Sys.dm_os_performance_counters (Transact-SQL)

Returns a row per performance counter maintained by the server.

Column name	*Data type*	*Description*
object_name	**nchar(128)**	Category to which this counter belongs.
counter_name	**nchar(128)**	Name of the counter.
instance_name	**nchar(128)**	Name of the specific instance of the counter. Often contains the database name.
cntr_value	**bigint**	Current value of the counter. For per-second counters, this value is cumulative. The rate value must be calculated by sampling the value at discrete time intervals. The difference between any two successive sample values is equal to the rate for the time interval used.
cntr_type	**int**	Type of counter as defined by the Windows performance architecture.

```
-- confirm that performance counters have been disabled
SELECT COUNT(*) FROM sys.dm_os_performance_counters;
```

Results | Messages

	(No column name)
1	5104

If the return value is 0 rows -> means that the performance counters have been disabled -> look at the setup log and search for error 3409, "Reinstall sqlctr.ini for this instance, and ensure that the instance login account has correct registry permissions." -> This denotes that performance counters were not enabled. The errors immediately before the 3409 error should indicate the root cause for the failure of performance counter enabling.

On SQL Server, requires VIEW SERVER STATE permission. On SQL Database, requires the VIEW DATABASE STATE permission in the database.

```
-- The following example returns performance counter values.
SELECT object_name, counter_name, instance_name, cntr_value, cntr_type
FROM sys.dm_os_performance_counters;
```

Results | Messages

	object_name	counter_name	instance_name	cntr_value	cntr_type
1	MSSQL$SQLEXPRESS:Memory Broker Clerks	Memory broker clerk size	Buffer Pool	1593	65792
2	MSSQL$SQLEXPRESS:Memory Broker Clerks	Simulation benefit	Buffer Pool	0	65792
3	MSSQL$SQLEXPRESS:Memory Broker Clerks	Simulation size	Buffer Pool	387	65792
4	MSSQL$SQLEXPRESS:Memory Broker Clerks	Internal benefit	Buffer Pool	0	65792
5	MSSQL$SQLEXPRESS:Memory Broker Clerks	Periodic evictions (pages)	Buffer Pool	0	65792
6	MSSQL$SQLEXPRESS:Memory Broker Clerks	Pressure evictions (pages/sec)	Buffer Pool	0	272696576
7	MSSQL$SQLEXPRESS:Memory Broker Clerks	Memory broker clerk size	Column store object pool	4	65792
8	MSSQL$SQLEXPRESS:Memory Broker Clerks	Simulation benefit	Column store object pool	0	65792
9	MSSQL$SQLEXPRESS:Memory Broker Clerks	Simulation size	Column store object pool	0	65792
10	MSSQL$SQLEXPRESS:Memory Broker Clerks	Internal benefit	Column store object pool	0	65792
11	MSSQL$SQLEXPRESS:Memory Broker Clerks	Periodic evictions (pages)	Column store object pool	0	65792

Query executed successfully. | DESKTOP-ATJN5FL\SQLEXPRESS ... | DESKTOP-ATJN5FL\Emi (51) | master | 00:00:00 | 5386 rows

The performance counters exposed by SQL Server are invaluable tools for monitoring various aspects of the instance health. The counter data is exposed as a shared memory object for the windows performance monitoring tools to query. The counter data exposed in the view are in a raw form (-> needs to be interpreted appropriately before it can be used). The **cntr_type** column value indicates how the values have to be interpreted. The type of each counter is indicated in the **cntr_type** column as a decimal value. The distinct values are:

Decimal	**Hexadecimal**	**Counter type define**
1073939712	0x40030500	PERF_LARGE_RAW_BASE

537003264	0x20020500	PERF_LARGE_RAW_FRACTION
1073874176	0x40020500	PERF_AVERAGE_BULK
272696576	0x10410500	PERF_COUNTER_BULK_COUNT
65792	0x00010100	PERF_COUNTER_LARGE_RAWCOUNT

1) PERF_LARGE_RAW_BASE

The *cntr_types* column value for the PERF_LARGE_RAW_BASE counter type is 1073939712 (0x40030500 – in hexadecimal). This counter value is raw data that is used as the denominator of a counter that presents a instantaneous arithmetic fraction (or denominator for further calculation). These counters collect the last observed value. The counters of this type are only used to calculate other counters available via the view. All counters that belong to this counter type have the word base in their names, so it is not a counter that provides useful info, it's just a base value for further calculations.

object_name	counter_name	instance_name	cntr_value	cntr_type
MSSQL$SQLSVR:Buffer Manager	Buffer cache hit ratio base		3170	1073939712

This value is the base for the MSSQL$SQLSVR: Buffer Manager\Buffer cache hit ratio calculation.

```
SELECT *
FROM sys.dm_os_performance_counters
WHERE counter_name = 'Buffer cache hit ratio base'
```

Results | Messages

	object_name	counter_name	instance_name	cntr_value	cntr_type
1	MSSQL$SQLEXPRESS:Buffer Manager	Buffer cache hit ratio base		176	1073939712

Some of these counters are: *Buffer Cache Hit Ratio Base, Log Cache Hit Ratio Base, Average Latch Wait Time Base, Cache Hit Ratio Base, CPU usage % base,* and more.

2) PERF_LARGE_RAW_FRACTION

The *cntr_types* column value for the PERF_LARGE_RAW_FRACTION counter type is 537003264 (0x20020500 – in hexadecimal). This counter value represents a fractional value as a ratio to its corresponding PERF_LARGE_RAW_BASE counter value. These counters show a ratio (presented in percents), (i.e. fraction between two values – the PERF_LARGE_RAW_FRACTION counter and its corresponding PERF_LARGE_RAW_BASE counter value). Additional calculation is needed to find out the value used for monitoring and troubleshooting performance issues.

object_name	counter_name	instance_name	cntr_value	cntr_type
MSSQL$SQLSVR:Buffer Manager	Buffer cache hit ratio		2911	537003264

Using the value and the base value from the previous example, one has
Hit ratio % = 100 * MSSQL$SQLSVR:Buffer Manager\Buffer cache hit ratio / MSSQL$SQLSVR:Buffer Manager\Buffer cache hit ratio base = 100 * 2911 / 3170 = 91.83%

Buffer Cache Hit Ratio shows how SQL Server utilizes buffer cache and it is the ratio of the data pages found and read from the SQL Server buffer cache and all data page requests. The recommended values are higher than 90%. The maximal possible value is 100% – when SQL Server reads all pages from the buffer cache and none from disk.

```
SELECT * FROM sys.dm_os_performance_counters
WHERE counter_name = 'Buffer cache hit ratio'
```

Results | Messages

	object_name	counter_name	instance_name	cntr_value	cntr_type
1	MSSQL$SQLEXPRESS:Buffer Manager	Buffer cache hit ratio		176	537003264

For this example, the counter shows that *Buffer Cache Hit Ratio* is 2,135%, which is not possible Calculate *Buffer Cache Hit Ratio* (with the counter value of the *Buffer Cache Hit Ratio* counter and the Buffer Cache Hit Ratio Base value from the example from PERF_LARGE_RAW_BASE counter type above) *Buffer Cache Hit Ratio* % = 100 * *Buffer Manager\Buffer Cache Hit Ratio / Buffer Manager\Buffer Cache Hit Ratio Base* = 100 * 2,135 / 3,573 = 59.75%
Some of the counters are: *Buffer Cache Hit Ratio, Log Cache Hit Ratio, Worktables From Cache Ratio, Cache Hit Ratio, CPU usage %, and Rem Req Cache Hit Ratio.*

3) PERF_AVERAGE_BULK

The *cntr_types* column value for the PERF_AVERAGE_BULK counter type is 1073874176 (0x40020500 – in hexadecimal). This counter value represents an average metric. The **cntr_value** is cumulative. The base value of type PERF_LARGE_RAW_BASE is used which is also cumulative. The value is obtained by first taking two samples of both the PERF_AVERAGE_BULK value A1 and A2 as well as the PERF_LARGE_RAW_BASE value B1 and B2. The difference between A1 and A2 and B1 and B2 are calculated. The final value is then calculated as the ratio of the differences.

Sample 1

object_name	**counter_name**	**instance_name**	**cntr_value**	**cntr_type**	
MSSQL$SQLSVR:Latches	Average Latch Wait Time (ms)		14257	1073874176	<== A1
MSSQL$SQLSVR:Latches	Average Latch Wait Time Base		359	1073939712	<== B1

Sample 2

object_name	**counter_name**	**instance_name**	**cntr_value**	**cntr_type**	
MSSQL$SQLSVR:Latches	Average Latch Wait Time (ms)		14272	1073874176	<== A2
MSSQL$SQLSVR:Latches	Average Latch Wait Time Base		360	1073939712	<== B2

Average Latch Wait Time (ms) for the interval = (A2 - A1) / (B2 - B1) = (14272 - 14257) / (360 - 359) = 15.00 ms

Now, consider *Average Wait Time (ms)* and *Average Wait Time Base* values, any two counters one of the 1073939712 type and the other 1073874176, that have identical names except for the word base: *Update conflict ratio base and Update conflict ratio, Avg. Time to Write Batch Base and Avg. Time to Write Batch (ms), Avg. Time Between Batches Base and Avg. Time Between Batches (ms).*

```
SELECT * FROM sys.dm_os_performance_counters
WHERE counter_name LIKE '%Average Wait Time%' AND instance_name = 'database'
```

Results | Messages

	object_name	counter_name	instance_name	cntr_value	cntr_type
1	MSSQL$SQLEXPRESS:Locks	Average Wait Time (ms)	Database	61142	1073874176
2	MSSQL$SQLEXPRESS:Locks	Average Wait Time Base	Database	75	1073939712

Average Wait Time (ms) for the interval between these two measurements is (53736 ms -52939 ms)/(23-18) = 797 ms / 5 = 159.4 ms

Some of the counters are: *Average Wait Time (ms), Average Latch Wait Time (ms), Update conflict ratio, Avg. Length of Batched Writes, Avg. Time to Write Batch (ms), Avg. Time Between Batches (ms).*

4) PERF_COUNTER_BULK_COUNT

The *cntr_types* column value for the PERF_COUNTER_BULK_COUNT counter type is 272696576 (0x10410500 – in hexadecimal). This counter value represents a rate metric. The **cntr_value** is cumulative. The value is obtained by taking two samples of the PERF_COUNTER_BULK_COUNT value. The difference between the sample values is divided by the number of seconds between the samples. This provides the value per second rate. It is important to know how long the sample period is, otherwise, one would not be able to calculate the value per second (usually a 5-minute period is used).

The formula for the current metric value is A2-A1)/(T2-T1), where A1 and A2 are the values of the monitored PERF_COUNTER_BULK_COUNT counter taken at sample times T1 and T2, and T1 and T2 are the times when the sample values are taken.

Sample 1

ms_ticks	object_name	counter_name	instance_name	cntr_value	cntr_type
488754390	MSSQL$SQLSVR:Databases	Transactions/sec	AdvWrks	1566	272696576

Sample 2

ms_ticks	object_name	counter_name	instance_name	cntr_value	cntr_type
488755468	MSSQL$SQLSVR:Databases	Transactions/sec	AdvWrks	2055	272696576

The value for Transactions/sec for the interval = (Value2 - Value1) / (seconds between samples) = (Value2 - Value1) / ((ms_value2 - ms_value1) / 1000) = (2055 - 1566) / ((488755468-488754390) / 1000) = 489 transactions/sec

The DELAY T-SQL method with 10 seconds

```
        DECLARE @PageLookups1 BIGINT;
SELECT @PageLookups1 = cntr_value
FROM sys.dm_os_performance_counters
WHERE counter_name = 'Page lookups/sec';
        WAITFOR DELAY '00:00:10';
SELECT (cntr_value - @PageLookups1) / 10 AS 'Page lookups/sec'
FROM sys.dm_os_performance_counters
WHERE counter_name = 'Page lookups/sec';
```

Results | Messages

	Page lookups/sec
1	1113

Another method is to use get the *ms_ticks* value (number of milliseconds since the machine was started) from the *sys.dm_os_sys_info* Dynamic Management View at the same time when the counter values are taken.

```
        SELECT ms_ticks FROM sys.dm_os_sys_info;
SELECT * FROM sys.dm_os_performance_counters
WHERE counter_name = 'Page lookups/sec';
```

Results | Messages

	ms_ticks
1	28403367

	object_name	counter_name	instance_name	cntr_value	cntr_type
1	MSSQL$SQLEXPRESS:Buffer Manager	Page lookups/sec		7805332	272696576

Based on the values obtained, the Page lookups/sec value is calculated as: Page lookups/sec = (854,521 – 852,433)/(621,366,686-621,303,043) = 2,088 / 63,643 ms = 2,088/63 sec = 32.1 /sec
The Getdate () statement and any other method for determining time difference can be used
Some of the counters are: *Page lookups/sec, Free list stalls/sec, Lazy writes/sec, Page reads/sec, Page writes/sec, Logins/sec.*

5) PERF_COUNTER_LARGE_RAWCOUNT

The *cntr_types* column value for the PERF_COUNTER_LARGE_RAWCOUNT counter type is 65792 (0x00010100 – in hexadecimal). This counter value shows the last observed value directly, not the average value. It is usually used to monitor object counts (if one is monitoring a counter type 65792, the value got in the counter_value column when query the view, is the current value of the counter; no additional calculation is required).

object_name	counter_name	instance_name	cntr_value	cntr_type
MSSQL$SQLSVR:Buffer Manager	Total pages		5504	65792

The value of the counter MSSQL$SQLSVR:Buffer Manager\Total pages = 5504.

```
-- To find out the current values for the Buffer Manager Page life expectancy
SELECT * FROM sys.dm_os_performance_counters
WHERE counter_name = 'Page Life expectancy' AND object_name LIKE '%buffer manager%';
```

Results | Messages

	object_name	counter_name	instance_name	cntr_value	cntr_type
1	MSSQL$SQLEXPRESS:Buffer Manager	Page life expectancy		1640	65792

The *Page life expectancy* is 47, no additional calculation is needed
Some of the counters are: *General Statistics User connections, Buffer Manager Page life expectancy and Database pages, Databases – Data and Log file size (KB), Log file used size (KB), Percent Log used, Memory Manager – Free Memory (KB).*

sys.dm_os_performance_counters - is a system Dynamic Management View (DMV) that returns one row for each SQL Server performance counter. It is useful for obtaining information about current performance counter values. These counter values are also shown in Windows Performance Monitor. The permission needed to query this view is VIEW SERVER STATE.

```sql
-- find the SQL Server performance counters
that can be tracked
SELECT DISTINCT [object_name]
FROM sys.dm_os_performance_counters
```

Results | Messages

	object_name
1	MSSQL$SQLEXPRESS:Broker Statistics
2	MSSQL$SQLEXPRESS:Cursor Manager by Type
3	MSSQL$SQLEXPRESS:Plan Cache
4	MSSQL$SQLEXPRESS:Workload Group Stats
5	MSSQL$SQLEXPRESS:Access Methods
6	MSSQL$SQLEXPRESS:External Scripts
7	MSSQL$SQLEXPRESS:Columnstore
8	MSSQL$SQLEXPRESS:Exec Statistics
9	MSSQL$SQLEXPRESS:Broker TO Statistics
10	MSSQL$SQLEXPRESS:Database Replica
11	MSSQL$SQLEXPRESS:Broker Activation

Query executed successfully.

Some of the counters are: *Free Memory (KB), Lock Memory (KB), Memory Grants Pending, Target Server Memory (KB), Total Server Memory (KB).*

Some counters are repeated in different categories. In total there are 405 different counters. The information that is usually overlooked, but that requires attention is the *cntr_types* column. It defines the type of the counter and thus the method that should be used to calculate the current counter value.

References:

https://docs.microsoft.com/en-us/sql/relational-databases/system-dynamic-management-views/sys-dm-os-performance-counters-transact-sql?view=sql-server-2017

https://docs.microsoft.com/en-us/previous-versions/sql/sql-server-2012/ms187743(v=sql.110)

https://blogs.msdn.microsoft.com/psssql/2013/09/23/interpreting-the-counter-values-from-sys-dm_os_performance_counters/

https://www.sqlshack.com/troubleshooting-sql-server-issues-sys-dm_os_performance_counters/

https://simplesqlserver.com/2013/08/13/sys-dm_os_perfomance_counters-demystified/