* [[http://support.microsoft.com/library/images/support/kbgraphics/public/en-us/downarrow.gif](http://support.microsoft.com/kb/907877)SUMMARY](http://support.microsoft.com/kb/907877)
* [[http://support.microsoft.com/library/images/support/kbgraphics/public/en-us/downarrow.gif](http://support.microsoft.com/kb/907877)INTRODUCTION](http://support.microsoft.com/kb/907877)
* [[http://support.microsoft.com/library/images/support/kbgraphics/public/en-us/downarrow.gif](http://support.microsoft.com/kb/907877)MORE INFORMATION](http://support.microsoft.com/kb/907877)
  + [[http://support.microsoft.com/library/images/support/kbgraphics/public/en-us/downarrow.gif](http://support.microsoft.com/kb/907877)Memory Manager](http://support.microsoft.com/kb/907877)
  + [[http://support.microsoft.com/library/images/support/kbgraphics/public/en-us/downarrow.gif](http://support.microsoft.com/kb/907877)Summary of memory usage](http://support.microsoft.com/kb/907877)
  + [[http://support.microsoft.com/library/images/support/kbgraphics/public/en-us/downarrow.gif](http://support.microsoft.com/kb/907877)Aggregate memory](http://support.microsoft.com/kb/907877)
  + [[http://support.microsoft.com/library/images/support/kbgraphics/public/en-us/downarrow.gif](http://support.microsoft.com/kb/907877)Buffer distribution](http://support.microsoft.com/kb/907877)
  + [[http://support.microsoft.com/library/images/support/kbgraphics/public/en-us/downarrow.gif](http://support.microsoft.com/kb/907877)Buffer pool details](http://support.microsoft.com/kb/907877)
  + [[http://support.microsoft.com/library/images/support/kbgraphics/public/en-us/downarrow.gif](http://support.microsoft.com/kb/907877)Procedure cache](http://support.microsoft.com/kb/907877)
  + [[http://support.microsoft.com/library/images/support/kbgraphics/public/en-us/downarrow.gif](http://support.microsoft.com/kb/907877)Global memory objects](http://support.microsoft.com/kb/907877)
  + [[http://support.microsoft.com/library/images/support/kbgraphics/public/en-us/downarrow.gif](http://support.microsoft.com/kb/907877)Query memory objects](http://support.microsoft.com/kb/907877)
  + [[http://support.microsoft.com/library/images/support/kbgraphics/public/en-us/downarrow.gif](http://support.microsoft.com/kb/907877)Optimization](http://support.microsoft.com/kb/907877)
  + [[http://support.microsoft.com/library/images/support/kbgraphics/public/en-us/downarrow.gif](http://support.microsoft.com/kb/907877)Memory brokers](http://support.microsoft.com/kb/907877)

[Expand all](javascript:void(0);) | [Collapse all](javascript:void(0);)

[SUMMARY](javascript:void(0);)

**This article discusses the output of the DBCC MEMORYSTATUS command. This command...**

*This article discusses the output of the DBCC MEMORYSTATUS command. This command is frequently used to troubleshoot Microsoft SQL Server memory consumption issues.  
  
This article describes the elements of the output for Memory Manager, for the summary of memory usage, for the aggregate memory information, for the buffer distribution information, for the buffer pool information, and for the procedure cache information. It also describes the output about global memory objects, about query memory objects, about optimization, and about memory brokers.*

[[http://support.microsoft.com/library/images/support/kbgraphics/public/en-us/uparrow.gif](http://support.microsoft.com/kb/907877#top)Back to the top](http://support.microsoft.com/kb/907877#top)

[INTRODUCTION](javascript:void(0);)

**The DBCC MEMORYSTATUS command provides a snapshot of the current memory status o...**

The DBCC MEMORYSTATUS command provides a snapshot of the current memory status of Microsoft SQL Server. You can use the output from this command to troubleshoot memory consumption issues in SQL Server or to troubleshoot specific out-of-memory errors. (Many out-of-memory errors automatically print this output in the error log.) Microsoft Customer Support Services may also request that you run this command during a specific support incident if you are experiencing an error that may be associated with a low-memory condition.  
  
**Note** Performance Monitor (PerfMon) and Task Manager do not account for memory correctly if Address Windowing Extentions (AWE) support is enabled.  
  
This article describes some of the data that you can obtain from the output of the DBCC MEMORYSTATUS command. Several sections of this article include proprietary implementation details that are not explained here. Microsoft Customer Support Services will not answer any questions or provide more information about the meaning of specific counters beyond the information that is supplied in this article.

[[http://support.microsoft.com/library/images/support/kbgraphics/public/en-us/uparrow.gif](http://support.microsoft.com/kb/907877#top)Back to the top](http://support.microsoft.com/kb/907877#top)

[MORE INFORMATION](javascript:void(0);)

**Important The DBCC MEMORYSTATUS command is intended to be a diagnostic tool for...**

**Important** The DBCC MEMORYSTATUS command is intended to be a diagnostic tool for Microsoft Customer Support Services. The format of the output and the level of detail that is provided are subject to change between service packs and product releases. The functionality that the DBCC MEMORYSTATUS command provides may be replaced by a different mechanism in later product versions. Therefore, in later product versions, this command may no longer function. No additional warnings will be made before this command is changed or removed. Therefore, applications that use this command may break without warning.  
  
The output of the DBCC MEMORYSTATUS command has changed from earlier releases of SQL Server. The output now contains several sections that were unavailable in earlier product versions.

[[http://support.microsoft.com/library/images/support/kbgraphics/public/en-us/uparrow.gif](http://support.microsoft.com/kb/907877#top)Back to the top](http://support.microsoft.com/kb/907877#top)

**Memory Manager**

The first section of the output is Memory Manager. This section shows overall memory consumption by SQL Server.

Memory Manager KB

------------------------------ --------------------

VM Reserved 1761400

VM Committed 1663556

AWE Allocated 0

Reserved Memory 1024

Reserved Memory In Use 0

(5 row(s) affected)

The elements in this section are the following:

* **VM Reserved**: This value shows the overall amount of virtual address space (VAS) that SQL Server has reserved.
* **VM Committed**: This value shows the overall amount of VAS that SQL Server has committed. VAS that is committed has been associated with physical memory.
* **AWE Allocated**: This value shows the overall amount of memory that is allocated through the AWE mechanism on the 32-bit version of SQL Server. Or, this value shows the overall amount of memory that locked pages consume on the 64-bit version of the product.
* **Reserved Memory**: This value shows the memory that is reserved for the dedicated administrator connection (DAC).
* **Reserved Memory In Use**: This value shows the reserved memory that is being used.

[[http://support.microsoft.com/library/images/support/kbgraphics/public/en-us/uparrow.gif](http://support.microsoft.com/kb/907877#top)Back to the top](http://support.microsoft.com/kb/907877#top)

**Summary of memory usage**

The Memory Manager section is followed by a summary of memory usage for each memory node. In a Non-uniform memory access (NUMA) enabled system, there will be a corresponding Memory node entry for each hardware NUMA node. In an SMP system, there will be a single Memory node entry.   
  
**Note** The memory node ID may not correspond to the hardware node ID.

Memory node Id = 0 KB

------------------------------ --------------------

VM Reserved 1757304

VM Committed 1659612

AWE Allocated 0

MultiPage Allocator 10760

SinglePage Allocator 73832

(5 row(s) affected)

**Note** These values show the memory that is allocated by threads that are running on this NUMA node. These values are not the memory that is local to the NUMA node.  
  
The elements in this section are the following:

* **VM Reserved**: This value shows the VAS that is reserved by threads that are running on this node.
* **VM Committed**: This value shows the VAS that is committed by threads that are running on this node.
* **AWE Allocated**: This value shows the memory that is allocated through the AWE mechanism on the 32-bit version of the product. Or, this value shows the overall amount of memory that is consumed by locked pages on the 64-bit version of the product.   
    
  In a NUMA-enabled system, this value can be incorrect or negative. However, the overall **AWE Allocated** value in the Memory Manager section is a correct value. To track memory that is allocated by individual NUMA nodes, use **SQL Server: Buffer Node** performance objects. (For more information, see *SQL Server Books Online*.)
* **MultiPage Allocator**: This value shows the memory that is allocated through the multipage allocator by threads that are running on this node. This memory comes from outside the buffer pool.
* **SinglePage Allocator**: This value shows the memory that is allocated through the single-page allocator by threads that are running on this node. This memory is stolen from the buffer pool.

**Note** The sums of the **VM Reserved** values and the **VM Committed** values on all memory nodes will be slightly less than the corresponding values that are reported in the Memory Manager section.

[[http://support.microsoft.com/library/images/support/kbgraphics/public/en-us/uparrow.gif](http://support.microsoft.com/kb/907877#top)Back to the top](http://support.microsoft.com/kb/907877#top)

**Aggregate memory**

The next section contains aggregate memory information for each clerk type and for each NUMA node. For a NUMA-enabled system, you may see output that is similar to the following.   
  
**Note** The following table contains only part of the output.

MEMORYCLERK\_SQLGENERAL (node 0) KB

---------------------------------------------------------------- --------------------

VM Reserved 0

VM Committed 0

AWE Allocated 0

SM Reserved 0

SM Commited 0

SinglePage Allocator 592

MultiPage Allocator 2160

(7 row(s) affected)

MEMORYCLERK\_SQLGENERAL (node 1) KB

---------------------------------------------------------------- --------------------

VM Reserved 0

VM Committed 0

AWE Allocated 0

SM Reserved 0

SM Commited 0

SinglePage Allocator 136

MultiPage Allocator 0

(7 row(s) affected)

MEMORYCLERK\_SQLGENERAL (Total) KB

---------------------------------------------------------------- --------------------

VM Reserved 0

VM Committed 0

AWE Allocated 0

SM Reserved 0

SM Commited 0

SinglePage Allocator 728

MultiPage Allocator 2160

(7 row(s) affected)

**Note** These node IDs correspond to the NUMA node configuration of the computer that is running SQL Server. The node IDs include possible software NUMA nodes that are defined on top of hardware NUMA nodes or on top of an SMP system. To find mapping between node IDs and CPUs for each node, view Information event ID number 17152. This event is logged in the Application log in Event Viewer when you start SQL Server.  
  
For an SMP system, you will see only one section for each clerk type. This section is similar to the following.

MEMORYCLERK\_SQLGENERAL (Total) KB

---------------------------------------------------------------- --------------------

VM Reserved 0

VM Committed 0

AWE Allocated 0

SM Reserved 0

SM Commited 0

SinglePage Allocator 768

MultiPage Allocator 2160

(7 row(s) affected)

Other information in these sections is about shared memory:

* **SM Reserved**: This value shows the VAS that is reserved by all clerks of this kind that are using the memory-mapped files API. This API is also known as *shared memory*.
* **SM Committed**: This value shows the VAS that is committed by all clerks of this kind that are using memory-mapped files API.

You can obtain summary information for each clerk type for all memory nodes by using the **sys.dm\_os\_memory\_clerks** dynamic management view (DMV). To do this, run the following query:

select

type,

sum(virtual\_memory\_reserved\_kb) as [VM Reserved],

sum(virtual\_memory\_committed\_kb) as [VM Committed],

sum(awe\_allocated\_kb) as [AWE Allocated],

sum(shared\_memory\_reserved\_kb) as [SM Reserved],

sum(shared\_memory\_committed\_kb) as [SM Committed],

sum(multi\_pages\_kb) as [MultiPage Allocator],

sum(single\_pages\_kb) as [SinlgePage Allocator]

from

sys.dm\_os\_memory\_clerks

group by type

[[http://support.microsoft.com/library/images/support/kbgraphics/public/en-us/uparrow.gif](http://support.microsoft.com/kb/907877#top)Back to the top](http://support.microsoft.com/kb/907877#top)

**Buffer distribution**

The next section shows the distribution of 8-kilobyte (KB) buffers in the buffer pool.

Buffer Distribution Buffers

------------------------------ -----------

Stolen 553

Free 103

Cached 161

Database (clean) 1353

Database (dirty) 38

I/O 0

Latched 0

(7 row(s) affected)

The elements in this section are the following:

* **Stolen**: *Stolen memory* describes 8-KB buffers that the server uses for miscellaneous purposes. These buffers serve as generic memory store allocations. Different components of the server use these buffers to store internal data structures. The lazywriter process is not permitted to flush Stolen buffers out of the buffer pool.
* **Free**: This value shows committed buffers that are not currently being used. These buffers are available for holding data. Or, other components may request these buffers and then mark these buffers as **Stolen**.
* **Cached**: This value shows the buffers that are used for various caches.
* **Database (clean)**: This value shows the buffers that have database content and that have not been modified.
* **Database (dirty)**: This value shows the buffers that have database content and that have been modified. These buffers contain changes that must be flushed to disk.
* **I/O**: This value shows the buffers that are waiting for a pending I/O operation.
* **Latched**: This value shows the *latched* buffers. A buffer is latched when a thread is reading or modifying the contents of a page. A buffer is also latched when the page is being read from disk or written to disk. A latch is used to maintain physical consistency of the data in the page while it is being read or modified. A lock is used to maintain logical and transactional consistency.

[[http://support.microsoft.com/library/images/support/kbgraphics/public/en-us/uparrow.gif](http://support.microsoft.com/kb/907877#top)Back to the top](http://support.microsoft.com/kb/907877#top)

**Buffer pool details**

You can obtain detailed information about buffer pool buffers for database pages by using the **sys.dm\_os\_buffer\_descriptors** DMV. And you can obtain detailed information about buffer pool pages that are being used for miscellaneous server purposes by using the **sys.dm\_os\_memory\_clerks** DMV.   
  
The next section lists details about the buffer pool plus additional information.

Buffer Counts Buffers

------------------------------ --------------------

Committed 1064

Target 17551

Hashed 345

Stolen Potential 121857

External Reservation 645

Min Free 64

Visible 17551

Available Paging File 451997

(8 row(s) affected)

The elements in this section are the following:

* **Committed**: This value shows the total buffers that are *committed*. Buffers that are committed have physical memory associated with them. The **Committed** value is the current size of the buffer pool. This value includes the physical memory that is allocated if AWE support is enabled.
* **Target**: This value shows the target size of the buffer pool. If the **Target** value is larger than the **Committed** value, the buffer pool is growing. If the **Target** value is less than the **Committed** value, the buffer pool is shrinking.
* **Hashed**: This value shows the data pages and index pages that are stored in the buffer pool.
* **Stolen Potential**: This value shows the maximum pages that can be stolen from the buffer pool.
* **ExternalReservation**: This value shows the pages that have been reserved for queries that will perform a sort operation or a hash operation. These pages have not yet been stolen.
* **Min Free**: This value shows the pages that the buffer pool tries to have on the free list.
* **Visible**: This value shows the buffers that are concurrently visible. These buffers can be directly accessed at the same time. This value is usually equal to the total buffers. However, when AWE support is enabled, this value may be less than the total buffers.
* **Available Paging File**: This value shows the memory that is available to be committed. This value is expressed as the number of 8-KB buffers. For more information, see the "GlobalMemoryStatusEx function" topic in the Windows API documentation.

[[http://support.microsoft.com/library/images/support/kbgraphics/public/en-us/uparrow.gif](http://support.microsoft.com/kb/907877#top)Back to the top](http://support.microsoft.com/kb/907877#top)

**Procedure cache**

The next section describes the makeup of the procedure cache.

Procedure Cache Value

------------------------------ -----------

TotalProcs 4

TotalPages 25

InUsePages 0

(3 row(s) affected)

The elements in this section are the following:

* **TotalProcs**: This value shows the total cached objects that are currently in the procedure cache. This value will match the entries in the **sys.dm\_exec\_cached\_plans** DMV.  
    
  **Note** Because of the dynamic nature of this information, the match may not be exact. You can use PerfMon to monitor the **SQL Server: Plan Cache** object and the **sys.dm\_exec\_cached\_plans** DMV for detailed information about the type of cached objects, such as triggers, procedures, and ad hoc objects.
* **TotalPages**: This value shows the cumulative pages that you must have to store all the cached objects in the procedure cache.
* **InUsePages**: This value shows the pages in the procedure cache that belong to procedures that are currently running. These pages cannot be discarded.

[[http://support.microsoft.com/library/images/support/kbgraphics/public/en-us/uparrow.gif](http://support.microsoft.com/kb/907877#top)Back to the top](http://support.microsoft.com/kb/907877#top)

**Global memory objects**

The next section contains information about various global memory objects. This section also contains information about how much memory the global memory objects use.

Global Memory Objects Buffers

------------------------------ --------------------

Resource 126

Locks 85

XDES 10

SETLS 2

SE Dataset Allocators 4

SubpDesc Allocators 2

SE SchemaManager 44

SQLCache 41

Replication 2

ServerGlobal 25

XP Global 2

SortTables 2

(12 row(s) affected)

The elements in this section are the following:

* **Resource**: This value shows the memory that the **Resource** object uses. The **Resource** object is used by the storage engine and for various server-wide structures.
* **Locks**: This value shows the memory that Lock Manager uses.
* **XDES**: This value shows the memory that Transaction Manager uses.
* **SETLS**: This value shows the memory that is used to allocate the Storage Engine-specific per-thread structure that uses thread local storage.
* **SE Dataset Allocators**: This value shows the memory that is used to allocate structures for table access through the **Access Methods** setting.
* **SubpDesc Allocators**: This value shows the memory that is used for managing subprocesses for parallel queries, backup operations, restore operations, database operations, file operations, mirroring, and asynchronous cursors. These subprocesses are also known as *parallel processes*.
* **SE SchemaManager**: This value shows the memory that Schema Manager uses to store Storage Engine-specific metadata.
* **SQLCache**: This value shows the memory that is used to store the text of ad hoc statements and of prepared statements.
* **Replication**: This value shows the memory that the server uses for internal replication subsystems.
* **ServerGlobal**: This value shows the global server memory object that is used generically by several subsystems.
* **XP Global**: This value shows the memory that extended stored procedures use.
* **Sort Tables**: This value shows the memory that sort tables use.

[[http://support.microsoft.com/library/images/support/kbgraphics/public/en-us/uparrow.gif](http://support.microsoft.com/kb/907877#top)Back to the top](http://support.microsoft.com/kb/907877#top)

**Query memory objects**

The next section describes Query Memory grant information. This section includes a snapshot of the query memory usage. Query memory is also known as *workspace memory*.

Query Memory Objects Value

------------------------------ -----------

Grants 0

Waiting 0

Available (Buffers) 14820

Maximum (Buffers) 14820

Limit 10880

Next Request 0

Waiting For 0

Cost 0

Timeout 0

Wait Time 0

Last Target 11520

(11 row(s) affected)

Small Query Memory Objects Value

------------------------------ -----------

Grants 0

Waiting 0

Available (Buffers) 640

Maximum (Buffers) 640

Limit 640

(5 row(s) affected)

If the size and the cost of a query satisfy “small” query memory thresholds, the query is put in a small query queue. This behavior prevents smaller queries from being delayed behind larger queries that are already in the queue.  
  
The elements in this section are the following:

* **Grants**: This value shows the running queries that have memory grants.
* **Waiting**: This value shows the queries that are waiting to obtain memory grants.
* **Available**: This value shows the buffers that are available to queries for use as hash workspace and as sort workspace. The **Available** value is updated periodically.
* **Maximum**: This value shows the total buffers that can be given to all queries for use as workspace.
* **Limit**: This value shows the query execution target for the big query queue. This value differs from the **Maximum (Buffers)** value because the **Maximum (Buffers)** value is not updated until there is change in the queue.
* **Next Request**: This value shows the memory request size, in buffers, for the next waiting query.
* **Waiting For**: This value shows the amount of memory that must be available to run the query to which the **Next Request** value refers. The **Waiting For** value is the **Next Request** value multiplied by a headroom factor. This value effectively guarantees that a specific amount of memory will be available when the next waiting query is run.
* **Cost**: This value shows the cost of the next waiting query.
* **Timeout**: This value shows the time-out, in seconds, for the next waiting query.
* **Wait Time**: This value shows the elapsed time, in milliseconds, since the next waiting query was put in the queue.
* **Last Target**: This value shows the overall memory limit for query execution. This value is the combined limit for both the big query queue and the small query queue.

[[http://support.microsoft.com/library/images/support/kbgraphics/public/en-us/uparrow.gif](http://support.microsoft.com/kb/907877#top)Back to the top](http://support.microsoft.com/kb/907877#top)

**Optimization**

The next section is a summary of the users who are trying to optimize queries at the same time.

Optimization Queue Value

------------------------------ --------------------

Overall Memory 156672000

Last Notification 1

Timeout 6

Early Termination Factor 5

(4 row(s) affected)

Small Gateway Value

------------------------------ --------------------

Configured Units 8

Available Units 8

Acquires 0

Waiters 0

Threshold Factor 250000

Threshold 250000

(6 row(s) affected)

Medium Gateway Value

------------------------------ --------------------

Configured Units 2

Available Units 2

Acquires 0

Waiters 0

Threshold Factor 12

(5 row(s) affected)

Big Gateway Value

------------------------------ --------------------

Configured Units 1

Available Units 1

Acquires 0

Waiters 0

Threshold Factor 8

(5 row(s) affected)

Queries are submitted to the server for compilation. The compilation process includes parsing, algebraization, and optimization. Queries are classified based on the amount of memory that each query will consume during the compilation process.  
  
**Note** This amount does not include the memory that is required to run the query.   
  
When a query starts, there is no limit on how many queries can be compiled. As the memory consumption increases and reaches a threshold, the query must pass a gateway to continue. There is a progressively decreasing limit of simultaneously compiled queries after each gateway. The size of each gateway depends on the platform and the load. Gateway sizes are chosen to maximize scalability and throughput.  
  
If the query cannot pass a gateway, the query will wait until memory is available. Or, the query will return a time-out error (Error 8628). Additionally, the query may not acquire a gateway if the user cancels the query or if a deadlock is detected. If a query passes several gateways, the query does not release the smaller gateways until the compilation process has completed.   
  
This behavior lets only a few memory-intensive compilations occur at the same time. Additionally, this behavior maximizes throughput for smaller queries.

[[http://support.microsoft.com/library/images/support/kbgraphics/public/en-us/uparrow.gif](http://support.microsoft.com/kb/907877#top)Back to the top](http://support.microsoft.com/kb/907877#top)

**Memory brokers**

The next three sections show information about memory brokers that control cached memory, stolen memory, and reserved memory. Information that these sections provide can only be used for internal diagnostics. Therefore, this information is not detailed here.

[[http://support.microsoft.com/library/images/support/kbgraphics/public/en-us/uparrow.gif](http://support.microsoft.com/kb/907877#top)Back to the top](http://support.microsoft.com/kb/907877#top)

MEMORYBROKER\_FOR\_CACHE Value

-------------------------------- --------------------

Allocations 1843

Rate 0

Target Allocations 1843

Future Allocations 0

Last Notification 1

(4 row(s) affected)

MEMORYBROKER\_FOR\_STEAL Value

-------------------------------- --------------------

Allocations 380

Rate 0

Target Allocations 1195

Future Allocations 0

Last Notification 1

(4 row(s) affected)

MEMORYBROKER\_FOR\_RESERVE Value

-------------------------------- --------------------

Allocations 0

Rate 0

Target Allocations 1195

Future Allocations 0

Last Notification 1

(4 row(s) affected)

[[http://support.microsoft.com/library/images/support/kbgraphics/public/en-us/uparrow.gif](http://support.microsoft.com/kb/907877#top)Back to the top](http://support.microsoft.com/kb/907877#top)

**APPLIES TO**

* Microsoft SQL Server 2005 Developer Edition
* Microsoft SQL Server 2005 Enterprise Edition
* Microsoft SQL Server 2005 Enterprise Edition for Itanium-based Systems
* Microsoft SQL Server 2005 Enterprise X64 Edition
* Microsoft SQL Server 2005 Standard Edition
* Microsoft SQL Server 2005 Standard X64 Edition
* Microsoft SQL Server 2005 Workgroup Edition
* Microsoft SQL Server 2005 Standard Edition for Itanium-based Systems

[[http://support.microsoft.com/library/images/support/kbgraphics/public/en-us/uparrow.gif](http://support.microsoft.com/kb/907877#top)Back to the top](http://support.microsoft.com/kb/907877#top)

|  |  |
| --- | --- |
| **Keywords:** | kbsql2005engine kbtshoot kbinfo KB907877 |

[[http://support.microsoft.com/library/images/support/kbgraphics/public/en-us/uparrow.gif](http://support.microsoft.com/kb/907877#top)Back to the top](http://support.microsoft.com/kb/907877#top)