

Server Memory Server Configuration Options



SQL Server 2012

This topic has not yet been rated [Rate this topic](#)

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 2147483647 MB. By default, SQL Server can change its memory requirements dynamically based on available system resources.

Note

Setting **max server memory** to the minimum value can severely reduce SQL Server performance and even prevent it from starting. If you cannot start SQL Server after changing this option, start it using the **-f** startup option and reset **max server memory** to its previous value. For more information, see [Database Engine Service Startup Options](#).

When SQL Server is using memory dynamically, it queries the system periodically to determine the amount of free memory. Maintaining this free memory prevents the operating system (OS) from paging. If less memory is free, SQL Server releases memory to the OS. If more memory is free, SQL Server may allocate more memory. SQL Server adds memory only when its workload requires more memory; a server at rest does not increase the size of its virtual address space.

SQL Server uses the memory notification API **QueryMemoryResourceNotification** to determine when the SQL Server Memory Manager may allocate memory and release memory.

Allowing SQL Server to use memory dynamically is recommended; however, you can set the memory options manually and restrict the amount of memory that SQL Server can access. Before you set the amount of memory for SQL Server, determine the appropriate memory setting by subtracting, from the total physical memory, the memory required for the OS and any other instances of SQL Server (and other system uses, if the computer is not wholly dedicated to SQL Server). This difference is the maximum amount of memory you can assign to SQL Server.

Setting the Memory Options Manually

Set **min server memory** and **max server memory** to span a range of memory values. This method is useful for system or database administrators to configure an instance of SQL Server in conjunction with the memory requirements of other applications that run on the same computer.

Use **min server memory** to guarantee a minimum amount of memory available to the SQL Server Memory Manager for an instance of SQL Server. SQL Server will not immediately allocate the amount of memory specified in **min server memory** on startup. However, after memory usage has reached this value due to client load, SQL Server cannot free memory unless the value of **min server memory** is reduced.

Note

SQL Server is not guaranteed to allocate the amount of memory specified in **min server memory**. If the load on the server never requires allocating the amount of memory specified in **min server memory**, SQL Server will run with less memory.

OS Type	Minimum Memory Amounts Allowable for max server memory
32-bit	64 MB
64-bit	128 MB

How to configure memory options using SQL Server Management Studio

Use the two server memory options, **min server memory** and **max server memory**, to reconfigure the amount of memory (in megabytes) managed by the SQL Server Memory Manager for an instance of SQL Server. By default, SQL Server can change its memory requirements dynamically based on available system resources.

Procedure for configuring a fixed amount of memory

To set a fixed amount of memory

1. In Object Explorer, right-click a server and select **Properties**.
2. Click the **Memory** node.
3. Under **Server Memory Options**, enter the amount that you want for **Minimum server memory** and **Maximum server memory**.

Use the default settings to allow SQL Server to change its memory requirements dynamically based on available system resources. The default setting for **min server memory** is 0, and the default setting for **max server memory** is 2147483647 megabytes (MB). For information on minimum amounts allowed for **max server memory**, see [Setting the Memory Options Manually](#)

Maximize Data Throughput for Network Applications

To optimize system memory use for SQL Server, you should limit the amount of memory that is used by the system for file caching. To limit the file system cache, make sure that **Maximize data throughput for file sharing** is not selected. You can specify the smallest file system cache by selecting **Minimize memory used** or **Balance**.

To check the current setting on your operating system

1. Click **Start**, click **Control Panel**, double-click **Network Connections**, and then double-click **Local Area Connection**.
2. On the **General** tab, click **Properties**, select **File and Printer Sharing Microsoft Networks**, and then click **Properties**.
3. If **Maximize data throughput for network applications** is selected, choose any other option, click **OK**, and then close the rest of the dialog boxes.

Lock Pages in Memory

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 SQL Server **Lock Pages in Memory** option is set to ON in 32-bit and 64-bit instances of SQL Server 2012 Standard edition and higher when the account with privileges to run sqlservr.exe has been granted the Windows "Locked Pages in Memory" (LPIM) user right. In earlier versions of SQL Server, setting the Lock Pages option for a 32-bit instance of SQL Server, requires that the account with privileges to run sqlservr.exe have the LPIM user right and the 'awe_enabled' configuration option is set to ON.

To disable the **Lock Pages In Memory** option for SQL Server, remove the "Locked Pages in Memory" user right for the SQL Server startup account.

To Disable Lock Pages in Memory

To disable the lock pages in memory option

1. On the **Start** menu, click **Run**. In the **Open** box, type **gpedit.msc**.

The **Group Policy** dialog box opens.

2. On the **Group Policy** console, expand **Computer Configuration**, and then expand **Windows Settings**.
3. Expand **Security Settings**, and then expand **Local Policies**.
4. Select the **User Rights Assignment** folder.

The policies will be displayed in the details pane.

5. In the pane, double-click **Lock pages in memory**.
6. In the **Local Security Policy Setting** dialog box, select the account with privileges to run sqlservr.exe and click **Remove**.

Virtual Memory Manager

The 32-bit operating systems provide access to 4 GB of virtual address space. 2 GB of virtual memory is private per process and available for application use. 2 GB is reserved for operating system use. All operating system editions include a switch that can provide applications with access up to 3 GB of virtual address space, limiting the operating system to 1 GB. For more information about how to use the switch memory configuration, see the Windows documentation about 4-gigabyte tuning (4GT). When the 32-bit SQL Server is running on 64-bit operating system its user available virtual address space is the full 4 GB.

The committed regions of address space are mapped to the available physical memory by the Windows Virtual Memory Manager (VMM).

For more information on the amount of physical memory supported by different operating systems, see the Windows documentation "Memory Limits for Windows Releases".

Virtual memory systems allow the over-commitment of physical memory, so that the ratio of virtual-to-physical memory can exceed 1:1. As a result, larger programs can run on computers with a variety of physical memory configurations. However, using significantly more virtual memory than the combined average working sets of all the processes can cause poor performance.

The **min server memory** and **max server memory** options are advanced options. If you are using the **sp_configure** system stored procedure to change these settings, you can change them only when **show advanced options** is set to 1. These settings take effect immediately without a server restart.

Running Multiple Instances of SQL Server

When you are running multiple instances of the Database Engine, there are three approaches you can use to manage memory:

- Use **max server memory** to control memory usage. Establish maximum settings for each instance, being careful that the total allowance is not more than the total physical memory on your machine. You might want to give each instance memory proportional to its expected workload or database size. This approach has the advantage that when new processes or instances start up, free memory will be available to them immediately. The drawback is that if you are not running all of the instances, none of the running instances will be able to utilize the remaining free memory.
- Use **min server memory** to control memory usage. Establish minimum settings for each instance, so that the sum of these minimums is 1-2 GB less than the total physical memory on your machine. Again, you may establish these minimums proportionately to the expected load of that instance. This approach has the advantage that if not all instances are running at the same time, the ones that are running can use the remaining free memory. This approach is also useful when there is another memory-intensive process on the computer, since it would insure that SQL Server would at least get a reasonable amount of memory. The drawback is that when a new instance (or any other process) starts, it may take some time for the running instances to release memory, especially if they must write modified pages back to their databases to do so.
- Do nothing (not recommended). The first instances presented with a workload will tend to allocate all of memory. Idle instances, or instances started later, may end up running with only a minimal amount of memory available. SQL Server makes no attempt to balance memory usage across instances. All instances will, however, respond to Windows Memory Notification signals to adjust the size of their memory footprint. Windows does not balance memory across applications with the Memory Notification API. It merely provides global feedback as to the availability of memory on the system.

You can change these settings without restarting the instances, so you can easily experiment to find the best settings for your usage pattern.

Providing the Maximum Amount of Memory to SQL Server

	32-bit	64-bit
Conventional memory	Up to process virtual address space limit in all SQL Server editions: <ul style="list-style-type: none"> • 2 GB • 3 GB with /3gb boot parameter¹ • 4 GB on WOW64² 	Up to process virtual address space limit in all SQL Server editions: <ul style="list-style-type: none"> • 8 TB on x64 architecture

¹ **/3gb** is an operating-system boot parameter. For more information, visit the [MSDN Library](#).

² WOW64 (Windows on Windows 64) is a mode in which 32-bit SQL Server runs on a 64-bit

operating system. For more information, visit the [MSDN Library](#).

Examples

The following example sets the `max server memory` option to 4 GB:

```
sp_configure 'show advanced options', 1;
GO
RECONFIGURE;
GO
sp_configure 'max server memory', 4096;
GO
RECONFIGURE;
GO
```

See Also

Reference

[RECONFIGURE \(Transact-SQL\)](#)

[sp_configure \(Transact-SQL\)](#)

Concepts

[Monitor and Tune for Performance](#)

[Server Configuration Options](#)

Did you find this helpful? Yes No

Community Content

© 2012 Microsoft. All rights reserved.