM	1,499				
WIN1610 (COPIES-5)	Secondary	Synchronous Commit	Synchronized	Healthy			2019-12-06 2:58:59 PM	1,499				

Screenshot #5: Details view of blocking

SQL Blocking

ID	Blocking Task	Status	Blocking Duration	Database Name	Blocked Time	Wait Type	Column	Login Name	Client Name	Session Usage	Open Cn.	Curr. Sess.	Percent	Logins/sec	Virtual ID	CPU Time
11	Start of blocking	blocking	00:00:00.000	TechDev	04:25:28.446	PENGLAMON/KN	AJAX/NTG/CORP/MO	SERPANT/Super	SERPAN	3	0	204	0.00	28.36	0.04	00:00:00.000
75	Blocked and blocking	suspended	00:00:00.000	TechDev	04:25:28.446	LCK_M_S	NULL	SERPANT/Super	SERPAN	0	1	0	0.00	0.00	0.00	00:00:00.000
76	Blocked process	suspended	00:00:00.000	TechDev	04:25:28.446	LCK_M_S	NULL	SERPANT/Super	SERPAN	0	1	0	0.00	0.00	0.00	00:00:00.000
81	Blocked process	suspended	00:00:00.000	TechDev	04:25:28.446	LCK_M_S	NULL	SERPANT/Super	SERPAN	0	1	0	0.00	0.00	0.00	00:00:00.000
82	Blocked and blocking	suspended	00:00:00.000	TechDev	04:25:28.446	LCK_M_S	NULL	SERPANT/Super	SERPAN	0	1	0	0.00	0.00	0.00	00:00:00.000
83	Blocked process	suspended	00:00:00.000	TechDev	04:25:28.446	LCK_M_S	NULL	SERPANT/Super	SERPAN	0	1	0	0.00	0.00	0.00	00:00:00.000
84	Blocked process	suspended	00:00:00.000	TechDev	04:25:28.446	LCK_M_S	NULL	SERPANT/Super	SERPAN	0	1	0	0.00	0.00	0.00	00:00:00.000
85	Blocked process	suspended	00:00:00.000	TechDev	04:25:28.446	LCK_M_S	NULL	SERPANT/Super	SERPAN	0	1	0	0.00	0.00	0.00	00:00:00.000
86	Blocked process	suspended	00:00:00.000	TechDev	04:25:28.446	LCK_M_S	NULL	SERPANT/Super	SERPAN	0	1	0	0.00	0.00	0.00	00:00:00.000
87	Blocked process	suspended	00:00:00.000	TechDev	04:25:28.446	LCK_M_S	NULL	SERPANT/Super	SERPAN	0	1	0	0.00	0.00	0.00	00:00:00.000
88	Blocked process	suspended	00:00:00.000	TechDev	04:25:28.446	LCK_M_S	NULL	SERPANT/Super	SERPAN	0	1	0	0.00	0.00	0.00	00:00:00.000
89	Blocked process	suspended	00:00:00.000	TechDev	04:25:28.446	LCK_M_S	NULL	SERPANT/Super	SERPAN	0	1	0	0.00	0.00	0.00	00:00:00.000
90	Blocked process	suspended	00:00:00.000	TechDev	04:25:28.446	LCK_M_S	NULL	SERPANT/Super	SERPAN	0	1	0	0.00	0.00	0.00	00:00:00.000

SQL Server Enterprise Edition (64-bit) - Microsoft SQL Server 2008 R2

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

SQ Blocks		Blocking Details		Resource Usage													
Session ID	Blocking Type	Status	Blocking Duration	Database Name	Wait Type	Command	Login Name	Client Name	Open T...	Open ...	Read Con...	% CPU ...	Logical R...	Writes (MB)	CPU Time	Blocks (MB)	Rows
57	Dead-end blocking	Gorping	00:00:00.000	TstDB	HINGULARION	AWAITING COMMAND	MSPAIN/Sageon	MSPAIN	2	0	179	0.00	11.74	0.83	00:00:00.062	2.77	
24	Blocked end blocking	suspended	00:00:51.885	TstDB	LCK_PL_S	SELECT	MSPAIN/Sageon	MSPAIN	0	1	0	0.00	0.64	0.00	00:00:00.027	1.13	
75	Blocked process	suspended	00:00:01.850	TstDB	LCK_PL_S	SELECT	MSPAIN/Sageon	MSPAIN	0	1	0	0.00	0.58	0.00	00:00:00.004	0.00	
76	Blocked process	suspended	00:00:29.363	TstDB	LCK_PL_S	SELECT	MSPAIN/Sageon	MSPAIN	0	1	0	0.00	0.68	0.00	00:00:00.004	0.00	
78	Blocked process	suspended	00:00:13.752	TstDB	LCK_PL_S	SELECT	MSPAIN/Sageon	MSPAIN	0	1	0	0.00	0.58	0.00	00:00:00.003	0.00	
77	Blocked process	suspended	00:00:06.123	TstDB	LCK_PL_S	SELECT	MSPAIN/Sageon	MSPAIN	2	1	0	0.00	0.49	0.00	00:00:00.001	1.14	
79	Blocked process	suspended	00:00:00.762	TstDB	LCK_PL_S	SELECT	MSPAIN/Sageon	MSPAIN	2	1	0	0.00	0.20	0.00	00:00:00.000	0.00	
52	Pending processing	suspended	00:00:23.914	TstTstDB	CHAINED	SELECT	MSPAIN/Sageon	MSPAIN	20	1	0	0.00	0.06	0.00	00:00:00.173	0.00	
59	Pending processing	running	00:00:00.075	TstTstDB	CHAINED	SELECT	MSPAIN/Sageon	MSPAIN	2	1	0	0.00	0.03	0.00	00:00:00.000	0.00	
51	Parallel processing and blocking	suspended	00:00:16.297	TstTstDB	CHAINED	SELECT	MSPAIN/Sageon	MSPAIN	0	1	0	0.00	0.00	0.00	00:00:00.007	0.00	
68	Blocked process	suspended	00:00:13.313	TstTstDB	LCK_PL_S	SELECT	MSPAIN/Sageon	MSPAIN	0	1	1	0.00	0.07	0.00	00:00:00.002	0.00	

SQ Text

SQ Statement

```
SELECT * FROM (select(10000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
```


Screenshot #6: Server configurations.

[illegible]

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

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	3,456	3,456	0	35.3	76,393	76,393	2	Occurs when a task is waiting to acquire a Shared lock.
2019-12-06 8:20:24 PM	Buffer IO	PAGELATCH_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	BACKUPFO	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 when waiting for a log flush to complete. Common
2019-12-06 8:20:24 PM	Buffer IO	PAGELATCH_EX	26	26	0	6.52	14,135	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 behin
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. The
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	0.36	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 Modifi
2019-12-06 8:20:24 PM	Buffer IO	PAGELATCH_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	BACKUPLATCH	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 no
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	SQTRACE_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

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