**Using DBCC MEMORYSTATUS to Monitor SQL Server Memory Usage**

[View products that this article applies to.](http://support.microsoft.com/kb/271624#appliesto)

This article was previously published under Q271624

[**On This Page**](javascript:void(0);)

* [[http://support.microsoft.com/library/images/support/kbgraphics/public/en-us/downarrow.gif](http://support.microsoft.com/kb/271624)SUMMARY](http://support.microsoft.com/kb/271624)
* [[http://support.microsoft.com/library/images/support/kbgraphics/public/en-us/downarrow.gif](http://support.microsoft.com/kb/271624)MORE INFORMATION](http://support.microsoft.com/kb/271624)
  + [[http://support.microsoft.com/library/images/support/kbgraphics/public/en-us/downarrow.gif](http://support.microsoft.com/kb/271624)First Section](http://support.microsoft.com/kb/271624)
  + [[http://support.microsoft.com/library/images/support/kbgraphics/public/en-us/downarrow.gif](http://support.microsoft.com/kb/271624)Second Section](http://support.microsoft.com/kb/271624)
  + [[http://support.microsoft.com/library/images/support/kbgraphics/public/en-us/downarrow.gif](http://support.microsoft.com/kb/271624)Third Section](http://support.microsoft.com/kb/271624)
  + [[http://support.microsoft.com/library/images/support/kbgraphics/public/en-us/downarrow.gif](http://support.microsoft.com/kb/271624)Fourth Section](http://support.microsoft.com/kb/271624)
* [[http://support.microsoft.com/library/images/support/kbgraphics/public/en-us/downarrow.gif](http://support.microsoft.com/kb/271624)REFERENCES](http://support.microsoft.com/kb/271624)

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

[SUMMARY](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 SQL Server. The output from this command is useful in troubleshooting issues that relate to the memory consumption of SQL Server or to specific out-of-memory errors (many of which automatically print this output in the error log). Microsoft Product Support Services may also request that you run this command during the course of a specific support incident if you are encountering an error that may be associated with a low memory condition.  
  
This command is intended to be a diagnostic tool for Microsoft Product Support Services. The format of the output and the level of detail that is provided is subject to change between service pack and product releases. The functionality provided by the DBCC MEMORYSTATUS command will likely be replaced by a different mechanism in future product versions, and this command may no longer function. No additional warnings will be made before changing or removing this command. Therefore, applications that rely on its behavior may break without warning.  
  
The output of DBCC MEMORYSTATUS contains seven sections. This article describes some of the key data that can be gleaned from the output in the first four sections. The remaining sections involve proprietary implementation details and they are not explained in this article. Microsoft Product Support Services will not answer any questions or provide additional information about the meaning of specific counters beyond what is supplied in this article.

[[http://support.microsoft.com/library/images/support/kbgraphics/public/en-us/uparrow.gif](http://support.microsoft.com/kb/271624#top)Back to the top](http://support.microsoft.com/kb/271624#top)

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

**First SectionThe first section describes the distribution of 8 KB buffers in the...**

**First Section**

The first section describes the distribution of 8 KB buffers in the buffer pool. There is a buffer structure for each potential 8 KB page in the buffer pool. This includes physical pages if Address Windowing Extensions (AWE) is enabled. This structure contains a number of bits that indicate the status of the underlying page (that is, whether the page has been modified, is pinned, is in IO, and more). This section details the number of buffers that have specific status bits.

Buffer Distribution Buffers

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

Stolen 241

Free 95

Procedures 89

Inram 0

Dirty 16

Kept 0

I/O 0

Latched 18

Other 880

(9 row(s) affected)

**Stolen**. **Stolen** memory describes buffers that are in use for sorting or for hashing operations (query workspace memory), or for those buffers that are being used as a generic memory store for allocations to store internal data structures such as locks, transaction context, and connection information. The lazywriter process is not permitted to flush **Stolen** buffers out of the buffer pool.   
  
**Free**. **Free** refers to committed buffers that are not currently in use. These are available for holding data, or they may be requested by other components and marked as **Stolen**.  
  
**Procedures**. **Procedures** refers to buffers that are holding cached stored procedure memory. These buffers contain compiled and executable plans for procedures and for cached ad hoc plans.  
  
**Inram**. **Inram** refers to pages for pinned tables that cannot be removed from memory (pinned by using the DBCC PINTABLE command or the **sp\_tableoption** stored procedure).  
  
**Dirty**. **Dirty** refers to data pages that contain changes that have not yet been flushed to disk.  
  
**Kept**. **Kept** refers to pages that are temporarily pinned in memory to prevent them from being flushed. This value is not used in SQL Server 2000.  
  
**I/O**. **I/O** refers to buffers that are waiting on a pending I/O operation. On SQL Server 2000, this includes mapping or unmapping a buffer through AWE.  
  
**Latched**. The latched buffers record the number of buffers where a connection is reading or is modifying a row from the page. A latch is used to ensure physical consistency of the data in the page while it is read or modified, while a lock is used to ensure logical and transactional consistency.  
  
**Other**. These are committed pages that do not meet any of the criteria mentioned earlier. Typically, the majority of buffers that meet this criteria are hashed data and index pages in the buffer cache.

[[http://support.microsoft.com/library/images/support/kbgraphics/public/en-us/uparrow.gif](http://support.microsoft.com/kb/271624#top)Back to the top](http://support.microsoft.com/kb/271624#top)

**Second Section**

The second section lists some totals that reflect the overall size and the makeup of the buffer pool.

Buffer Counts Buffers

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

Committed 1339

Target 6318

Hashed 914

InternalReservation 120

ExternalReservation 0

Min Free 16

(6 row(s) affected)

**Committed.** The total number of buffers that are committed in Microsoft Windows NT. The **Committed** value is the current "size" of the buffer pool. This value includes the physical memory that is allocated if AWE is enabled.  
  
**Target**. The target committed count, or how big the buffer pool would like to be. If the **Target** count is greater than the **Committed** value, the buffer pool is growing. If the **Target** count is less than the **Committed** value, the buffer pool is shrinking. The lazywriter process uses this value to determine whether to grow or to shrink the buffer pool.  
  
**Hashed**. The number of data and index pages that are stored in the buffer pool.  
  
**InternalReservation**. The number of pages that the buffer pool keeps for itself. The **InternalReservation** value is the minimum number of pages that are required to satisfy the simple data demands for the current number of users.  
  
**ExternalReservation**. The number of pages that have been reserved for queries that will perform a sort or a hash operation but that have not yet been stolen. This is used as a hint to the lazywriter that there will be an upcoming requirement for a large amount of memory.  
  
**Min Free**. The number of pages that the buffer pool tries to keep on the free list. If the free list falls below the **Min Free** value, the buffer pool tries to populate the free list by discarding old pages from the data cache or old procedures from the procedure cache.

[[http://support.microsoft.com/library/images/support/kbgraphics/public/en-us/uparrow.gif](http://support.microsoft.com/kb/271624#top)Back to the top](http://support.microsoft.com/kb/271624#top)

**Third Section**

The third section describes the makeup of the procedure cache.

Procedure Cache Value

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

TotalProcs 17

TotalPages 89

InUsePages 40

(3 row(s) affected)

**TotalProcs**. The total number of cached objects currently in the procedure cache. This value will match the number of entries in the **syscacheobjects** virtual table. You can use Performance Monitor to monitor the SQL Server:Cache Manager object for a detailed breakdown of the type of cached objects --for example, triggers, procedures, and ad hoc.  
  
**TotalPages**. The cumulative number of pages that you must have to store all the cached objects.  
  
**InUsePages**. The number of pages in the procedure cache that belong to procedures that are currently executing. These cannot be discarded.

[[http://support.microsoft.com/library/images/support/kbgraphics/public/en-us/uparrow.gif](http://support.microsoft.com/kb/271624#top)Back to the top](http://support.microsoft.com/kb/271624#top)

**Fourth Section**

The fourth section describes the relative distribution of stolen buffers among major component groups in the server. SQL Server implements its own memory managers that use buffer pool pages for small allocations and that fall back to operating system routines only for allocations that are larger than approximately 8 KB. These larger allocations are outside the buffer pool, and they are commonly referred to as allocations from the **MemToLeave** area. The size of this area can be controlled by using the **-g** command-line option.   
  
Typically, the majority of memory allocations from **MemToLeave** are from non-SQL Server memory consumers that are running in-process such as COM objects, extended stored procedures, and linked servers. The operating system-related values that are reported in DBCC MEMORYSTATUS only reflect **MemToLeave** allocations that are made by SQL Server directly. Any other allocations that occur in **MemToLeave** from these other components are not included because SQL Server has no knowledge of these memory requests.

Dynamic Memory Manager Buffers

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

Stolen 330

OS Reserved 152

OS Committed 138

OS In Use 129

General 322

QueryPlan 87

Optimizer 0

Utilities 10

Connection 40

(9 row(s) affected)

**Stolen**. Total number of buffers that are stolen from the buffer pool by the five memory managers (General, Query Plan, Optimizer, Utilities, Connection). These pages are used to service memory allocations that are smaller than 8 KB.  
  
**OS Reserved**. The number of pages that are reserved from the operating system to handle allocation requests that are larger than approximately 8 KB.  
  
**OS Committed**. The amount of memory in the reserved regions that has been committed. The **OS Committed** value should be less than or equal to the **OS Reserved** value because buffers are first reserved and then committed. Some of the memory may be decommitted when it is no longer used.   
  
**Note** There is a bug in SQL Server 2000 (that is fixed in SQL Server 2000 Service Pack 3) that causes this value not to be maintained accurately and therefore it may fall outside the valid range.  
  
**OS In Use**. The **OS Committed** buffers that are currently backing outstanding memory allocations.  
  
**General**. Memory that is used by general memory consumers in the server, including parsing or normalization, locks, transaction context, internal data structures describing the in-memory metadata for tables and indexes, and others. This value is the total number of 8 KB buffers. Some of these may be stolen buffers from the buffer pool, and others may be from **MemToLeave**.  
  
**QueryPlan**. Memory used to store query plans, either transiently or in cache. This value is the total number of 8 KB buffers. Some of these may be stolen buffers from the buffer pool, and others may be from **MemToLeave**.  
  
**Optimizer**. Memory that is in use by the query optimizer. This value is the total number of 8 KB buffers. Some of these may be stolen buffers from the buffer pool, and others may be from **MemToLeave**.  
  
**Utilities**. Memory in use by various utilities routines such as BCP, Log Manager, parallel queries, **::fn\_trace\_gettable**, and others. This value is the total number of 8 KB buffers. Some of these may be stolen buffers from the buffer pool, and others may be from **MemToLeave**.  
  
**Connection**. Memory that is used to store the data structures holding the connection context for each user connection. This also includes the memory for the network send and receive buffers that are associated with a connection. This value is the total number of 8 KB buffers. Some of these may be stolen buffers from the buffer pool, and others that may be from **MemToLeave**.