

Product	Version	Theme	Environment	Date	Department	Person	Language
SEE Electrical PLM	V4Rx	Database server	Windows		After Sales	Technical Support	EN

Subject:

SQL servers recommended configuration

Description:

This document describes the recommendations for the configuration of the Microsoft SQL server. Options list is valid for Microsoft SQL Server versions 2008 to 2014.

Our proposed solution:

1 - Server level options

To obtain your general server configuration options, execute: `"SELECT * FROM sys.configurations ORDER BY name"`

The recommended values are provided in the form of the system view above.

The option's description is taken from Books Online.

1.1 - Access check cache bucket count

Recommended value: 0

The default value of 0 indicates that SQL Server is managing these options. Microsoft recommends only changing these options when directed by Microsoft Customer Support Services.

1.2 - Access check cache quota

Recommended value: 0

The default values of 0 indicates that SQL Server is managing these options. Microsoft recommends only changing these options when directed by Microsoft Customer Support Services.

1.3 - Affinity I/O mask

Recommended value: 0

The affinity I/O mask option binds SQL Server disk I/O to a specified subset of CPUs. In high-end SQL Server online transactional processing (OLTP) environments, this extension can enhance the performance of SQL Server threads issuing I/Os.

1.4 - Affinity mask

Recommended value: 0

To carry out multitasking, Microsoft Windows 2000 and Windows Server 2003 sometimes move process threads among different processors. Although efficient from an operating system point of view, this activity can reduce SQL Server performance under heavy system loads, as each processor cache is repeatedly reloaded with data. Assigning processors to specific threads can improve performance under these conditions by eliminating processor reloads and reducing thread migration across processors (thereby reducing context switching); such an association between a thread and a processor is called processor affinity.

1.5 - Affinity64 I/O mask

Recommended value: 0

The affinity64 I/O mask binds SQL Server disk I/O to a specified subset of CPUs, similar to the affinity I/O mask option. Use affinity I/O mask to bind the first 32 processors, and use affinity64 I/O mask to bind the remaining processors on the computer

1.6 - Affinity64 mask

Recommended value: 0

The affinity64 mask binds processors to specific threads, similar to the affinity mask option. Use affinity mask to bind the first 32 processors, and use affinity64 mask to bind the remaining processors on the computer. This option is only visible on the 64-bit version of SQL Server.

1.7 - Backup compression default

Recommended value: 1

Because a compressed backup is smaller than an uncompressed backup of the same data, compressing a backup typically requires less device I/O and therefore usually increases backup speed significantly. By default, compression significantly increases CPU usage, and the additional CPU consumed by the compression process might adversely impact concurrent operations.

1.8 - Cost threshold for parallelism

Recommended value: 30

Use the cost threshold for parallelism option to specify the threshold at which Microsoft SQL Server creates and runs parallel plans for queries. SQL Server creates and runs a parallel plan for a query only when the estimated cost to run a serial plan for the same query is higher than the value set in cost threshold for parallelism. The cost refers to an estimated elapsed time in seconds required to run the

serial plan on a specific hardware configuration?

1.9 - Disallow results from triggers

Recommended value: 0

Use the disallow results from triggers option to control whether triggers return result sets. Triggers that return result sets may cause unexpected behavior in applications that are not designed to work with them.

1.10 - Index create memory (KB)

Recommended value: 0

In SQL Server, the index create memory option controls the maximum amount of memory initially allocated for creating indexes. If more memory is later needed for index creation, and the memory is available, the server will use it, thus exceeding the setting of this option. If additional memory is not available, the index creation will continue using the memory already allocated.

1.11 - Lightweight pooling

Recommended value: 0

Use the lightweight pooling option to provide a means of reducing the system overhead associated with the excessive context switching sometimes seen in symmetric multiprocessing (SMP) environments. When excessive context switching is present, lightweight pooling can provide better throughput by performing the context switching inline, thus helping to reduce user/kernel ring transitions.

1.12 - Locks

Recommended value: 0

Use the locks option to set the maximum number of available locks, thereby limiting the amount of memory the Database Engine uses for them. The default setting is 0, which allows the Database Engine to allocate and deallocate lock structures dynamically, based on changing system requirements.

1.13 - Max degree of parallelism

Recommended value: 1

The max degree of parallelism option determines the maximum number of processors to use in parallel plan execution

1.14 - Max server memory (MB)

Recommended value:???

The min server memory and max server memory configuration options establish upper and lower limits to the amount of memory used by

the buffer pool of the Microsoft SQL Server Database Engine.

There are several calculation algorithm in the web, giving similar results.

Here's simplified one by Jonathan Kehayias (<https://www.sqlskills.com/blogs/jonathan/how-much-memory-does-my-sql-server-actually-need/>):

Reserve 1 GB of RAM for the OS, 1 GB for each 4 GB of RAM installed from 4–16 GB, and then 1 GB for every 8 GB RAM installed above 16 GB RAM.

Then all to SQL.

1.15 - Max worker threads

Recommended value: 0

The max worker threads option configures the number of worker threads that are available to SQL Server processes. SQL Server uses the native thread services of the operating systems so that one or more threads support each network that SQL Server supports simultaneously, another thread handles database checkpoints, and a pool of threads handles all users.

1.16 - Min memory per query (KB)

Recommended value: 1024

Use the min memory per query option to specify the minimum amount of memory (in kilobytes) that will be allocated for the execution of a query

1.17 - Min server memory (MB)

Recommended value: 0

The min server memory and max server memory configuration options establish upper and lower limits to the amount of memory used by the buffer pool of the Microsoft SQL Server Database Engine.

Some people advice to set 50% of the 'max server memory'.

1.18 - Nested triggers

Recommended value: 1

Use the nested triggers option to control whether an AFTER trigger can cascade; that is, perform an action that initiates another trigger, which initiates another trigger, and so on

1.19 - Network packet size (B)

Recommended value: 4096

Use the network packet size option to set the packet size (in bytes) used across the entire network. Packets are the fixed-size chunks of data that transfer requests and results between clients and servers

1.20 - Optimize for ad hoc workloads

Recommended value: 0

The optimize for ad hoc workloads option is used to improve the efficiency of the plan cache for workloads that contain many single use ad hoc batches

1.21 - Priority boost

Recommended value: 0

Use the priority boost option to specify whether Microsoft SQL Server should run at a higher Microsoft Windows scheduling priority than other processes on the same computer.

1.22 - Query wait (s)

Recommended value: -1

Use the query wait option to specify the time in seconds (from 0 through 2147483647) that a query waits for resources before timing out. If the default value of -1 is used, or if -1 is specified, then the time-out is calculated as 25 times of the estimated query cost.

1.23 - User connections

Recommended value: 0

Use the user connections option to specify the maximum number of simultaneous user connections allowed on Microsoft SQL Server. The actual number of user connections allowed also depends on the version of SQL Server you are using and the limits of your application or applications and hardware. SQL Server allows a maximum of 32,767 user connections.

2 - Additional server options

Options that are available by: `SELECT SERVERPROPERTY (property)`
And also Trace Flags, defined at SQL server startup.

2.1 - Collation

Recommended value: it can be any, but it must be Case Insensitive (CI)

Name of the default collation for the server.

2.2 - Edition

Recommended value: Standard, Enterprise

Installed product edition of the instance of SQL Server.

2.3 - ProductVersion

Recommended value: 2012 with latest service pack

Version of the instance of SQL Server.

2.4 - TraceFlag 2389 and 2390

Recommended value: Yes

Enhance the server performance in case of important increase in the number of records in some table.

3 - User database options

To get the general user databases configuration options, start the following SQL command:

```
"SELECT * FROM sys.databases WHERE database_id = DB_ID()".
```

This chapter describes options that could hurt database usage.

The recommended values are provided in the form of the system view above.

The option's description is taken from Books Online.

3.1 - Compatibility_level

Recommended value: highest possible for given SQL Server version

It defines certain database behavior to be compatible with a specified earlier version of Microsoft SQL Server

3.2 - Collation_name

Recommended value: it can be any, but it must be Case Insensitive (CI)

It is the default collation in the database.

3.3 - Is_auto_close_on

Recommended value: 0 (OFF)

When set to ON, the database is shut down cleanly and its resources are freed after the last user exits. The database automatically reopens when a user tries to use the database again.

When set to OFF, the database remains open after the last user exits.

3.4 - Is_auto_shrink_on

Recommended value: 0 (OFF)

When set to ON, the database files are candidates for periodic shrinking. Both data file and log files can be shrunk automatically by SQL Server. AUTO_SHRINK reduces the size of the transaction log only if the database is set to SIMPLE recovery model or if the log is backed up.

When set to OFF, database files are not automatically shrunk during periodic checks for unused space.

3.5 - Snapshot_isolation_state

Recommended value: 1 (ON)

When ON is specified, transactions can specify the SNAPSHOT transaction isolation level. When a transaction runs at the SNAPSHOT isolation level, all statements see a snapshot of data as it exists at the start of the transaction.

When OFF is specified, transactions cannot specify the SNAPSHOT transaction isolation level.

3.6 - Is_read_committed_snapshot_on

Recommended value: 0 (OFF)

When ON is specified, transactions specifying the READ COMMITTED isolation level use row versioning instead of locking. When a transaction runs at the READ COMMITTED isolation level, all statements see a snapshot of data as it exists at the start of the statement.

When OFF is specified, transactions that specify the READ COMMITTED isolation level use locking

3.7 - Page_verify_option

Recommended value: 2 (CHECKSUM)

When CHECKSUM is specified, the Database Engine calculates a checksum over the contents of the whole page and stores the value in the page header when a page is written to disk. When the page is read from disk, the checksum is recomputed and compared to the checksum value stored in the page header.

3.8 - Is_auto_create_stats_on

Recommended value: 1 (ON)

When set to ON, statistics are automatically created on columns used in a predicate.

When set to OFF, statistics are not automatically created; instead, statistics can be manually created.

3.9 - Is_auto_update_stats_on

Recommended value: 1 (ON)

When set to ON, any missing statistics required by a query for optimization are automatically built during query optimization.

When set to OFF, statistics must be manually created

3.10 - Is_auto_update_stats_async_on

Recommended value: 0 (OFF)

This option enables faster plan generation, but may create sub-optimal query plans because the plan compilation may be based on stale statistics.

3.11 - Is_recursive_triggers_on

Recommended value: 0 (OFF)

When ON is specified, recursive firing of AFTER triggers is allowed.

When OFF is specified, only direct recursive firing of AFTER triggers is not allowed.

3.12 - Is_cursor_close_on_commit_on

Recommended value: 0 (OFF)

When set to ON, any cursors open when a transaction is committed or rolled back are closed.

When set to OFF, remain open when a transaction is committed; rolling back a transaction closes any cursors except those defined as INSENSITIVE or STATIC.

3.13 - Is_local_cursor_default

Recommended value: 0 (GLOBAL)

When LOCAL is specified and a cursor is not defined as GLOBAL when created, the scope of the cursor is local to the batch, stored procedure, or trigger in which the cursor was created. The cursor name is valid only within this scope.

When GLOBAL is specified, and a cursor is not defined as LOCAL when created, the scope of the cursor is global to the connection. The cursor name can be referenced in any stored procedure or batch executed by the connection.

3.14 - Is_parameterization_forced

Recommended value: 0 (SIMPLE)

When SIMPLE specified, queries are parametrized based on the default behavior of the database

When FORCED is specified, SQL Server parametrizes all queries in the database.

3.15 - Containment

Recommended value: 0 (NONE)

A contained database is a database that is isolated from other databases and from the instance of SQL Server that hosts the database

