[**http://msdn.microsoft.com/en-us/library/ms177526(SQL.90).aspx**](http://msdn.microsoft.com/en-us/library/ms177526(SQL.90).aspx)

**sys.dm\_os\_schedulers**

**SQL Server 2005**

[Other Versions](javascript:;)

http://i3.msdn.microsoft.com/Hash/33169f5f6ffee1bf41ee6eb0b266ef57.png

* [SQL Server 2008](http://msdn.microsoft.com/en-us/library/ms177526(v=SQL.100).aspx)

Updated: **14 April 2006**

Returns one row per scheduler in SQL Server where each scheduler is mapped to an individual processor. Use this view to monitor the condition of a scheduler or to identify runaway tasks.

|  |  |  |
| --- | --- | --- |
| Column name | Data type | Description |
| **scheduler\_address** | **varbinary(8)** | Memory address of the scheduler. Is not nullable. |
| **parent\_node\_id** | **int** | ID of the node that the scheduler belongs to, also known as the parent node. This represents a nonuniform memory access (NUMA) node. Is not nullable. |
| **scheduler\_id** | **int** | ID of the scheduler. All schedulers that are used to run regular queries have ID numbers less than 255. Those schedulers that have IDs greater than or equal to 255 are used internally by SQL Server, such as the dedicated administrator connection scheduler. Is not nullable. |
| **cpu\_id** | **smallint** | ID of the CPU with which this scheduler is associated. If SQL Server is configured to run with affinity, the value is the ID of the CPU on which the scheduler is supposed to be running.  255 = Affinity mask is not specified.  Is not nullable. |
| **status** | **nvarchar(60)** | Indicates the status of the scheduler. Can be one of the following values:   * HIDDEN ONLINE * HIDDEN OFFLINE * VISIBLE ONLINE * VISIBLE OFFLINE * VISIBLE ONLINE (DAC)   Is not nullable.  HIDDEN schedulers are used to process requests that are internal to the Database Engine. VISIBLE schedulers are used to process user requests.  OFFLINE schedulers map to processors that are offline in the affinity mask and are, therefore, not being used to process any requests. ONLINE schedulers map to processors that are online in the affinity mask and are available to process threads.  DAC indicates the scheduler is running under a dedicated administrator connection. |
| **is\_online** | **bit** | If SQL Server is configured to use only some of the available processors on the server, this configuration can mean that some schedulers are mapped to processors that are not in the affinity mask. If that is the case, this column returns 0. This value means that the scheduler is not being used to process queries or batches.  Is not nullable. |
| **is\_idle** | **bit** | 1 = Scheduler is idle. No workers are currently running. Is not nullable. |
| **preemptive\_switches\_count** | **int** | Number of times that workers on this scheduler have switched to the preemptive mode.  To execute code that is outside SQL Server (for example, extended stored procedures and distributed queries), a thread has to execute outside the control of the non-preemptive scheduler. To do this, a worker switches to preemptive mode. |
| **context\_switches\_count** | **int** | Number of context switches that have occurred on this scheduler. Is not nullable.  To allow for other workers to run, the current running worker has to relinquish control of the scheduler or switch context.   |  | | --- | | **ms177526.note(en-US,SQL.90).gifNote:** If a worker yields the scheduler and puts itself into the runnable queue and then finds no other workers, the worker will select itself. In this case, the **context\_switches\_count** is not updated, but the **yield\_count** is updated. | |
| **idle\_switches\_count** | **int** | Number of times the scheduler has been waiting for an event while idle. This column is similar to **context\_switches\_count**. Is not nullable. |
| **current\_tasks\_count** | **int** | Number of current tasks that are associated with this scheduler. This count includes the following:   * Tasks that are waiting for a worker to execute them. * Tasks that are currently waiting or running (in SUSPENDED or RUNNABLE state).   When a task is completed, this count is decremented. Is not nullable. |
| **runnable\_tasks\_count** | **int** | Number of workers, with tasks assigned to them, that are waiting to be scheduled on the runnable queue. Is not nullable. |
| **current\_workers\_count** | **int** | Number of workers that are associated with this scheduler. This count includes workers that are not assigned any task. Is not nullable. |
| **active\_workers\_count** | **int** | Number of workers that are active. An active worker is always nonpreemptive, must have an associated task, and is either running, runnable, or suspended. Is not nullable. |
| **work\_queue\_count** | **bigint** | Number of tasks in the pending queue. These tasks are waiting for a worker to pick them up. Is not nullable. |
| **pending\_disk\_io\_count** | **int** | Number of pending I/Os that are waiting to be completed. Each scheduler has a list of pending I/Os that are checked to determine whether they have been completed every time there is a context switch. The count is incremented when the request is inserted. This count is decremented when the request is completed. This number does not indicate the state of the I/Os. Is not nullable. |
| **load\_factor** | **int** | Internal value that indicates the perceived load on this scheduler. This value is used to determine whether a new task should be put on this scheduler or another scheduler. This value is useful for debugging purposes when it appears that schedulers are not evenly loaded. In SQL Server 2000, a task is routed to a particular scheduler. However, in SQL Server 2005, the routing decision is made based on the load on the scheduler. SQL Server 2005 also uses a load factor of nodes and schedulers to help determine the best location to acquire resources. When a task is enqueued, the load factor is increased. When a task is completed, the load factor is decreased. Using the load factors helps SQL Server OS balance the work load better. Is not nullable. |
| **yield\_count** | **int** | Internal value that is used to indicate progress on this scheduler. This value is used by the Scheduler Monitor to determine whether a worker on the scheduler is not yielding to other workers on time. This value does not indicate that the worker or task transitioned to a new worker. Is not nullable. |
| **last\_timer\_activity** | **bigint** | In CPU ticks, the last time that the scheduler timer queue was checked by the scheduler. Is not nullable. |
| **failed\_to\_create\_worker** | **bit** | Set to 1 if a new worker could not be created on this scheduler. This generally occurs because of memory constraints. Is nullable. |
| **active\_worker\_address** | **varbinary(8)** | Memory address of the worker that is currently active. Is nullable. For more information, see [sys.dm\_os\_workers](http://msdn.microsoft.com/en-us/library/ms178626(v=SQL.90).aspx). |
| **memory\_object\_address** | **varbinary(8)** | Memory address of the scheduler memory object. Not NULLABLE. |
| **task\_memory\_object\_address** | **varbinary(8)** | Memory address of the task memory object. Is not nullable. For more information, see [sys.dm\_os\_memory\_objects](http://msdn.microsoft.com/en-us/library/ms179875(v=SQL.90).aspx). |

**Permissions**

Requires VIEW SERVER STATE permission on the server.

**Examples**

**A. Monitoring hidden and nonhidden schedulers**

The following query outputs the state of workers and tasks in SQL Server across all schedulers. This query was executed on a computer system that has the following:

* Two processors (CPUs)
* Two (NUMA) nodes
* One CPU per NUMA node
* Affinity mask set to 0x03.

[Copy](javascript:CodeSnippet_CopyCode('CodeSnippetContainerCode0');" \o "Copy to clipboard.)

SELECT

scheduler\_id,

cpu\_id,

parent\_node\_id,

current\_tasks\_count,

runnable\_tasks\_count,

current\_workers\_count,

active\_workers\_count,

work\_queue\_count

FROM sys.dm\_os\_schedulers;

Here is the result set.

[Copy](javascript:CodeSnippet_CopyCode('CodeSnippetContainerCode1');" \o "Copy to clipboard.)

scheduler\_id cpu\_id parent\_node\_id current\_tasks\_count

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

0 1 0 9

257 255 0 1

1 0 1 10

258 255 1 1

255 255 32 2

runnable\_tasks\_count current\_workers\_count

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

0 11

0 1

0 18

0 1

0 3

active\_workers\_count work\_queue\_count

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

6 0

1 0

8 0

1 0

1 0

The output provides the following information:

* There are five schedules. Two schedulers have an ID value < 255. Schedulers with ID >= 255 are known as hidden schedulers. Scheduler 255 represents the dedicated administrator connection (DAC). There is one DAC scheduler per instance. Resource monitors that coordinate memory pressure use scheduler 257 and scheduler 258, one per NUMA node
* There are 23 active tasks in the output. These tasks include user requests in addition to resource management tasks that have been started by SQL Server. Examples of SQL Server tasks are RESOURCE MONITOR (one per NUMA node), LAZY WRITER (one per NUMA node), LOCK MONITOR, CHECKPOINT, and LOG WRITER.
* NUMA node 0 is mapped to CPU 1 and NUMA node 1 is mapped to CPU 0. SQL Server typically starts on a NUMA node other than node 0. For more information, see [Understanding Non-uniform Memory Access](http://msdn.microsoft.com/en-us/library/ms178144(v=SQL.90).aspx).
* With runnable\_tasks\_count returning 0, there are no actively running tasks. However, active sessions may exist.
* Scheduler 255 representing DAC has 3 workers associated with it. These workers are allocated at SQL Server startup and do not change. These workers are used to process DAC queries only. The two tasks on this scheduler represent a connection manager and an idle worker.
* active\_workers\_count represents all workers that have associated tasks and are running under non-preemptive mode. Some tasks, such as network listeners, run under preemptive scheduling.
* Hidden schedulers do not process typical user requests. The DAC scheduler is the exception. This DAC scheduler has one thread to process requests.

**B. Monitoring nonhidden schedulers in a busy system**

The following query shows the state of heavily loaded nonhidden schedulers, where more requests exist than can be handled by available workers. In this example, 256 workers are assigned tasks. Some tasks are waiting for an assignment to a worker. Lower runnable count implies that multiple tasks are waiting for a resource.

|  |
| --- |
| **ms177526.note(en-US,SQL.90).gifNote:** |
| You can find the state of workers by querying **sys.dm\_os\_workers**. For more information, see [sys.dm\_os\_workers](http://msdn.microsoft.com/en-us/library/ms178626(v=SQL.90).aspx). |

Here is the query:

[Copy](javascript:CodeSnippet_CopyCode('CodeSnippetContainerCode2');" \o "Copy to clipboard.)

SELECT

scheduler\_id,

cpu\_id,

current\_tasks\_count,

runnable\_tasks\_count,

current\_workers\_count,

active\_workers\_count,

work\_queue\_count

FROM sys.dm\_os\_schedulers

WHERE scheduler\_id < 255;

Here is the result set.

[Copy](javascript:CodeSnippet_CopyCode('CodeSnippetContainerCode3');" \o "Copy to clipboard.)

scheduler\_id current\_tasks\_count runnable\_tasks\_count

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

0 144 0

1 147 1

current\_workers\_count active\_workers\_count work\_queue\_count

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

128 125 16

128 126 19

By comparison, the following result shows multiple runnable tasks where no task is waiting to obtain a worker. The work\_queue\_count is 0 for both schedulers.

[Copy](javascript:CodeSnippet_CopyCode('CodeSnippetContainerCode4');" \o "Copy to clipboard.)

scheduler\_id current\_tasks\_count runnable\_tasks\_count

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

0 107 98

1 110 100

current\_workers\_count active\_workers\_count work\_queue\_count

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

128 104 0

128 108 0