

# SQL Server 2017 dynamic management views

Quest®

<b>Common language runtime</b>
Useful for understanding and troubleshooting CLR objects executing within SQL Server. CLR objects are cached for better performance after they are used and are not destroyed immediately. CLR objects are unloaded only when SQL Server comes under memory pressure.
<b>sys.dm_clr_appdomains</b>
Returns a row for each AppDomain, the unit of isolation for an application running in .NET, running on the server. SQL Server creates one AppDomain per database per owner so that all CLR objects are always executed in the same AppDomain.
<b>sys.dm_clr_loaded_assemblies</b>
Returns a row for each managed user assembly, i.e. managed code DLL files, loaded into the server address space.
<b>sys.dm_clr_properties</b>
Returns a row for each property of a CLR assembly available to the system, such as the CLR version, current state of the hosted CLR, or the CLR install directory.
<b>sys.dm_clr_tasks</b>
Returns a row for all currently running CLR tasks.

<b>Columnstore</b>
<b>sys.dm_column_store_object_pool</b>
Returns access counts and memory pool usage for columnstore index objects.
<b>sys.dm_column_store_row_group_operational_stats</b>
Returns current row-level I/O locking, and access metadata activity for compressed rowgroups in a columnstore index. Not relevant for in-memory columnstore index, can be used to identify rowgroups that are encountering significant I/O activity or hot spots.
<b>sys.dm_column_store_row_group_physical_stats</b>
Provides current rowgroup-level information about all of the columnstore indexes in the current database.

## Database

<b>sys.dm_db_database_page_allocations</b>
Displays the information as to how objects keep data in different pages and its allocation in the database. Syntax: sys.dm_db_database_page_allocations(@Databaseid, @Tableid, @Indexid, @Partitionid, @Mode)
<b>sys.dm_db_file_space_usage</b>
Returns space usage information for each file in the database.
<b>sys.dm_db_log_info</b>
Returns virtual log file (VLF) information on the transaction log of a given database.
<b>sys.dm_db_log_stats</b>
Returns summary level attributes and information on transaction log files of databases. Can be used instead of DBCC commands for monitoring and diagnostics of transaction log health.
To determine the database where log is not truncating and the reason for that:
<pre>select s.database_id, name, total_log_size_mb, active_log_size_mb, active_log_size_mb*100.0/total_log_size_mb as pct_used, log_truncr-- from sys.databases s CROSS APPLY sys.dm_db_log_stats(s.database_id) tsv where tsv.truncation_log_trunc_reason != 'NOTHING'</pre>
<b>sys.dm_db_partition_stats</b>
Returns one row per partition in the database, showing page and row-count information for every partition in the current database, including the space used to store and manage in-row data, LOB data, and row overflow data for all partitions in a database.
<b>sys.dm_db_perf_stats_sku_features</b>
Certain features of the SQL Server Database Engine affect the way that the Database Engine stores information in the database files. These features are restricted to specific editions of SQL Server, and prevent moving a database that contains these features to an edition of SQL Server that does not support them. To view a list of all edition-specific features that are enabled in the current database, use the sys.dm_db_perf_stats_sku_features dynamic management view.
<b>sys.dm_db_rds_migration_stats</b>
Returns information about batch of migrated data from each Stretch-enabled table on the local instance of SQL Server in the current database context.
<b>sys.dm_db_rds_schema_update_stats</b>
Returns information about each schema update task for the remote data archive of each Stretch-enabled table in the current database.
<b>sys.dm_db_session_space_usage</b>
Returns the number of pages allocated and deallocated by each session, user or internal, in tempdb.
<b>sys.dm_db_stats_histogram</b>
Returns the statistics histogram for the specified database object (table or indexed view) in the current SQL Server database. Can be used instead of DBCC SHOW_STATISTICS WITH HISTOGRAM
<b>sys.dm_db_stats_properties</b>
Returns properties of statistics for the specified database object (table or indexed view) in the current SQL Server database. Syntax: sys.dm_db_stats_properties (object_id, stats_id)
<b>sys.dm_db_task_space_usage</b>
Returns page allocation and deallocation activity by task for tempdb, excluding IAM pages.
<b>sys.dm_db_uncontained_entities</b>
Displays all uncontained objects used in the database. Uncontained objects are objects that cross the database boundary in a contained database; however, this view is accessible from both contained and non-contained databases. If sys.dm_db_uncontained_entities is empty your database does not use any uncontained entities.

To determine the amount of space used by internal and user objects in tempdb:

```
SELECT SUM(internal_object_reserved_page_count) AS internal_pages, (SUM(internal_object_reserved_page_count)*1.0/128) AS internal_space_MB, SUM(user_object_reserved_page_count) AS user_pages, (SUM(user_object_reserved_page_count)*1.0/128) AS user_space_MB FROM sys.dm_db_file_space_usage;
```

## Transaction

<b>sys.dm_tran_active_snapshot_database_transactions</b>
Returns a virtual table for each active transaction that could potentially generate access row versions. Each transaction it returns has an XSN (transaction sequence number) which is given to any transaction that accesses a version store.
<b>sys.dm_tran_active_transactions</b>
Returns information about transactions (such as the state of the transaction, when it began, whether it is read-only or not, and so forth) executing within the SQL Server instance.
<b>sys.dm_tran_current_snapshot</b>
Returns a virtual table of all active XSNs currently running when the current snapshot transaction starts. Returns no rows if the current transaction is not a snapshot transaction.
<b>sys.dm_tran_current_transaction</b>
Returns a single row that displays the state and snapshot sequence information of a transaction in the current session. Not useful for transactions running in isolation levels other than snapshot isolation level.
<b>sys.dm_tran_database_transactions</b>
Returns information about transactions at the database level, especially transaction log information such as the log sequence number (LSN), log bytes used, and transaction type.
<b>sys.dm_tran_locks</b>
Returns information about currently active requests to the lock manager, broken into a resource group and a request group. The request status may be active, convert, or may be waiting (wait) and includes details such as the resource on where the lock request wants a log (for the resource group) or the lock request itself (for the request group).
<b>sys.dm_tran_session_transactions</b>
Returns correlation information for associated transactions and sessions, especially useful for monitoring distributed transactions.
<b>sys.dm_tran_version_generators</b>
Returns a virtual table for the objects that are producing the most versions in the version store, as found in sys.dm_tran_version_store system view. Use with caution. This is a very resource intensive DMV.
<b>sys.dm_tran_transactions_snapshot</b>
Returns information about active transactions when each snapshot transaction starts. Using this, you can find how many snapshot transactions are currently active and identify any data modifications that are ignored by any given snapshot transaction.
<b>sys.dm_tran_version_store</b>
Returns each versioned record, both its XSN and its binary version sequence number. Because the version store can be quite large, this DMV can be very resource intensive.

<b>Transaction (continued)</b>
<b>sys.dm_tran_version_store_space_usage</b>
Returns aggregated information about overall space in tempdb used by version store records for each database.
To find all active distributed transactions on the SQL Server instance
<pre>SELECT FROM sys.dm_tran_active_transactions WHERE transaction_type = 4;</pre>
To seeing blocked and blocking transactions on the server:
<pre>SELECT t.resource_type, t.resource_database_id, t.resource_associated_entity_id, t.request_mode, t.request_session_id, t.blocking_session_id FROM sys.dm_tran_locks as tl INNER JOIN sys.dm_os_waiting_tasks AS w ON t.lock_owner_address = w.resource_address;</pre>

## Execution & Thread

These DMVs and functions show what activity is executing on the server. The DMV sys.dm_exec_query_plan is used internally by SQL Server 2005.
<b>sys.dm_exec_background_job_queue</b>
Returns a row for each query processor job that is scheduled for asynchronous, i.e. background, execution. In SQL Server 2005, this will show update statistics jobs.
<b>sys.dm_exec_background_job_queue_stats</b>
Returns a row that provides aggregate statistics for each query processor job submitted for background execution.
<b>sys.dm_exec_exec_cached_plans</b>
Similar to syscacheobjects in SQL Server 2000, Returns a row for each query plan that is held in procedure cache, and the corresponding information like the cached query plans, the cached query text, the amount of memory taken by eachd plans, and the reuse count of the cached plans.
<b>sys.dm_exec_compute_nodes</b>
Lists all compute nodes in a scale-out PolyBase cluster data management env with their role, name and IP address.
<b>sys.dm_exec_compute_node_errors</b>
Returns errors that occur on PolyBase compute nodes
<b>sys.dm_exec_compute_node_stats</b>
Returns information about the performance and status of all PolyBase compute nodes
<b>sys.dm_exec_connections</b>
Returns server-level information about a connection to SQL Server, such as the client network address, client TCP port, and client authorization scheme.
<b>sys.dm_exec_cursors</b>
Returns information about cursors that are open in one or more databases on the server. Syntax: dm_exec_cursors (session_id   0)
<b>sys.dm_exec_describe_first_result_set</b>
This dynamic management function takes a Transact-SQL statement as a parameter and describes the metadata of the first result set for the statement. Syntax: sys.dm_exec_describe_first_result_set([sql_statement, @include_browse_information])
<b>sys.dm_exec_describe_first_result_set_for_object</b>
This dynamic management function takes an object_id as a parameter and describes the first result metadata for the module with that ID. The @object_id specified can be the ID of a Transact-SQL stored procedure or a Transact-SQL trigger. If it is the ID of any other object (such as a view, table, function, or CLR procedure), an error will be specified in the error columns of the result. Syntax: sys.dm_exec_describe_first_result_set_for_object ( @object_id, @include_browse_information )
<b>sys.dm_exec_distributed_requests</b>
Returns information about all requests currently or recently active in PolyBase engines. Use 'global' execution_id to track the distributed steps of all operations associated with one particular request and operates across all compute nodes.

To find the longest running queries:
<pre>SELECT top (100) execution_id, st.text, edr.total_elapsed_time FROM sys.dm_exec_distributed_requests edr CROSS APPLY sys.dm_exec_sql_text(sql_handle) st ORDER BY total_elapsed_time DESC;</pre>
<b>sys.dm_exec_distributed_request_stats</b>
Returns information about all steps that compose a given PolyBase request or query
<b>sys.dm_exec_distributed_sql_requests</b>
Returns information about all SQL query distributions as part of a SQL step in the query. The view shows the data for the last 1000 requests as well as active requests
<b>sys.dm_exec_dms_services</b>
Returns information about all of the Data Movement Service(DMS) services running on the PolyBase compute nodes.
<b>sys.dm_exec_dms_workers</b>
Returns information about all workers completing DMS steps. The view shows the data for the last 1000 requests as well as active requests.
<b>sys.dm_exec_external_operations</b>
Returns information about external PolyBase operations of type Hadoop operation.
<b>sys.dm_exec_external_work</b>
Returns information about the activity per worker, on each compute node. Use it to identify and monitor the workers spun up to communicate with the external data sources (like Hadoop).
<b>sys.dm_exec_function_stats</b>
Returns aggregate performance statistics for cached functions. The view returns one row for each cached function, and the lifetime of the row is as long as the stored procedure remains cached. When a stored procedure is removed from the cache, the corresponding row is eliminated from this view. Returns information about scalar functions, including in-memory functions and CLR scalar functions, but not about table valued functions.
<b>sys.dm_exec_input_buffer</b>
Returns the text of the statements (batch) submitted to an instance of SQL Server and executed in a specific session.
<b>sys.dm_exec_plan_attributes</b>
Returns one row per plan attribute for a specific plan identified by its plan handle, such as information like the cache key values or the number of current simultaneous executions of the plan. Syntax: sys.dm_exec_plan_attributes (plan_handle)
<b>sys.dm_exec_procedure_stats</b>
Returns aggregate performance statistics for cached stored procedures. The view returns one row for each cached stored procedure plan, and the lifetime of the row is as long as the stored procedure remains cached. When a stored procedure is removed from the cache, the corresponding row is eliminated from this view. At the same time, a Performance Statistics SQL trace event is raised similar to sys.dm_exec_query_stats.
<b>sys.dm_exec_query_memory_grants</b>
Returns information about queries that have acquired memory grants or that still require a memory grant to execute. This DMV is useful for determining query timeouts, since only queries that have to wait on a memory grant will appear in this view.
<b>sys.dm_exec_query_optimizer_info</b>
Returns detailed statistics about the operation of the SQL Server query optimizer, such as the total number of optimizations, the elapsed time value, or sophisticated assessments like comparing the query optimization cost of the current workload to that of a tuned workload. Some counters shown by this DMV are for internal use only.
<b>sys.dm_exec_query_parallel_workers</b>
Returns worker availability information per NUMA node
<b>sys.dm_exec_query_plan</b>
Returns a given query's Showplan output in XML format, as specified by the plan handle. Syntax: sys.dm_exec_query_plan (plan_handle )
<b>sys.dm_exec_query_resource_semaphores</b>
Returns two rows, one for the regular resource semaphore and the other for the small query resource semaphore, information about the current query resource semaphore status. When used with sys.dm_os_memory_clerks, this DMV provides a complete picture of memory status information about query executions and allows you to determine whether the system can access enough memory.
<b>sys.dm_exec_query_stats</b>
Returns one row per query statement within a cached plan, detailing aggregated performance statistics for cached query plans. Because the information is aggregated, you may sometimes get better information by running this DMV.
<b>sys.dm_exec_requests</b>
Returns one row for each request executing within SQL Server, but does not contain information for each task that executes outside of SQL Server, such as distributed queries or extended stored procedures.
<b>sys.dm_exec_session_wait_stats</b>
Returns information about all the waits encountered by threads that executed for each session. Similar to sys.dm_os_wait_stats but includes the session level. The data in this view is aggregated as long as the session is open. It will reset if the session is closed or reset (with connection pooling).
<b>sys.dm_exec_sessions</b>
Returns one row per authenticated session, including both active users and internal tasks, running anywhere on SQL Server.

<b>Execution &amp; Thread (continued)</b>
<b>sys.dm_exec_sql_text</b>
Returns the text of the SQL batch that is identified by the specified sql_handle. Syntax: sys.dm_exec_sql_text(sql_text_handle   plan_handle)
<b>sys.dm_exec_text_query_plan</b>
Returns Showplan output in text format for a given TSQL batch or a statement within the batch. It's similar sys.dm_exec_query_plan, except that it returns data as text, is not limited in size, and may be specific to an individual statement within a batch.
<b>sys.dm_exec_trigger_stats</b>
Returns aggregate performance statistics for cached triggers. The view contains one row per trigger, and the lifetime of the row is as long as the trigger remains cached. When a trigger is removed from the cache, the corresponding row is eliminated from this view.
<b>sys.dm_exec_xml_handles</b>
Returns information about one or all active handles that opened with the sp_xml_preparedocument system stored procedure. Syntax: dm_exec_xml_handles (session_id   0)

## SQL Server OS

The following SQL Server Operating System related dynamic management views are for internal use only: sys.dm_os_function_symbolic_name, sys.dm_os_ring_buffers, sys.dm_os_memory_allocations, sys.dm_os_sublatches, and sys.dm_os_worker_local_storage. The remaining DMVs are user-accessible and contain a panoply of information to help you understand and manage the SQL Server Operating System (SLOS) responsible for governing resources within the system.
<b>sys.dm_os_buffer_descriptors</b>
Returns one row for each data page that are in the SQL Server buffer cache, excluding free, stolen, or erroneously read pages. SQL Server 2005 tracks anything put into the buffer cache using buffer descriptors. You can easily scope the results to a particular database and one or all of the objects within the database.
<b>sys.dm_os_buffer_pool_extension_configuration</b>
Returns configuration information about the buffer pool extension in SQL Server.
<b>sys.dm_os_child_instances</b>
Returns a row showing the state and pipe name of each user instance spawned from the parent server instance.
<b>sys.dm_os_cluster_nodes</b>
Returns a row for each node in the failover cluster instance configuration or an empty rowset if the server is not configured as in a failover cluster
<b>sys.dm_os_dispatcher_pools</b>
Returns information about session dispatcher pools. Dispatcher pools are thread pools used by system components to perform background processing.
<b>sys.dm_os_enumerate_fixed_drives</b>
Returns free space information on all drives mounted on the SQL Server machine. Works for both Windows and Linux installation
<b>sys.dm_os_host_info</b>
Returns one row that displays operating system version information. Information is similar to what is available in osVersion
<b>sys.dm_os_hosts</b>
Returns a row for each host (such as an OLE DB provider) currently registered in an instance of SQL Server and each resource used by these hosts.
<b>sys.dm_os_latch_stats</b>
Returns information about all latch waits, organized by latch class. Latches are held on every lightweight and transient system resources, such as an address in memory. This DMV is useful for troubleshooting latch contention. It does not track latch usage where the latch was granted immediately or failed immediately.
<b>sys.dm_os_loaded_modules</b>
Returns a row for each module loaded into the server.
<b>sys.dm_os_memory_broker_clerks</b>
Returns the set of all memory clerks that are currently active in the instance of SQL Server.
<b>sys.dm_os_memory_brokers</b>
Alerts that are sent internal to SQL Server use the SQL Server memory manager.
<b>sys.dm_os_memory_cache_clock_hands</b>
Returns the status (suspended or running) of each hand for a specific cache clock. The process used to age items out of a cache is called the cache clock and each clock can have one or more processes (called a hand) to sweep the cache clean.
<b>sys.dm_os_memory_cache_counters</b>
Returns an overview of the cache health, including run-time information about cache entries allocated, the source of memory for the cache entries, and how they are used.
<b>sys.dm_os_memory_cache_entries</b>
Returns information, such as statistics, for all entries in the various caches. This DMV is useful for linking cache entries back to their associated database objects.
<b>sys.dm_os_memory_cache_hash_buckets</b>
Returns information about each active cache in the instance of SQL Server, such as the type of cache, the number and type of hash buckets, the length of time the hash buckets have been in use, etc.
<b>sys.dm_os_memory_clerks</b>
Returns all currently active memory clerks within SQL Server. A memory clerk is the primary means for allocating memory to the various users of SQL Server.
<b>sys.dm_os_memory_nodes</b>
Allocations that are internal to SQL Server use the SQL Server memory manager.
<b>sys.dm_os_memory_objects</b>
Returns all memory objects that are currently allocated by SQL Server. Memory objects are more granular than memory clerks and are used by internal SQL Server processes and components, but not users like memory clerks. This DMV is ideal for analyzing memory usage and identifying memory leaks.
<b>sys.dm_os_memory_pools</b>
Returns a row for each memory pool object store in the SQL Server instance. Memory pool objects are certain homogeneous, equally important, statistical types of data. This information is sometimes useful for identifying bad caching behavior.
<b>sys.dm_os_nodes</b>
Returns information about the currently active nodes on the instance. Nodes are created by SQL OS to mimic hardware processor locality and can be altered by SQL OS using soft-NUMA techniques.
<b>sys.dm_os_process_memory</b>
Returns information providing a complete picture of the process memory address space in kilobytes, including things like the page_fault_count, amount of virtual address space available and committed, process physical and virtual memory, and so forth.
<b>sys.dm_os_performance_counters</b>
Returns a row per performance counter, the same counters as in Windows PerfMon, maintained by the OS. To calculate a discrete value for the various per-second counters, you must sample two discrete values and then subtract the earlier cumulative per-second value from the later.
<b>sys.dm_os_schedulers</b>
Returns one row per scheduler for schedulers mapped to an individual processor. Active-worker scheduler threads are those devoted to the regular user-controlled work of the server, such as queries and SIS jobs, while a variety of system and hidden schedulers may be active on the system at any time. This DMV is very useful for monitoring the thread scheduler and to find runaway tasks.
<b>sys.dm_os_spinlock_stats</b>
List of all spinlocks on the system. A spinlock is another lightweight synchronization primitive used to control access to certain data structures in the engine.
<b>sys.dm_os_stats</b>
Used internally by SQL Server track debug data, such as outstanding allocations, and to validate the logic in a component that assumes that a certain call was made.
<b>sys.dm_os_sys_info</b>
Returns a miscellaneous set of useful information about the computer, such as the hyperthread ratio, the max worker count, and other resources used by and available to SQL Server.
<b>sys.dm_os_sys_memory</b>
Returns a complete picture of memory at the operating system level, including information about total and available physical memory, total and available page memory, system cache, kernel space and so forth.
<b>sys.dm_os_tasks</b>
Returns one row for each OS task that is active in the instance of SQL Server.
<b>sys.dm_os_threads</b>
Returns a row for each SLOS threads running under the SQL Server process.
<b>sys.dm_os_volume_stats</b>
Returns information about the operating system volume (directory) on which the specified databases and files are stored in SQL Server 2012 syntax: sys.dm_os_volume_stats (database_id, file_id)
<b>sys.dm_os_wait_stats</b>
Returns information about waits encountered by currently executing threads. There are a limited number of reasons why a thread might be forced to wait before it can continue its work. This view is used by the SQL Server Books On-Line for more information about all possible wait states. This is an excellent DMV to use to diagnose performance issues with SQL Server and also with specific queries and batches because it records any time anything within SQL Server has to wait.
<b>sys.dm_os_waiting_tasks</b>
Returns information about the wait queue of SLOS tasks that are waiting on some resource, such as blocking and latch contention.

<b>SQL Server OS (continued)</b>
<b>sys.dm_os_windows_info</b>
Returns one row that displays Windows operating system version information.
<b>sys.dm_os_workers</b>
Returns a row for every worker in the system.
<b>sys.dm_os_subscriptions</b>
Returns information about the active query notifications subscriptions in the server.
<b>sys.dm_server_memory_dumps</b>
Returns one row for each memory dump file generated by the SQL Server Database Engine.
<b>sys.dm_server_registry</b>
Returns configuration and installation information that is stored in the Windows registry for the current instance of SQL Server
<b>sys.dm_server_services</b>
Returns information about the SQL Server, Full-Text, and SQL Server Agent services in the current instance of SQL Server

## Query Store

The SQL Server Query Store feature, available since version 2016, provides insight on query plan choice and performance for each database it is enabled for. Following DMVs can be used to explore the information gathered.
<b>sys.query_store_plan</b>
Returns information about each execution plan associated with a query
<b>sys.query_store_query</b>
Returns information about a query and its associated overall aggregated runtime execution statistics.
<b>sys.query_store_query_text</b>
Contains the Transact-SQL text and the SQL handle of the query
<b>sys.query_store_runtime_stats</b>
Returns information about the runtime execution statistics for a query plan. Need to aggregate metrics to get full details on specific plan for each interval.
To find out all queries that did not finished successfully in the last week:
<pre>select qt.query_sql_text, q.query_hash, q.query_id, qt.query_text_id, p.plan_id, sum(count_executions) as executions from sys.query_store_query_text AS qt JOIN sys.query_store_query AS q ON qt.query_text_id = q.query_text_id JOIN sys.query_store_plan AS p ON q.query_id = p.query_id JOIN sys.query_store_runtime_stats AS rs ON p.plan_id = rs.plan_id JOIN sys.query_store_runtime_stats_interval si ON rs.runtime_stats_interval_id = si.runtime_stats_interval_id where execution_type &lt;&gt; 0 --only aborted execution state and rs.last_execution_time &gt;= DATETIME(DAY, -7, GETUTCDATE()) group by qt.query_sql_text, q.query_hash, q.query_id, qt.query_text_id, p.plan_id</pre>
<b>sys.query_store_wait_stats</b>
Returns information about the waits encountered by a query plan. Waits are reported by categories. Each category represents several wait types that have a similar troubleshooting process. The wait categories are:
<b>sys.sys.dm_db_tuning_recommendations</b>
Returns information about tuning recommendations identified by the database engine. Recommendations are only gathered if Query Store is turned on and is in write state. Recommendations are kept until SQL Server is restarted.

## Security

<b>sys.asymmetric_keys</b>
Returns a row for each asymmetric key, containing (by default) both a public and private key. (The private key is protected by the database master key)
<b>sys.certificates</b>
Returns a row for each certificate in the database. A certificate is a database-level security object loaded from a file or an assembly which follows the X.509 standard.
<b>sys.credentials</b>
Returns a row for each credential in the instance. A credential is a record containing authentication information needed to connect to resource external to SQL Server, including a Windows user and password.
<b>sys.crypt_properties</b>
Returns a row for each cryptographic property associated with each symmetric or asymmetric key on the instance.
<b>sys.database_permissions</b>
Returns a row for each permission or counter-permission in the database that is, permissions on a column that are different from the corresponding higher-level, object permission).
<b>sys.database_principals</b>
Returns a row for each principal in the database. A principal is an entity (such as a Windows login, a SQL Server login, or an application role)
<b>sys.audit_actions</b>
Returns a row for every audit action that can be reported in the audit log and every action group that can be configured as part of SQL Server Audit.
<b>sys.audit_class_type_map</b>
Returns a table that maps the class_type field in the audit log to the class_desc field in sys.dm_audit_actions.
<b>sys.dm_cryptographic_provider_keys</b>
Returns information about the keys provided by an Extensible Key Management (EKM) provider. Syntax: sys.dm_cryptographic_provider_keys (provider_id )
<b>sys.dm_cryptographic_provider_sessions</b>
Returns information about open sessions for a cryptographic provider. Syntax: sys.dm_cryptographic_provider_sessions (session_identifier)
<b>sys.dm_database_encryption_keys</b>
Returns information about the encryption state of a database and its associated database encryption keys.
<b>sys.dm_cryptographic_provider_algorithms</b>
Returns the algorithms supported by an Extensible Key Management (EKM) provider. Syntax: sys.dm_cryptographic_provider_algorithms (provider_id )
<b>sys.dm_cryptographic_provider_properties</b>
Returns information about registered cryptographic providers.
<b>sys.dm_server_audit_stats</b>
Returns a row for each server audit, indicating the audit's current state.
<b>sys.key_encryptations</b>
Returns a row for each symmetric key encryption that was created using the statement CREATE SYMMETRIC KEY with ENCRYPTION BY.
<b>sys.login_token</b>
Returns a row for each server principal (such as a Windows login, SQL Server login, or application role) that is part of the login token.
<b>sys.master_key_passwords</b>
Returns a row for each database master key password (used to protect the master keys kept in the credential store created with the system stored procedure sp_control_dbmasterkey_password stored procedure).
<b>sys.openkeys</b>
Returns a row for each encryption key that is open in the current session.
<b>sys.securable_classes</b>
Returns a list of all securable classes on the instance.
<b>sys.server_permissions</b>
Returns a row for each server permission granted, including servers, server principals and endpoints.
<b>sys.server_principals</b>
Returns a row for each server-level principal, such as Windows and SQL Server logins, as well as logins mapped to certificates and asymmetric keys.
<b>sys.server_role_members</b>
Returns a row for each member of each fixed server role on the instance.
<b>sys.sql_logins</b>
Returns a row for each SQL login on the instance.
<b>sys.system_components_surface_area_configuration</b>
Returns a row for each executable system object, like a stored procedure or user-defined function, that can be enabled or disabled by a surface area configuration component.
<b>sys.symmetric_keys</b>
Returns a row for each symmetric key created using the statement CREATE SYMMETRIC KEY.
<b>sys.user_token</b>
Returns a row for each database principal (such as a Windows login, SQL Server login, or application role) that is part of the user token.

<b>Change Data Capture</b>
<b>sys.dm_cdc_errors</b>
Returns one row for each error encountered during the change data capture log scan session.
<b>sys.dm_cdc_log_scan_sessions</b>
Returns one row for each log scan session in the current database. The last row returned represents the current session. This view can be used to return either status information about the current log scan session, or aggregated information about all sessions since the SQL Server instance was last started.
<b>sys.dm_repl_traninfo</b>
Returns information on each capture transaction that was replicated, or whose data was modified.
<b>sys.dm_tran_commit_table</b>
Displays one row for each transaction committed for a table that is tracked by the SQL Server change tracking mechanism.

## SQL Server extended events

SQL Server 2008 introduces the concept of extended events, a general event-handling system that can correlate data from SQL Server, the OS, and applications using Event-Tracing for Windows.
<b>sys.dm_xe_map_values</b>
Returns the extended event mapping of internal number keys as human-readable text.
<b>sys.dm_xe_object_columns</b>
Returns the extended event schema information for specified object. The object_name must exist in sys.dm_xe_objects.
<b>sys.dm_xe_objects</b>
Returns a row for each object exposed by an event package, including events, actions, targets, predicates, and types.
<b>sys.dm_xe_packages</b>
Returns a row for each package registered with the extended events engine.
<b>sys.dm_xe_session_event_actions</b>
Returns a row for each event session action, including the number of times the action has been fired and its total run time.
<b>sys.dm_xe_session_events</b>
Returns a row for each session event, its memory address, package GUID, and predicate.
<b>sys.dm_xe_session_object_columns</b>
Returns a row for each configuration value assigned to objects that are bound to a given session.
<b>sys.dm_xe_session_targets</b>
Returns information about session targets, such as the number of times the target has been executed for the session and the total time that the target has been executing.
<b>sys.dm_xe_sessions</b>
Returns information about the active extended event sessions (e.g. events, actions, and targets) currently active in memory.

## Service broker

<b>dm_broker_activated_tasks</b>
Returns a row for each stored procedure activated by Service Broker.
<b>dm_broker_connections</b>
Returns a row for each Service Broker network connection.
<b>dm_broker_forwards_messages</b>
Returns a row for each Service Broker message that an instance of SQL Server is in the process of forwarding.
<b>dm_broker_queue_monitors</b>
Returns a row for each queue monitor in the instance. A queue monitor manages activation for a queue.

## Index

<b>sys.dm_db_index_operational_stats</b>
Returns current low-level I/O, locking, latching, and access method activity for one or all databases, and one or all tables, indexes and partitions within each database. Syntax: sys.dm_db_index_operational_stats ( ( database_id   NULL   0   DEFAULT ) , ( object_id   NULL   0   DEFAULT ) , ( index_id   0   NULL   1   DEFAULT ) , ( partition_number   NULL   0   DEFAULT ) )
<b>sys.dm_db_index_physical_stats</b>
Returns size and fragmentation information for the data pages and index pages for one or all databases, and one or all tables, indexes and partitions within each database. You may also specify a scanning mode to speed processing. You will usually get multiple rows even if specifying a specific index on a specific table in a specific database. The DMV returns one row for indexes for each level of the B-tree in each partition. One row is returned for LOB data and two overflow data, if they exist in each partition. One row is returned for keeps for the IN_ROW_DATA allocation unit of each partition. Syntax: sys.dm_db_index_physical_stats ( ( database_id   NULL   0   DEFAULT ) , ( object_id   NULL   0   DEFAULT ) , ( index_id   NULL   0   1   DEFAULT ) , ( partition_number   NULL   0   DEFAULT ) , ( mode   NULL   DEFAULT ) )
<b>sys.dm_index_usage_stats</b>
Returns a count of different types of index operations (such as seeks, scans, lookups, and updates) and the last time each operation was performed.
<b>sys.dm_missing_index_columns</b>
Returns information about columns that are missing an index, which can then be used to create indexes on columns. Syntax: sys.dm_missing_index_columns (indexname   indexhandle)
<b>sys.dm_missing_index_details</b>
Returns detailed information about missing indexes. Used in conjunction with the other sys.xxx_missing_index_... DMVs and functions.
<b>sys.dm_missing_index_group_stats</b>
Returns summary information about groups of missing indexes. Used in conjunction with the other sys.xxx_missing_index_... DMVs and functions.
<b>sys.dm_missing_index_groups</b>
Returns information showing which specific missing indexes are contained in a specific missing index group. Used in conjunction with the other sys.xxx_missing_index_... DMVs and functions.
<b>sys.dm_indexes_disabled_on_compatibility_level_change</b>
Lists the indexes and constraints that will be disabled as a result of changing compatibility level.

## I/O

The DMVs in this section help you monitor I/O activity on the instance.
<b>sys.dm_io_backup_tapes</b>
Returns the list of tape devices and the status of mount requests for backups.
<b>sys.dm_io_cluster_shared_drives</b>
If the current server instance is a clustered server, this view returns the drive name of each of the shared drives; otherwise it returns an empty row set.
<b>sys.dm_io_pending_io_requests</b>
Returns a row for each pending I/O request in SQL Server.
<b>sys.dm_io_virtual_file_stats</b>
Returns I/O statistics for data and log files. Syntax: sys.dm_io_virtual_file_stats ( ( database_id   NULL ) , ( file_id   NULL ) )
<b>sys.dm_io_workers</b>
Returns a row for each I/O worker thread in the instance.
<b>sys.dm_io_referenced_entities</b>
Returns a row for each user-defined entity that is referenced by a server-level DDL trigger, a database-level DDL trigger, or a specific object in the current database context. (You can get server level information only when in the master database.) No dependency information is kept for rules, defaults, system objects and temporary tables and procedures. Syntax: sys.dm_io_referenced_entities ( [ schema_name   referencing_entity_name ] , [ referenced_class   'referencing_class' ] )
<b>sys.dm_io_referencing_entities</b>
Returns a row for each entity that references by name an object object you specify, whether that object is a partition, function, an XML schema, collection, a type, or a specific object in the current database context. (You can get server level information only when in the master database.) No dependency information is kept for rules, defaults, system objects and temporary tables and procedures. Syntax: sys.dm_io_referencing_entities ( [ schema_name   referenced_entity_name ] , [ referenced_class   'referenced_class' ] )
<b>sys.dm_io_referencing_entities</b>
Returns a row for each entity that references by name an object object you specify, whether that object is a partition, function, an XML schema, collection, a type, or a specific object in the current database context. (You can get server level information only when in the master database.) No dependency information is kept for rules, defaults, system objects and temporary tables and procedures. Syntax: sys.dm_io_referencing_entities ( [ schema_name   referenced_entity_name ] , [ referenced_class   'referenced_class' ] )

## Object

The DMVs in this section help you keep track of objects that are referenced by other objects in their definition, such as triggers, stored procedures, and functions.

<b>Resource governor</b>
<b>dm_resource_governor_resource_pool_volumes</b>
Returns statistics information about the current resource pool I/O for each disk volume.
<b>sys.dm_resource_governor_configuration</b>
Returns a row containing the current configuration state of the Resource Governor in memory. To see the same data, but stored in the long-term metadata, refer to sys_resource_governor_configuration.
<b>sys.dm_resource_governor_resource_pool_affinity</b>
Tracks resource pool affinity.
<b>sys.dm_resource_governor_resource_pools</b>
Returns a row for each current resource pool that is active in memory -- their states, current configurations, and resource pool statistics. To see the same data, but stored in the long-term metadata, refer to sys_resource_governor_resource_pools.
<b>sys.dm_resource_governor_workload_groups</b>
Returns a row for each current workload group that is active in memory, including configuration information and