# [Configure on Windows](https://docs.microsoft.com/en-us/sql/database-engine/configure-windows/configure-database-engine-instances-sql-server)

[Disk striping and RAID](https://technet.microsoft.com/library/ms190764(v=sql.105).aspx)

**Place Data and Log Files on Separate Drives**

This rule checks whether data and log files are placed on separate logical drives. Placing both data AND log files on the same device can cause contention for that device, resulting in poor performance. Placing the files on separate drives allows the I/O activity to occur at the same time for both the data and log files.

**View or Change the Default Locations for Data and Log Files**

The best practice for protecting your data files and log files is to ensure that they are protected by access control lists (ACLs). Set the ACLs on the directory root under which the files are created.

**Affinity mask Server Configuration Option**

To carry out multitasking, Microsoft Windows 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.

SQL Server supports processor affinity by means of two affinity mask options: affinity mask (also known as **CPU affinity mask**) and affinity I/O mask.

Affinity support for servers with 33 to 64 processors is only available on 64-bit operating systems.

When you set **affinity mask** to 0 (the default), the Microsoft Windows 2000 or Windows Server 2003 scheduling algorithms set the thread's affinity. When you set **affinity mask** to any nonzero value, SQL Server affinity interprets the value as a bitmask that specifies those processors eligible for selection.

A one-byte **affinity mask** covers up to 8 CPUs in a multiprocessor computer.

sp\_configure 'affinity mask', 38;

RECONFIGURE;

**Server Memory Server Configuration Options**

Use the two server memory options, **min server memory** and **max server memory**, to reconfigure the amount of memory (in megabytes) that is managed by the SQL Server Memory Manager for a SQL Server process used by an instance of SQL Server.

The default setting for **min server memory** is 0, and the default setting for **max server memory** is 2,147,483,647 megabytes (MB). By default, SQL Server can change its memory requirements dynamically based on available system resources.

Setting **max server memory** value too high can cause a single instance of SQL Server might have to compete for memory with other SQL Server instances hosted on the same host. However, setting this value too low could cause significant memory pressure and performance problems. Setting **max server memory** to the minimum value can even prevent SQL Server from starting.

## Lock Pages in Memory (LPIM)

This Windows policy determines which accounts can use a process to keep data in physical memory, preventing the system from paging the data to virtual memory on disk. Locking pages in memory may keep the server responsive when paging memory to disk occurs. The **Lock Pages in Memory** option is set to ON in instances of SQL Server Standard edition and higher when the account with privileges to run sqlservr.exe has been granted the Windows Lock Pages in Memory (LPIM) user right.

To disable the **Lock Pages In Memory** option for SQL Server, remove the Lock Pages in Memory user right for the account with privileges to run sqlservr.exe (the SQL Server startup account) startup account.