**What's New (Database Engine)**

**SQL Server 2012**

[Other Versions](javascript:;)

Description: http://i.msdn.microsoft.com/Areas/Epx/Content/Images/ImageSprite.png

* [SQL Server 2008 R2](http://msdn.microsoft.com/en-us/library/bb510411(d=printer,v=sql.105).aspx)
* [SQL Server 2008](http://msdn.microsoft.com/en-us/library/bb510411(d=printer,v=sql.100).aspx)

This latest release of the SQL Server Database Engine introduces new features and enhancements that increase the power and productivity of architects, developers, and administrators who design, develop, and maintain data storage systems.

These are the areas in which the Database Engine has been enhanced.

|  |  |
| --- | --- |
| **Topic** | **Description** |
| [Availability Enhancements (Database Engine)](http://msdn.microsoft.com/en-us/library/c2da566b-9803-4794-a861-232a7dd01b2d) | Describes enhancements to high availability features. |
| [Manageability Enhancements (Database Engine)](http://msdn.microsoft.com/en-us/library/704f6493-fa87-4c3a-9aec-bb15bfe1a86b) | Describes enhancements to tools and monitoring features. |
| [Programmability Enhancements (Database Engine)](http://msdn.microsoft.com/en-us/library/09f0096e-ab95-4be0-8c01-f98753255747) | Describes programmability enhancements in the Database Engine. |
| [Scalability and Performance Enhancements (Database Engine)](http://msdn.microsoft.com/en-us/library/d8b36fb6-4bd5-4b18-af0a-fc3542fce2f5) | Describes scalability and performance enhancements in the Database Engine |
| [Security Enhancements (Database Engine)](http://msdn.microsoft.com/en-us/library/42919d6c-1fd4-466e-ac75-d579248c07df) | Describes security enhancements in the SQL Server Database Engine |
| [Resource Governor Enhancements (Database Engine)](http://msdn.microsoft.com/en-us/library/55ac5fa0-5013-4347-bf77-5894cb5d52e8) | Describes Resource Governor enhancements in the Database Engine. |
| [Backup and Restore Enhancements](http://msdn.microsoft.com/en-us/library/058f99bc-d6c5-442b-ab3c-4ba2a15c3d1d) | Describes enhancements to backup and restore features |
| [Microsoft.SqlServer.Dac](http://msdn.microsoft.com/en-us/library/microsoft.sqlserver.dac.aspx) and [Microsoft.SqlServer.Dac.Extensions](http://msdn.microsoft.com/en-us/library/microsoft.sqlserver.dac.extensions.aspx) | The [Microsoft.SqlServer.Dac](http://msdn.microsoft.com/en-us/library/microsoft.sqlserver.dac.aspx) namespace provides classes you can use in your code to perform operations on DACPAC and BACPAC packages. The [Microsoft.SqlServer.Dac.Extensions](http://msdn.microsoft.com/en-us/library/microsoft.sqlserver.dac.extensions.aspx) namespace provides classes containing extension methods you can use in your code to retrieve information from DACPAC and BACPAC packages or utilize the expanded functionality not currently present in the [Microsoft.SqlServer.Dac](http://msdn.microsoft.com/en-us/library/microsoft.sqlserver.dac.aspx) namespace. |

# Availability Enhancements (Database Engine)

**SQL Server 2012**

[Other Versions](javascript:;)

Description: http://i.msdn.microsoft.com/Areas/Epx/Content/Images/ImageSprite.png

* [SQL Server 2008 R2](http://msdn.microsoft.com/en-us/library/c2da566b-9803-4794-a861-232a7dd01b2d(d=printer,v=sql.105))
* [SQL Server 2008](http://msdn.microsoft.com/en-us/library/c2da566b-9803-4794-a861-232a7dd01b2d(d=printer,v=sql.100))

For protecting application databases in an enterprise environment from both planned and unplanned downtime, SQL Server 2012 introduces the AlwaysOn Availability Groups feature and a number of other high-availability enhancements.

In This Topic

* [New or Enhanced Features in SQL Server 2012 SP1](http://msdn.microsoft.com/en-us/library/c2da566b-9803-4794-a861-232a7dd01b2d(d=printer)#bkmk_SP1)
* [AlwaysOn SQL Server Failover Cluster Instances](http://msdn.microsoft.com/en-us/library/c2da566b-9803-4794-a861-232a7dd01b2d(d=printer)#FCI)
* [AlwaysOn Availability Groups](http://msdn.microsoft.com/en-us/library/c2da566b-9803-4794-a861-232a7dd01b2d(d=printer)#ssHADR)
* [Online Operations](http://msdn.microsoft.com/en-us/library/c2da566b-9803-4794-a861-232a7dd01b2d(d=printer)#OnlineOps)

[New or Enhanced Features in SQL Server 2012 SP1](javascript:void(0))

For a summary of all SQL Server 2012 SP1 enhanced features, see [New or Enhanced Features in SQL Server 2012 SP1](http://msdn.microsoft.com/en-us/library/bb500435#bkmk_sp1).

### Cross-Cluster Migration of AlwaysOn Availability Groups for OS Upgrade

SQL Server 2012 SP1 introduces support for cross-cluster migration of AlwaysOn Availability Groups for deployments to a new Windows Server Failover Clustering (WSFC) cluster. A cross-cluster migration moves one AlwaysOn availability group or a batch of availability groups to the new, destination WSFC cluster with minimal downtime. The cross-cluster migration process enables you to maintain your service level agreements (SLAs) when upgrading to a Windows Server 2012 cluster. SQL Server 2012 SP1 must be installed and enabled for AlwaysOn on the destination WSFC cluster. The success of a cross-cluster migration depends on thorough planning and preparation of the destination WSFC cluster.

For more information, see [Cross-Cluster Migration of AlwaysOn Availability Groups for OS Upgrade](http://msdn.microsoft.com/en-us/library/jj873730.aspx).

[AlwaysOn SQL Server Failover Cluster Instances](javascript:void(0))

* Multi-subnet failover clusters: A SQL Server multi-subnet failover cluster is a configuration where each failover cluster node is connected to a different subnet or different set of subnets. These subnets can be in the same location or in geographically dispersed sites. Clustering across geographically dispersed sites is sometimes referred to as Stretch clusters. As there is no shared storage that all the nodes can access, data should be replicated between the data storage on the multiple subnets. With data replication, there is more than one copy of the data available. Therefore, a multi-subnet failover cluster provides a disaster recovery solution in addition to high availability. For more information, see [SQL Server Multi-Subnet Clustering (SQL Server)](http://msdn.microsoft.com/en-us/library/ff878716).
* Flexible failover policy for cluster health detection: In a SQL Server failover cluster instance, only one node can own the cluster resource group at a given time. The client requests are served through this node for that failover cluster instance. In the case of a failure, the group ownership is moved to another node in the failover cluster. This process is called failover. The improved failure detection introduced in SQL Server 2012, and addition of failure condition level property allows you to configure a more flexible failover policy. For more information, see [Failover Policy for Failover Cluster Instances](http://msdn.microsoft.com/en-us/library/ff878664).
* Indirect checkpoints: The indirect checkpoints feature provides a database-specific alternative to automatic checkpoints, which are configured by a server property. Indirect checkpoints implements a new checkpointing algorithm for the Database Engine. This algorithm provides a more accurate guarantee of database recovery time in the event of a crash or a failover than is provided by automatic checkpoints. To ensure that database recovery does not exceed allowable downtime for a given database, you can specify the maximum allowable downtime for that database.

|  |
| --- |
| **Note** |
| An online transactional workload on a database that is configured for indirect checkpoints could experience performance degradation. |

* For more information, see [Database Checkpoints (SQL Server)](http://msdn.microsoft.com/en-us/library/ms189573).

[AlwaysOn Availability Groups](javascript:void(0))

Deploying AlwaysOn Availability Groups involves creating and configuring one or more availability groups. An availability group is a container that defines a set user databases (availability databases) to fail over as a single unit, and a set of availability replicas to host copies of each availability database. Each availability group requires at least two availability replicas: the primary replica and one secondary replica.

AlwaysOn Availability Groups provides a rich set of options that improve database availability and that enable improved resource use. The key components are as follows:

* Multiple secondary replicas: one primary replica and up to four secondary replicas. For more information, see [Overview of AlwaysOn Availability Groups (SQL Server)](http://msdn.microsoft.com/en-us/library/ff877884).

|  |
| --- |
| **Important** |
| Each availability replica must reside on a different node of a single Windows Server Failover Clustering (WSFC) cluster. For more information about prerequisites, restrictions, and recommendations for availability groups, see [Prerequisites, Restrictions, and Recommendations for AlwaysOn Availability Groups (SQL Server)](http://msdn.microsoft.com/en-us/library/ff878487). |

* Alternative availability modes: Asynchronous-commit mode and Synchronous-commit mode. For more information, see [Availability Modes (AlwaysOn Availability Groups)](http://msdn.microsoft.com/en-us/library/ff877931).
* Several failover modes: automatic failover, planned manual failover, and forced manual failover. For more information, see [Failover and Failover Modes (AlwaysOn Availability Groups)](http://msdn.microsoft.com/en-us/library/hh213151).
* Active secondary replicas, as follows:
  + Read-only access to the secondary replicas. For more information, see [Active Secondaries: Readable Secondary Replicas (AlwaysOn Availability Groups)](http://msdn.microsoft.com/en-us/library/ff878253).
  + Performing backup operations on secondary replicas. For more information, see [Active Secondaries: Backup on Secondary Replicas (AlwaysOn Availability Groups)](http://msdn.microsoft.com/en-us/library/hh245119).

Active secondary capabilities improve IT efficiency and reduce cost through better resource utilization of secondary hardware. In addition, offloading read-intent applications and backup jobs to secondary replicas helps to improve performance on the primary replica.

* Availability group listeners that provide fast application failover after an availability group fails over. For more information, see [Availability Group Listeners, Client Connectivity, and Application Failover (SQL Server)](http://msdn.microsoft.com/en-us/library/hh213417).
* A flexible failover policy for each availability group to provide some control over the automatic failover process. For more information, see [Failover and Failover Modes (AlwaysOn Availability Groups)](http://msdn.microsoft.com/en-us/library/hh213151).
* Automatic page repair for protection against page corruption. For more information, see [Automatic Page Repair (Availability Groups/Database Mirroring)](http://msdn.microsoft.com/en-us/library/bb677167).
* Forcing WSFC quorum (forced quorum). For more information, see [Windows Server Failover Clustering (WSFC) with SQL Server](http://msdn.microsoft.com/en-us/library/hh270278).
* Encryption and compression, which provide a secure, high performing transport.
* Interoperation with the following SQL Server features:
  + Change data capture
  + Change tracking
  + Contained databases
  + Database encryption
  + Database snapshots
  + FILESTREAM
  + FileTable
  + Full-text search—Full-Text indexes are synchronized with AlwaysOn secondary databases.
  + Log shipping
  + Remote Blob Store (RBS)
  + Replication
  + Service Broker
  + SQL Server Agent

For more information, [AlwaysOn Availability Groups: Interoperability (SQL Server)](http://msdn.microsoft.com/en-us/library/hh710077).

### AlwaysOn Availability Groups Tools

AlwaysOn Availability Groups provides an integrated set of tools to simplify deployment and management of availability groups, including:

* Transact-SQL DDL statements for creating and managing availability groups. For more information, see [Overview of Transact-SQL Statements for AlwaysOn Availability Groups (SQL Server)](http://msdn.microsoft.com/en-us/library/ff877941).
* Several SQL Server Management Studio wizards:
  + The New Availability Group Wizard creates and configures an availability group. In some environments, this wizard can also automatically prepare the secondary databases and start data synchronization for each of them. For more information, see [Use the New Availability Group Dialog Box (SQL Server Management Studio)](http://msdn.microsoft.com/en-us/library/gg509103).
  + The Add Database to Availability Group Wizard adds one or more primary databases to an existing availability group. In some environments, this wizard can also automatically prepare the secondary databases and start data synchronization for each of them. For more information, see [Use the Add Database to Availability Group Wizard (SQL Server)](http://msdn.microsoft.com/en-us/library/hh245118).
  + The Add Replica to Availability Group Wizard adds one or more secondary replicas to an existing availability group. In some environments, this wizard can also automatically prepare the secondary databases and start data synchronization for each of them. For more information, see [Use the Add Replica to Availability Group Wizard (SQL Server Management Studio)](http://msdn.microsoft.com/en-us/library/hh213239).
  + The Fail Over Availability Group Wizard initiates a manual failover on an availability group. Depending on the configuration and state of the secondary replica that you specify as the failover target, the wizard can perform either a planned or forced manual failover. For more information, see [Use the Fail Over Availability Group Wizard (SQL Server Management Studio)](http://msdn.microsoft.com/en-us/library/hh270276).
* The AlwaysOn Dashboard provides an at-a-glance view of the health of an availability group. To help database administrators make quick operational decisions, the dashboard provides visual indicators of the key states of availability groups and their component availability replicas and databases. The dashboard also provides launch points for various troubleshooting scenarios. For more information, see [Use the AlwaysOn Dashboard (SQL Server Management Studio)](http://msdn.microsoft.com/en-us/library/hh213474).
* The Object Explorer Details pane displays basic information about existing availability groups. For more information, see [Use the Object Explorer Details to Monitor Availability Groups (SQL Server Management Studio)](http://msdn.microsoft.com/en-us/library/ff878267).
* PowerShell cmdlets. For more information, see [Overview of PowerShell Cmdlets for AlwaysOn Availability Groups (SQL Server)](http://msdn.microsoft.com/en-us/library/ff878391).

[Online Operations](javascript:void(0))

* Extended support for online index builds

Indexes that include columns of large object (LOB) data—varchar(max), nvarchar(max), varbinary(max), or XML data—can now be built, rebuilt, or dropped online. For information about online index operations, see [Guidelines for Online Index Operations](http://msdn.microsoft.com/en-us/library/ms190981) and [Perform Index Operations Online](http://msdn.microsoft.com/en-us/library/ms177442).

* Reduced downtime for application upgrade

Adding table columns that contain default values is now a metadata-only operation, so only a brief exclusive lock needs to be taken on the object, typically lasting for less than a second. Long-term exclusive table locks are no longer needed. For information about adding table columns, see [Add Columns to a Table (Database Engine)](http://msdn.microsoft.com/en-us/library/ms190238).

**Manageability Enhancements (Database Engine)**

**SQL Server 2012**

[Other Versions](javascript:;)

Description: http://i.msdn.microsoft.com/Areas/Epx/Content/Images/ImageSprite.png

* [SQL Server 2008 R2](http://msdn.microsoft.com/en-us/library/704f6493-fa87-4c3a-9aec-bb15bfe1a86b(d=printer,v=sql.105))
* [SQL Server 2008](http://msdn.microsoft.com/en-us/library/704f6493-fa87-4c3a-9aec-bb15bfe1a86b(d=printer,v=sql.100))

Manageability of the SQL Server 2012 Database Engine is improved by enhancements to tools and monitoring features.

[SQL Server Management Studio](javascript:void(0))

SQL Server Management Studio introduces the following features in SQL Server 2012:

* Alternative keyboard shortcut schemes

SQL Server Management Studio supports two keyboard shortcut schemes. The new default keyboard shortcuts are based on the Microsoft Visual Studio 2010 keyboard shortcuts. You can configure Management Studio to use the keyboard shortcuts from SQL Server 2008 R2.

|  |
| --- |
| **Note** |
| SQL Server 2012 does not include the SQL Server 2000 Enterprise Manager shortcuts. |

For more information, see [SQL Server Management Studio Keyboard Shortcuts](http://msdn.microsoft.com/en-us/library/ms174205).

* Query Editor

The Database Engine Query Editor introduces enhanced functionality for Transact-SQL debugging and IntelliSense.

* + The Transact-SQL debugger introduces the following new features:
    - You can now debug Transact-SQL scripts running on instances of SQL Server 2005 Service Pack 2 (SP2) or later.
    - Transact-SQL breakpoints now support the following functionality:
      * A breakpoint condition is a Transact-SQL expression whose evaluation determines whether the breakpoint is invoked. For more information, see [Specify a Breakpoint Condition](http://msdn.microsoft.com/en-us/library/ff878392).
      * A breakpoint hit count specifies the number of times a breakpoint is encountered before it is invoked. For more information, see [Specify a Hit Count](http://msdn.microsoft.com/en-us/library/ff877958).
      * A breakpoint filter limits the breakpoint to operating only on specified computers, processes, or threads. For more information, see [Specify a Breakpoint Filter](http://msdn.microsoft.com/en-us/library/ff878257).
      * A breakpoint action specifies a custom task that is performed when the breakpoint is invoked. For more information, see [Specify a Breakpoint Action](http://msdn.microsoft.com/en-us/library/ff878534).
      * You can edit a breakpoint location to move a breakpoint from one Transact-SQL statement to another. For more information, See [Edit a Breakpoint Location](http://msdn.microsoft.com/en-us/library/ff878187).
    - The Watch window and Quick Watch now support watching Transact-SQL expressions. For more information, see [Transact-SQL Debugger Information](http://msdn.microsoft.com/en-us/library/cc646012).
    - When you move the cursor over a Transact-SQL identifier, a Quick Info pop up displays the name of the expression and its current value. For more information, see [Transact-SQL Debugger Information](http://msdn.microsoft.com/en-us/library/cc646012).
  + Transact-SQL IntelliSense

Transact-SQL IntelliSense introduces the following new features:

* + - Breakpoint validation prevents setting a breakpoint in an invalid location.
    - Transact-SQL code snippets are templates you can use as starting points when building Transact-SQL statements in batches and scripts. For more information, see [Insert Transact-SQL Snippets](http://msdn.microsoft.com/en-us/library/ff878440).
    - Transact-SQL Surround with snippets are templates you can use as staring points when enclosing sets of Transact-SQL statements in a BEGIN, IF, or WHILE block. For more information, see [Insert Surround-with Transact-SQL snippets](http://msdn.microsoft.com/en-us/library/ff878185).
  + Database Recovery Advisor

To recover a database from a failure, a database administrator has to restore a set of backups in a logically correct and meaningful restore sequence. The Database Recovery Advisor facilitates constructing restore plans that implement optimal correct restore sequences. Many known database restore issues and enhancements requested by customers have been addressed. Major enhancements introduced by the Database Recovery Advisor include the following:

* + - Restore-plan algorithm:  The algorithm used to construct restore plans has improved significantly, particularly for complex restore scenarios. Many edge cases, including forking scenarios in point-in-time restores, are handled more efficiently than in previous versions of SQL Server.
    - Point-in-time restores:  The Database Recovery Advisor greatly simplifies restoring a database to a given point in time. A visual backup timeline significantly enhances support for point-in-time restores. This visual timeline allows you to identify a feasible point in time as the target recovery point for restoring a database. The timeline facilitates traversing a forked recovery path (a path that spans recovery forks). A given point-in-time restore plan automatically includes the backups that are relevant to the restoring to your target point in time (date and time). For more information, see [Restore a SQL Server Database to a Point in Time (Full Recovery Model)](http://msdn.microsoft.com/en-us/library/ms179451).

For more information, see about the Database Recovery Advisor, see the following SQL Server Manageability blogs:

* + - [Recovery Advisor: An Introduction](http://blogs.msdn.com/b/managingsql/archive/2011/07/13/recovery-advisor-an-introduction.aspx)
    - [Recovery Advisor: Using SSMS to create/restore split backups](http://blogs.msdn.com/b/managingsql/archive/2011/07/13/recovery-advisor-using-ssms-to-create-restore-split-backups.aspx)
  + Page Restore dialog

Starting in SQL Server 2012, SQL Server Management Studio supports page restores. A new Page Restore dialog enables you to check database pages for corruption and to restore selected corrupt pages from a database backup and subsequent log backups. For information about page restore, see [Restore Pages (SQL Server)](http://msdn.microsoft.com/en-us/library/ms175168).

[Startup Options](javascript:void(0))

Database Engine startup options are now configured by using a new Startup Parameters tab of SQL Server Configuration Manager. For more information, see [Configure Server Startup Options (SQL Server Configuration Manager)](http://msdn.microsoft.com/en-us/library/ms345416).

[Contained Databases](javascript:void(0))

Moving databases from one instance of the SQL Server Database Engine to another instance of the Database Engine is simplified by introducing contained databases. Users in a contained database are no longer associated with logins on the instance of SQL Server. Many other dependencies on the instance are also removed. For more information, see [Contained Databases](http://msdn.microsoft.com/en-us/library/ff929071).

[Data-tier Applications](javascript:void(0))

The following changes have been made for data-tier applications (DACs) in CTP3:

* The data-tier application (DAC) upgrade has been changed to an in-place process that alters the existing database to match the schema defined in the new version of the DAC. This replaces the side-by-side upgrade process, which created a new database with the new schema definitions. The Upgrade a Data-Tier Application wizard has been updated to perform an in-place upgrade. The Upgrade method of the DacStore type is now deprecated, and replaced with a new IncrementalUpgrade method. Upgrades are also supported for DACs deployed to SQL Database. For more information, see [Upgrade a Data-tier Application](http://msdn.microsoft.com/en-us/library/ee634742).
* In addition to just extracting a schema definition as a new DAC package file, you can now export both the schema definition and data from a database as a DAC export file. You can then import the file to create a new database with the same schema and data. For more information, see [Export a Data-tier Application](http://msdn.microsoft.com/en-us/library/hh213241) and [Import a BACPAC File to Create a New User Database](http://msdn.microsoft.com/en-us/library/hh710052).
* Data-tier applications now support many more objects than in SQL Server 2008 R2. For more information, see [DAC Support For SQL Server Objects and Versions](http://msdn.microsoft.com/en-us/library/ee210549).

[Windows PowerShell](javascript:void(0))

Starting with SQL Server 2012, Windows PowerShell is no longer installed by SQL Server Setup. Windows PowerShell 2.0 is a pre-requisite for installing SQL Server 2012. If PowerShell 2.0 is not installed or enabled on your computer, you can enable it by following the instructions on the [Windows Management Framework](http://go.microsoft.com/fwlink/?LinkId=186214) page. For more information about SQL Server PowerShell, see [SQL Server PowerShell](http://msdn.microsoft.com/en-us/library/hh245198).

SQL Server 2012 now uses the new Windows PowerShell 2.0 feature called modules for loading the SQL Server components into a PowerShell environment. Users import the **sqlps** module into PowerShell, and the module then loads the SQL Server snap-ins. For more information, see [Run Windows PowerShell from SQL Server Management Studio](http://msdn.microsoft.com/en-us/library/hh212996).

The **sqlps** Utility is no longer a PowerShell 1.0 mini-shell; it now starts PowerShell 2.0 and imports the **sqlps** module. This improves SQL Server interoperability by making it easier for PowerShell scripts to also load the snap-ins for other products. The **sqlps** utility is also added to the list of deprecated features starting in SQL Server 2012.

The SQL Server PowerShell provider includes two new cmdlets: backup-sqldatabase and restore-sqldatabase. For more information, use the get-help cmdlet after loading in the sqlps module.

[-K Option Added to bcp.exe and sqlcmd.exe](javascript:void(0))

The [bcp Utility](http://msdn.microsoft.com/en-us/library/ms162802) and [sqlcmd Utility](http://msdn.microsoft.com/en-us/library/ms162773) utilities now have -K, which allows you to specify read-only access to a secondary replica in an AlwaysOn availability group. sqlcmd also has –M, to support AlwaysOn Availability Groups.

[sqlcmd.exe Behavior Change](javascript:void(0))

If you use sqlcmd.exe with XML mode (:XML ON command) when executing a SELECT \* from T FOR XML …, expect behavior changes.

Behavior prior to SQL Server 2012:

* Text data containing a single quote was replaced with the &apos; escape sequence.
* Legacy datetime data values with 0 fractional seconds were shown with three decimal digits.
* money data values with no decimal value were represented as whole numbers.

Beginning in SQL Server 2012:

* Text data containing a single quote is no longer replaced with &apos;. It is still valid XML and an XML parser will give the same result.
* Legacy datetime data values with no fractional seconds will not return 3 decimal digits. Other date/time types are not affected.
* money data values with no decimal value shows the 4 decimal digits.

[Database Engine Tuning Advisor Enhancements](javascript:void(0))

In SQL Server 2012, you can use the query plan cache as a Database Engine Tuning Advisor (DTA) workload. By doing this, you can avoid having to manually create a workload from a script or trace file. When you specify the plan cache as the DTA workload, the Database Engine Tuning Advisor selects the top 1,000 events to use for analysis. The number of events can be changed using the **–n** option of the DTA utility. For more information, see [Start and Use the Database Engine Tuning Advisor](http://msdn.microsoft.com/en-us/library/ms174202).

[New and Modified Dynamic Management Views and Functions](javascript:void(0))

The following system views have been added or modified.

|  |  |
| --- | --- |
| **System View** | **Description** |
| [sys.dm\_exec\_query\_stats (Transact-SQL)](http://msdn.microsoft.com/en-us/library/ms189741) | Added four columns to help troubleshoot long running queries. You can use the total\_rows, min\_rows, max\_rows and last\_rows aggregate row count columns to separate queries that are returning a large number of rows from problematic queries that may be missing an index or have a bad query plan. |
| [sys.dm\_os\_volume\_stats (Transact-SQL)](http://msdn.microsoft.com/en-us/library/hh223223) | This dynamic management functions returns information about the operating system volume (directory) on which the specified databases and files are stored. Use this dynamic management function to check the attributes of the physical disk drive or return available free space information about the directory. |
| [sys.dm\_os\_windows\_info (Transact-SQL)](http://msdn.microsoft.com/en-us/library/hh204565) | This dynamic management view returns one row that displays Windows operating system version information such as the OS version or language ID. |
| [sys.dm\_server\_memory\_dumps (Transact-SQL)](http://msdn.microsoft.com/en-us/library/hh204543)  [sys.dm\_server\_services (Transact-SQL)](http://msdn.microsoft.com/en-us/library/hh204542)  [sys.dm\_server\_registry (Transact-SQL)](http://msdn.microsoft.com/en-us/library/hh204561) | These dynamic management views return property information associated with the SQL Server, Full-text, and SQL Server Agent services that are installed on the host server. These views also contain configuration, installation, and memory dump file information. |

# Programmability Enhancements (Database Engine)

**SQL Server 2012**

[Other Versions](javascript:;)

Description: http://i.msdn.microsoft.com/Areas/Epx/Content/Images/ImageSprite.png

* [SQL Server 2008 R2](http://msdn.microsoft.com/en-us/library/09f0096e-ab95-4be0-8c01-f98753255747(d=printer,v=sql.105))
* [SQL Server 2008](http://msdn.microsoft.com/en-us/library/09f0096e-ab95-4be0-8c01-f98753255747(d=printer,v=sql.100))

Programmability enhancements in the Database Engine include FileTables, statistical semantic search, property-scoped full-text search and customizable proximity search, ad-hoc query paging, circular arc segment support for spatial types, support for sequence objects, default support for 15,000 partitions, and numerous improvements and additions to Transact-SQL.

[Native XML Web Services (SOAP/HTTP endpoints) is Removed](javascript:void(0))

Beginning in SQL Server 2012, you can no longer use CREATE ENDPOINT or ALTER ENDPOINT to add or modify SOAP/HTTP endpoints.

[FileTables](javascript:void(0))

The FileTable feature builds on top of the SQL Server FILESTREAM technology to bring support for the Windows file namespace and compatibility with Windows applications to the file data stored in SQL Server. This lets an application integrate its storage and data management components, and provides integrated SQL Server services (including full-text search and semantic search) over unstructured data and metadata, along with easy policy management and administration.

In summary, you can now store files and documents in special tables in SQL Server, but access them from Windows applications as if they were stored in the file system, without making any changes to the Windows applications.

For more information about the FileTable features, see [FileTables (SQL Server)](http://msdn.microsoft.com/en-us/library/ff929144).

[Statistical Semantic Search](javascript:void(0))

Statistical Semantic Search provides deep insight into unstructured documents stored in SQL Server databases by extracting statistically relevant key phrases, and then - based on these phrases – identifying similar documents. These results are made available as structured data through three Transact-SQL rowset functions.

Semantic search builds upon the existing full-text search feature in SQL Server but enables new scenarios that extend beyond syntactical keyword searches. While full-text search lets you query the words in a document, semantic search lets you query the meaning of the document. New scenarios include automatic tag extraction, related content discovery, and hierarchical navigation across similar content. For example, you can query the document similarity index to identify resumes that match a job description. Or, you can query the index of key phrases to build the taxonomy for an organization, or for a corpus of documents.

For more information, see [Semantic Search (SQL Server)](http://msdn.microsoft.com/en-us/library/gg492075).

Before you can use Semantic Search, you have to install, attach, and register an additional database. For more information, see [Install and Configure Semantic Search](http://msdn.microsoft.com/en-us/library/gg509085).

[Full-Text Search](javascript:void(0))

### Property Search

Beginning in SQL Server 2012, you can configure a full-text index to support property-scoped searching on properties, such as Author and Title, which are emitted by IFilters. This form of searching is known as property searching. Whether property searching is possible on a given type of document depends on whether the corresponding [filter](http://msdn.microsoft.com/en-us/library/ms142499) (IFilter) extracts search properties during full-text indexing. Among IFilters that extract a number of document properties are the IFilters for Microsoft Office 2007 document file types, such as .docx, .xlsx, and .pptx. For more information, see [Search Document Properties with Search Property Lists](http://msdn.microsoft.com/en-us/library/ee677637).

### Customizable NEAR

Beginning in SQL Server 2012, you can customize a proximity search by using the new custom NEAR option of the [CONTAINS](http://msdn.microsoft.com/en-us/library/ms187787) predicate or [CONTAINSTABLE](http://msdn.microsoft.com/en-us/library/ms189760) function. Custom NEAR enables you to optionally specify the maximum number of non-search terms that separate the first and last search terms in a match. Custom NEAR also enables you to optionally specify that words and phrases are matched only if they occur in the order in which you specify them. For more information, see [Search for Words Close to Another Word with NEAR](http://msdn.microsoft.com/en-us/library/ms142568).

### New Word Breakers and Stemmers

All the word breakers and stemmers used by Full-Text Search and Semantic Search, with the exception of the Korean language, are updated in this release. For consistency between the contents of indexes and the results of queries, we recommend that you repopulate existing full-text indexes after upgrading.

1. The third-party word breakers for English that were included with previous releases of SQL Server have been replaced with Microsoft components. If you have to retain the previous behavior, see [Change the Word Breaker Used for US English and UK English](http://msdn.microsoft.com/en-us/library/gg509108).
2. The third-party word breakers for Danish, Polish, and Turkish that were included with previous releases of SQL Server have been replaced with Microsoft components. The new components are enabled by default.
3. There are new word breakers for Czech and Greek. Previous releases of SQL Server Full-Text Search did not include support for these two languages.
4. The behavior of the new word breakers has changed. For more information, see [Behavior Changes to Full-Text Search](http://msdn.microsoft.com/en-us/library/ms143272). If you have to retain the previous behavior, see [Revert the Word Breakers Used by Search to the Previous Version](http://msdn.microsoft.com/en-us/library/hh510185).
5. This release installs the latest Microsoft word breakers and stemmers, but does not install the latest Microsoft filters. To download the latest filters, see [Microsoft Office 2010 Filter Packs](http://go.microsoft.com/fwlink/?LinkId=218293).

[New and Enhanced Spatial Features](javascript:void(0))

The new spatial features in SQL Server 2012 represent a significant milestone in the evolution of spatial data support in SQL Server. The support for full globe spatial objects and for circular arcs on the ellipsoid are industry firsts for relational database systems. The **geography** data type has achieved parity with the **geometry** data type in the functionality and the variety of methods that it supports. Overall performance, from spatial indexes to methods, has significantly improved. These and other improvements to spatial data support represent a significant step forward in the spatial capabilities of SQL Server.

For a detailed description and examples of these new spatial features, download the white paper, [New Spatial Features in SQL 2012](http://go.microsoft.com/fwlink/?LinkId=226407).

### Enhancements to spatial data types

New circular arcs and related methods

* New subtypes. There are 3 new subtypes of circular arcs:
  + **CircularString**
  + **CompoundCurve**
  + **CurvePolygon**
* New methods. All existing methods work on these circular objects. The following new methods are also introduced:
  + **BufferWithCurves()** uses circular arcs to construct a buffered object with a greatly reduced number of points compared to **STBuffer()**.
  + **STNumCurves()** and **STCurveN()** are used for iteration through the list of the circular arc edges.
  + **STCurveToLine()** and **CurveToLineWithTolerance()** are used for approximating circular arcs with line segments within default and user-specified tolerance.

New and updated methods and aggregates for geometry and geography

* New methods.
  + **IsValidDetailed()** returns a message that can help to identify why a spatial object is not valid.
  + **HasZ** returns 1 (true) if a spatial object contains at least one Z value.
  + **HasM** returns 1 (true) if a spatial object contains at least one M value.
  + **AsBinaryZM()** adds support for Z and M values to the OGC WKB format.
  + **ShortestLineTo()** returns a LineString that represents the shortest distance between two objects.
  + **STLength()** has been updated and now works on both valid and invalid LineStrings.
  + **MinDbCompatibilityLevel()** is a new method used for backward compatibility. It indicates whether spatial objects can be recognized by SQL Server 2008 and SQL Server 2008 R2.
* New aggregates. These aggregates are available only in Transact-SQL, and not in the client-side programming library.
  + **UnionAggregate**
  + **EnvelopeAggregate**
  + **CollectionAggregate**
  + **ConvexHullAggregate**

Improved precision.

All constructions and relations are now done with 48 bits of precision, compared to 27 bits used in SQL Server 2008 and SQL Server 2008 R2. This can reduce the errors caused by the rounding of floating-point coordinates.

Enhancements to the geography type

* Full globe. SQL Server now supports spatial objects that are larger than a logical hemisphere. Geography features were restricted to slightly less than a logical hemisphere in SQL Server 2008. In SQL Server 2012, they can now be as big as the entire globe. A new type of object, called FULLGLOBE, can be constructed or received as a result of an operation.
* New methods.
  + For invalid objects. The geography type now allows invalid objects to be inserted into a table. **STIsValid()** and **MakeValid()** allow invalid geography objects to be detected and corrected in a similar fashion to the geometry type.
  + For ring orientation. Geography polygons can now be accommodated without regard to ring orientation. This can lead to unintended behavior. **ReorientObject()** can be used to reorient polygon rings for cases where they are constructed with the wrong orientation.
  + geometry methods added for the geography type. **STWithin()**, **STContains()**, **STOverlaps()**, and **STConvexHull()** methods were previously available only for the geometry type, but have now been added for the geography type. With the exception of **STConvexHull()**, these new methods are supported by spatial indexes.
* New SRID. A new spatial reference id (SRID), 104001, has been added to the list of supported spatial reference systems. This new SRID is an Earth unit sphere (a sphere of radius 1) and can be used with the geography type to perform optimized numerical computations when more precise ellipsoidal mathematics are not required.

### Spatial performance improvements

Spatial index improvements

* New auto grid. A new auto grid spatial index is available for both spatial types (**geometry\_auto\_grid** and **geography\_auto\_grid**). The new auto grid uses a different strategy to pick the right tradeoff between performance and efficiency. For more information, see [CREATE SPATIAL INDEX (Transact-SQL)](http://msdn.microsoft.com/en-us/library/bb934196).
* New spatial index hint, **SPATIAL\_WINDOW\_MAX\_CELLS**. This new spatial hint is critical for fine-tuning query performance using a spatial index. Dense spatial data often requires a higher SPATIAL\_WINDOW\_MAX\_CELLS, whereas sparse spatial data often demands a lower SPATIAL\_WINDOW\_MAX\_CELLS for optimum performance. This hint does not guarantee that a spatial index will be used in the query plan. However, if it is used, this hint will override the default WINDOW\_MAX\_CELLS parameter.
* Compression for spatial indexes. For more information, see [CREATE SPATIAL INDEX (Transact-SQL)](http://msdn.microsoft.com/en-us/library/bb934196).

Additional performance improvements

* An optimized nearest neighbor query plan is available when certain syntax is used.
* Several other methods have been optimized for common scenarios.
* Spatial aggregates have better performance as a result of improvements that affect all CLR UDT aggregates.

### Other spatial improvements

Spatial helper stored procedures

Two new helper stored procedures are available. These procedures can be used to evaluate the distribution of spatial data within a table over a given spatial column.

* [sp\_help\_spatial\_geography\_histogram (Transact-SQL)](http://msdn.microsoft.com/en-us/library/gg492090)
* [sp\_help\_spatial\_geometry\_histogram (Transact-SQL)](http://msdn.microsoft.com/en-us/library/gg509094)

Support for persisted computed columns

UDTs and spatial types can now be persisted in computed columns.

### Changes in the client-side spatial programming library

* New sink interfaces, [IGeometrySink110](http://msdn.microsoft.com/en-us/library/microsoft.sqlserver.types.igeometrysink110) and [IGeographySink110](http://msdn.microsoft.com/en-us/library/microsoft.sqlserver.types.igeographysink110), are available.
* Geometry and geography builders ([SqlGeometryBuilder](http://msdn.microsoft.com/en-us/library/microsoft.sqlserver.types.sqlgeometrybuilder) and [SqlGeographyBuilder](http://msdn.microsoft.com/en-us/library/microsoft.sqlserver.types.sqlgeographybuilder)) now support circular arc constructions.
* A new method, **Deserialize**, has been added to both types ([Deserialize(SqlBytes)](http://msdn.microsoft.com/en-us/library/microsoft.sqlserver.types.sqlgeometry.deserialize) and [Deserialize(SqlBytes)](http://msdn.microsoft.com/en-us/library/microsoft.sqlserver.types.sqlgeography.deserialize)). This method simplifies deserialization.

[Metadata Discovery](javascript:void(0))

The [SET FMTONLY](http://msdn.microsoft.com/en-us/library/ms173839) option for determining the format of a response without actually running the query is replaced with [sp\_describe\_first\_result\_set](http://msdn.microsoft.com/en-us/library/ff878602), [sp\_describe\_undeclared\_parameters](http://msdn.microsoft.com/en-us/library/ff878260), [sys.dm\_exec\_describe\_first\_result\_set](http://msdn.microsoft.com/en-us/library/ff878258), and [sys.dm\_exec\_describe\_first\_result\_set\_for\_object](http://msdn.microsoft.com/en-us/library/ff878236).

[EXECUTE Statement](javascript:void(0))

The EXECUTE statement can now specify the metadata returned from the statement by using the WITH RESULT SETS argument. For more information, see [EXECUTE (Transact-SQL)](http://msdn.microsoft.com/en-us/library/ms188332).

[UTF-16 Support](javascript:void(0))

### UTF-16 Supplementary Characters (SC) Collations

A new family of supplementary characters (SC) collations can be used with the data types nchar, nvarchar and sql\_variant. For example: Latin1\_General\_100\_CI\_AS\_SC or, if using a Japanese collation, Japanese\_Bushu\_Kakusu\_100\_CI\_AS\_SC. These collations encode Unicode characters in the UTF-16 format. Characters with codepoint values larger than 0xFFFF require two consecutive 16-bit words. These characters are called supplementary characters, and the two consecutive 16-bit words are called surrogate pairs. SC collations can improve searching and sorting by functions that use the Unicode types nchar and nvarchar. For more information, see [Collation and Unicode Support](http://msdn.microsoft.com/en-us/library/ms143726).

### UTF-16 Supplementary Characters (SC) Collation Option for XML

SQL Server 2012 adds a new collation option – "SC" or "supplementary characters" – that identifies whether a collation is UTF-16-aware. For more information, see [Collation and Unicode Support](http://msdn.microsoft.com/en-us/library/ms143726). SQL Server 2012 also adds support for this collation option in the SQL Types XML schema and in other locations where SQL Server exposes or consumes this information in an XML context. The affected locations are the following:

* SQL Types XML schema. The schema version is now 1.2. The schema is backward-compatible and the target namespace has not changed. The schema now exposes the supplementaryCharacters global attribute.
* XMLSCHEMA directive with FOR XML. The new global attribute is exposed in the inline schemas and instance annotations generated by the XMLSCHEMA directive, alongside similar attributes such as localeId and sqlCompareOptions. This directive is supported with FOR XML in RAW and AUTO modes, but not in EXPLICIT or PATH modes.
* sys.sys XML schema collection. The new global attribute is prepopulated in the built-in sys.sys XML schema collection and is made available implicitly in all other XML schema collections that import the SQL Types XML schema.
* Catalog views. The new global attribute is now listed in the following catalog views:
  + sys.xml\_schema\_components
  + sys.xml\_schema\_attributes
  + sys.xml\_schema\_component\_placements
* Upgraded XML schema collections. After upgrade from a previous version of SQL Server, the new global attribute is exposed in all XML schema collections that import the SQL Types XML schema.
* XML column sets. The new global attribute is added to XML column set values that represent sql\_variant strings that use the new UTF-16 collations. It can also be applied during inserts and updates to set string values of type sql\_variant in sparse columns to use the UTF-16 aware collation.

[Ad-hoc Query Paging Implementation](javascript:void(0))

You can specify a range of rows returned by a SELECT statement based on row offset and row count values that you provide. This is useful when you want to control the number of rows sent to a client application for a given query. For more information, see [ORDER BY Clause (Transact-SQL)](http://msdn.microsoft.com/en-us/library/ms188385).

[Sequence Objects](javascript:void(0))

A sequence object is a user-defined schema-bound object that generates a sequence of numeric values according to the specification with which the sequence was created. It operates similar to an identity column, but sequence numbers are not restricted to use in a single table. For more information, see [Sequence Numbers](http://msdn.microsoft.com/en-us/library/ff878058).

[THROW statement](javascript:void(0))

The THROW statement can be used to raise an exception and transfer execution to a CATCH block of a TRY…CATCH construct. For more information, see [THROW (Transact-SQL)](http://msdn.microsoft.com/en-us/library/ee677615).

[14 New Functions and 1 Changed Function](javascript:void(0))

SQL Server 2012 introduces 14 new built-in functions. These functions ease the path of migration for information workers by emulating functionality that is found in the expression languages of many desktop applications. However these functions will also be useful to experienced users of SQL Server.

The new functions are:

Conversion functions

* [PARSE (Transact-SQL)](http://msdn.microsoft.com/en-us/library/hh213316)
* [TRY\_CONVERT (Transact-SQL)](http://msdn.microsoft.com/en-us/library/hh230993)
* [TRY\_PARSE (Transact-SQL)](http://msdn.microsoft.com/en-us/library/hh213126)

Date and time functions

* [DATEFROMPARTS (Transact-SQL)](http://msdn.microsoft.com/en-us/library/hh213228)
* [DATETIME2FROMPARTS (Transact-SQL)](http://msdn.microsoft.com/en-us/library/hh213312)
* [DATETIMEFROMPARTS (Transact-SQL)](http://msdn.microsoft.com/en-us/library/hh213233)
* [DATETIMEOFFSETFROMPARTS (Transact-SQL)](http://msdn.microsoft.com/en-us/library/hh231077)
* [EOMONTH (Transact-SQL)](http://msdn.microsoft.com/en-us/library/hh213020)
* [SMALLDATETIMEFROMPARTS (Transact-SQL)](http://msdn.microsoft.com/en-us/library/hh213396)
* [TIMEFROMPARTS (Transact-SQL)](http://msdn.microsoft.com/en-us/library/hh213398)

Logical functions

* [CHOOSE (Transact-SQL)](http://msdn.microsoft.com/en-us/library/hh213019)
* [IIF (Transact-SQL)](http://msdn.microsoft.com/en-us/library/hh213574)

String functions

* [CONCAT (Transact-SQL)](http://msdn.microsoft.com/en-us/library/hh231515)
* [FORMAT (Transact-SQL)](http://msdn.microsoft.com/en-us/library/hh213505)

In addition to the 14 new functions, one existing function has been changed. The existing [LOG (Transact-SQL)](http://msdn.microsoft.com/en-us/library/ms190319) function now has an optional second base parameter.

[SQL Server Express LocalDB](javascript:void(0))

SQL Server Express **LocalDB** is a new lightweight edition of Express that has all its programmability features, yet runs in user mode and has a fast, zero-configuration installation and short list of pre-requisites. The **LocalDB** edition of SQL Server is targeted to program developers. **LocalDB** installation copies a minimal set of files necessary to start the SQL Server Database Engine. Once **LocalDB** is installed, developers initiate a connection by using a special connection string. When connecting, the necessary infrastructure is automatically created and started, enabling the application to use the database without complex or time consuming configuration tasks. Developer Tools can provide developers with a SQL Server Database Engine that lets them write and test Transact-SQL code without having to manage a full server instance of SQL Server. An instance of SQL Server Express **LocalDB** can be managed by using the SqlLocalDB.exe utility. SQL Server Express **LocalDB** should be used in place of the SQL Server Express user instance feature which is deprecated. For more information, see [SQL Server 2012 Express LocalDB](http://msdn.microsoft.com/en-us/library/hh510202).

[New and Enhanced Query Optimizer Hints](javascript:void(0))

The syntax for the FORCESEEK table hint has been modified. You can now specify an index and index columns to further control the access method on the index. The existing FORCESEEK syntax remains unmodified and works as before. No changes to applications are necessary if you do not plan to use the new functionality.

The FORCESCAN table hint has been added. It complements the FORCESEEK hint by specifying that the query optimizer use only an index scan operation as the access path to the table or view referenced in the query. The FORCESCAN hint can be useful for queries in which the optimizer underestimates the number of affected rows and chooses a seek operation rather than a scan operation. FORCESCAN can be specified with or without an INDEX hint. For more information, see [Table Hints (Transact-SQL)](http://msdn.microsoft.com/en-us/library/ms187373).

[Extended Event Enhancements](javascript:void(0))

The following new Extended Events are available.

**page\_allocated**:

* Fields: worker\_address, number\_pages, page\_size, page\_location, allocator\_type, page\_allocator\_type, pool\_id

**page\_freed**:

* Fields: worker\_address, number\_pages, page\_size, page\_location, allocator\_type, page\_allocator\_type, pool\_id

**allocation\_failure**:

* Fields: worker\_address, failure\_type, allocation\_failure\_type, resource\_size, pool\_id, factor

The following Extended Events have been modified.

**resource\_monitor\_ring\_buffer\_record**:

* Fields removed: single\_pages\_kb, multiple\_pages\_kb
* Fields added: target\_kb, pages\_kb

**memory\_node\_oom\_ring\_buffer\_recorded**:

* Fields removed: single\_pages\_kb, multiple\_pages\_kb
* Fields added: target\_kb, pages\_kb

[OVER Clause Support Enhanced](javascript:void(0))

The OVER clause has been extended to support window functions. Window functions perform a calculation across a set of rows that are in some relationship to the current row. For example, you can use the ROWS or RANGE clause over a set of rows to calculate a moving average or cumulative total. For more information, see [OVER Clause (Transact-SQL)](http://msdn.microsoft.com/en-us/library/ms189461).

In addition, ordering rows within a partition is now supported in the aggregate functions that allow the OVER clause to be specified.

[Analytic Functions](javascript:void(0))

The following analytic functions have been added.

|  |  |  |
| --- | --- | --- |
| [CUME\_DIST (Transact-SQL)](http://msdn.microsoft.com/en-us/library/hh231078) | [LAST\_VALUE (Transact-SQL)](http://msdn.microsoft.com/en-us/library/hh231517) | [PERCENTILE\_DISC (Transact-SQL)](http://msdn.microsoft.com/en-us/library/hh231327) |
| [FIRST\_VALUE (Transact-SQL)](http://msdn.microsoft.com/en-us/library/hh213018) | [LEAD (Transact-SQL)](http://msdn.microsoft.com/en-us/library/hh213125) | [PERCENT\_RANK (Transact-SQL)](http://msdn.microsoft.com/en-us/library/hh213573) |
| [LAG (Transact-SQL)](http://msdn.microsoft.com/en-us/library/hh231256) | [PERCENTILE\_CONT (Transact-SQL)](http://msdn.microsoft.com/en-us/library/hh231473) |  |

[XQuery Functions Are Surrogate-Aware](javascript:void(0))

The W3C recommendation for XQuery functions and operators requires them to count a surrogate pair that represents a high-range Unicode character as a single glyph in UTF-16 encoding. However, in versions of SQL Server prior to SQL Server 2012, string functions did not recognize surrogate pairs as a single character. Some string operations – such as string length calculations and substring extractions – returned incorrect results. SQL Server 2012 now fully supports UTF-16 and the correct handling of surrogate pairs. For more information, see the section "XQuery Functions Are Surrogate-Aware" in the topic [Breaking Changes to Database Engine Features in SQL Server 2012](http://msdn.microsoft.com/en-us/library/ms143179).

**Scalability and Performance Enhancements (Database Engine)**

**SQL Server 2012**

[Other Versions](javascript:;)

Description: http://i.msdn.microsoft.com/Areas/Epx/Content/Images/ImageSprite.png

* [SQL Server 2008 R2](http://msdn.microsoft.com/en-us/library/d8b36fb6-4bd5-4b18-af0a-fc3542fce2f5(d=printer,v=sql.105))
* [SQL Server 2008](http://msdn.microsoft.com/en-us/library/d8b36fb6-4bd5-4b18-af0a-fc3542fce2f5(d=printer,v=sql.100))

Scalability and performance enhancements in the Database Engine include columnstore indexes.

[Columnstore Indexes](javascript:void(0))

The SQL Server 2012 introduces a new data warehouse query acceleration feature based on a new type of index called an xVelocity memory optimized columnstore. This new index, combined with enhanced query processing features, improves data warehouse query performance by hundreds to thousands of times in some cases, and can routinely give a tenfold speedup for a broad range of decision support queries. This can allow end users to get more business value from their data through fast, interactive exploration. IT workers can reduce development costs and ETL times because columnstore indexes limit or eliminate the need to rely on pre-built aggregates, including user-defined summary tables, and indexed (materialized) views. Furthermore, columnstore indexes can greatly improve ROLAP performance, making ROLAP more attractive. For more information, see [Columnstore Indexes](http://msdn.microsoft.com/en-us/library/gg492088).

[Online Index Create, Rebuild, and Drop](javascript:void(0))

Indexes containing XML, varchar(max), nvarchar(max), and varbinary(max) columns can now be created, rebuilt, and dropped as an online operation.

[Partition Support Increased](javascript:void(0))

SQL Server 2012 supports up to 15,000 partitions by default. In earlier versions, the number of partitions was limited to 1,000 by default.

[FILESTREAM Filegroups Can Contain Multiple Files](javascript:void(0))

A FILESTREAM filegroup can contain more than one file. For a code example that demonstrates how to create a FILESTREAM filegroup that contains multiple files, see [CREATE DATABASE (Transact-SQL)](http://msdn.microsoft.com/en-us/library/ms176061).

You can improve I/O scalability for FILESTREAM data by placing different files within the same FILESTREAM filegroup on different volumes. This feature eliminates the need for complicated workarounds that use partitioning and multiple FILESTREAM filegroups.

**Security Enhancements (Database Engine)**

**SQL Server 2012**

[Other Versions](javascript:;)

Description: http://i.msdn.microsoft.com/Areas/Epx/Content/Images/ImageSprite.png

* [SQL Server 2008 R2](http://msdn.microsoft.com/en-us/library/42919d6c-1fd4-466e-ac75-d579248c07df(d=printer,v=sql.105))
* [SQL Server 2008](http://msdn.microsoft.com/en-us/library/42919d6c-1fd4-466e-ac75-d579248c07df(d=printer,v=sql.100))

Security enhancements in the SQL Server Database Engine include provisioning during setup, new SEARCH PROPERTY LIST permissions, new user-defined server roles, and new ways of managing server and database roles.

[Provisioning During Setup](javascript:void(0))

To enhance role separation, **BUILTIN\administrators** and **Local System** (**NT AUTHORITY\SYSTEM**) are not automatically provisioned in the sysadmin fixed server role. Local administrators can still access the Database Engine when in single user mode.

SQL Server now supports Managed Service Accounts and Virtual Accounts when installed on Windows 7 or Windows Server 2008 R2. For more information, see [Configure Windows Service Accounts and Permissions](http://msdn.microsoft.com/en-us/library/ms143504).

The protection of operating services under a per-service SID is now extended to all operating systems. For more information, see [Configure Windows Service Accounts and Permissions](http://msdn.microsoft.com/en-us/library/ms143504).

[New Permissions](javascript:void(0))

There are 19 new permissions available in the Database Engine. To see all permissions execute the following statement.

Transact-SQL

[Copy](javascript:if%20(window.epx.codeSnippet)window.epx.codeSnippet.copyCode('CodeSnippetContainerCode_8e8c8766-e350-49bc-ab3c-34a98ad22cba');" \o "Copy to clipboard.)

SELECT \* FROM sys.fn\_builtin\_permissions('');

The new permissions are as follows:

New GRANT, DENY, and REVOKE permissions to CONTROL/VIEW DEFINTION/TAKE OWNERSHIP/REFERENCES/ALTER on a search property list are available.

New GRANT, DENY, and REVOKE permissions to ALTER ANY SERVER ROLE, CREATE SERVER ROLE, and CONTROL/VIEW DEFINTION/TAKE OWNERSHIP/ALTER on a server role.

New GRANT, DENY, and REVOKE permissions to ALTER ANY AVAILABILITY GROUP, CREATE AVAILABILITY GROUP, and CONTROL/VIEW DEFINTION/TAKE OWNERSHIP/ALTER on an availability group.

New GRANT, DENY, and REVOKE permissions to the CREATE SEQUENCE permission.

New GRANT, DENY, and REVOKE permissions to the ALTER ANY EVENT SESSION permission.

[New Role Management](javascript:void(0))

User-defined server roles are now available. To manage user-defined server roles use [CREATE SERVER ROLE](http://msdn.microsoft.com/en-us/library/ee677610), [ALTER SERVER ROLE](http://msdn.microsoft.com/en-us/library/ee677634), and [DROP SERVER ROLE](http://msdn.microsoft.com/en-us/library/ee677643). To add and remove members from all server roles, use [ALTER SERVER ROLE](http://msdn.microsoft.com/en-us/library/ee677634) … WITH ADD MEMBER. [sp\_addsrvrolemember](http://msdn.microsoft.com/en-us/library/ms186320) and [sp\_dropsrvrolemember](http://msdn.microsoft.com/en-us/library/ms186270) are deprecated.

[ALTER ROLE](http://msdn.microsoft.com/en-us/library/ms189775) is modified to add or remove members from roles by using ADD MEMBER syntax. [sp\_addrolemember](http://msdn.microsoft.com/en-us/library/ms187750) and [sp\_droprolemember](http://msdn.microsoft.com/en-us/library/ms188369) are deprecated.

[IS\_ROLEMEMBER](http://msdn.microsoft.com/en-us/library/ee677633) is added to check the membership of database roles.

[Default Schema for Groups](javascript:void(0))

You can now define a default schema for a Windows group. When an object is created by a Windows user and when a default schema is not specified, SQL Server no longer automatically creates a schema. For more information about default schemas, see [CREATE USER](http://msdn.microsoft.com/en-us/library/ms173463).

[SQL Server Audit Enhancements](javascript:void(0))

Support for server level auditing is expanded to include all editions of SQL Server. Database level auditing is limited to Enterprise, Developer, and Evaluation editions.

SQL Server Audit is now more resilient to failures to write to the audit log. For example, if the target directory is on a remote share and the network goes down, SQL Server Audit will now be able to recover once the network connection is re-established. In addition, a new option has been introduced to fail an operation that would otherwise generate an audit event to be written to a failed audit target. For more information, see the **FAIL\_OPERATION** option for the **ON\_FAILURE** event in [CREATE SERVER AUDIT](http://msdn.microsoft.com/en-us/library/cc280448).

Previously, Audit logs could have an indeterminate number of log files or else be rolled-over after a predefined number. A new option has been introduced to cap the number of audit files without rolling over, in order to allow customers to control the amount of audit information collected without losing audit records. For more information, see the **MAX\_FILES** option in [CREATE SERVER AUDIT](http://msdn.microsoft.com/en-us/library/cc280448).

When possible, the audit log provides additional Transact-SQL stack frame information. In many cases, auditors can now determine whether a query was issued through a stored procedure or directly by an application.

SQL Server audit specifications now support a user-defined audit group. Audited events can be written to the audit log by using the new [sp\_audit\_write (Transact-SQL)](http://msdn.microsoft.com/en-us/library/gg471540) procedure. User-defined audit events allow applications to write custom information to the audit log, such as the name of the application user who has connected in cases where a common login is used to connect to SQL Server.

New columns are added to [sys.server\_file\_audits](http://msdn.microsoft.com/en-us/library/cc280544), [sys.server\_audits](http://msdn.microsoft.com/en-us/library/cc280727), and [sys.fn\_get\_audit\_file](http://msdn.microsoft.com/en-us/library/cc280765) to track user-defined audit events.

SQL Server Audit now supports the ability to filter audit events before they are written to the audit log. For more information, see the **WHERE** clause in [CREATE SERVER AUDIT](http://msdn.microsoft.com/en-us/library/cc280448) and [ALTER SERVER AUDIT](http://msdn.microsoft.com/en-us/library/cc280563).

New audit groups support the monitoring of contained database users.

The new audit options have been added to the audit dialog boxes in Management Studio.

[Database Engine Access is Allowed Through Contained Databases](javascript:void(0))

Access to contained databases is permitted through contained database users which do not require logins. SQL Server system administrators should understand how contained databases change the SQL Server security model. For more information, see [Security Best Practices with Contained Databases](http://msdn.microsoft.com/en-us/library/ff929055).

[Hashing Algorithms](javascript:void(0))

The HASHBYTES function now supports the **SHA2\_256**, and **SHA2\_512** algorithms.

[Further Deprecation of RC4](javascript:void(0))

The RC4 algorithm is only supported for backward compatibility. New material can only be encrypted using RC4 or RC4\_128 when the database is in compatibility level 90 or 100. (Not recommended.) Use a newer algorithm such as one of the AES algorithms instead. In SQL Server 2012 material encrypted using RC4 or RC4\_128 can be decrypted in any compatibility level.

[Certificate Key Length](javascript:void(0))

When creating certificates, the maximum length of private keys imported from an external source is expanded from 3,456 to 4,096 bits.

[Service Master Key and Database Master Key Encryption changes from 3DES to AES](javascript:void(0))

SQL Server 2012 uses the AES encryption algorithm to protect the service master key (SMK) and the database master key (DMK). AES is a newer encryption algorithm than 3DES used in earlier versions. After upgrading an instance of the Database Engine to SQL Server 2012 the SMK and DMK should be regenerated in order to upgrade the master keys to AES. For more information about regenerating the SMK, see [ALTER SERVICE MASTER KEY (Transact-SQL)](http://msdn.microsoft.com/en-us/library/ms187788) and [ALTER MASTER KEY (Transact-SQL)](http://msdn.microsoft.com/en-us/library/ms186937).

[Certificates Can be Created from Binary](javascript:void(0))

[CREATE CERTIFICATE (Transact-SQL)](http://msdn.microsoft.com/en-us/library/ms187798) has the FROM BINARY option to allow specifying the binary description of an ASN encoded certificate. New functions [CERTENCODED (Transact-SQL)](http://msdn.microsoft.com/en-us/library/hh510218) and [CERTPRIVATEKEY (Transact-SQL)](http://msdn.microsoft.com/en-us/library/hh510206) can be used to extract a binary description of an existing certificate.

**Resource Governor Enhancements (Database Engine)**

**SQL Server 2012**

The enhancements to the Resource Governor enable you to more effectively govern performance in multi-tenancy environments like private cloud. The enhancements include support for 64 resource pools, greater CPU usage control, and resource pool affinity for partitioning of physical resources and predictable resource allocation.

For a description of the new Resource Governor features in SQL Server 2012, see [Resource Governor in SQL Server 2012](http://msdn.microsoft.com/en-us/library/jj573256.aspx).

The following topics provide additional information.

* [ALTER RESOURCE POOL (Transact-SQL)](http://msdn.microsoft.com/en-us/library/bb934024)
* [CREATE RESOURCE POOL (Transact-SQL)](http://msdn.microsoft.com/en-us/library/bb895329)
* [sys.dm\_exec\_query\_resource\_semaphores (Transact-SQL)](http://msdn.microsoft.com/en-us/library/ms366321)
* [sys.dm\_resource\_governor\_resource\_pool\_affinity (Transact-SQL)](http://msdn.microsoft.com/en-us/library/hh710066)
* [sys.dm\_resource\_governor\_resource\_pools (Transact-SQL)](http://msdn.microsoft.com/en-us/library/bb934023)
* [sys.dm\_exec\_query\_memory\_grants (Transact-SQL)](http://msdn.microsoft.com/en-us/library/ms365393)
* [sys.resource\_governor\_workload\_groups (Transact-SQL)](http://msdn.microsoft.com/en-us/library/bb895237)

**ALTER RESOURCE POOL (Transact-SQL)**

**SQL Server 2012**

[Other Versions](javascript:;)

Description: http://i.msdn.microsoft.com/Areas/Epx/Content/Images/ImageSprite.png

* [SQL Server 2008 R2](http://msdn.microsoft.com/en-us/library/bb934024(d=printer,v=sql.105))
* [SQL Server 2008](http://msdn.microsoft.com/en-us/library/bb934024(d=printer,v=sql.100))

Changes an existing Resource Governor resource pool configuration.

[Transact-SQL Syntax Conventions](http://msdn.microsoft.com/en-us/library/ms177563).The introduction is required.

[Syntax](javascript:void(0))

ALTER RESOURCE POOL { pool\_name | "default" }

[WITH

     ( [ MIN\_CPU\_PERCENT = value ]

     [ [ , ] MAX\_CPU\_PERCENT = value ]

     [ [ , ] CAP\_CPU\_PERCENT = value ]

     [ [ , ] AFFINITY {SCHEDULER = AUTO | (Scheduler\_range\_spec) | NUMANODE = (NUMA\_node\_range\_spec)}]

     [ [ , ] MIN\_MEMORY\_PERCENT = value ]

     [ [ , ] MAX\_MEMORY\_PERCENT = value ] )

]

[;]

Scheduler\_range\_spec::=

{SCHED\_ID | SCHED\_ID TO SCHED\_ID}[,…n]

NUMA\_node\_range\_spec::=

{NUMA\_node\_ID | NUMA\_node\_ID TO NUMA\_node\_ID}[,…n]

[Arguments](javascript:void(0))

{ pool\_name | "default" }

Is the name of an existing user-defined resource pool or the default resource pool that is created when SQL Server 2012 is installed.

"default" must be enclosed by quotation marks ("") or brackets ([]) when used with ALTER RESOURCE POOL to avoid conflict with DEFAULT, which is a system reserved word. For more information, see [Database Identifiers](http://msdn.microsoft.com/en-us/library/ms175874).

|  |
| --- |
| **Note** |
| Predefined workload groups and resource pools all use lowercase names, such as "default". This should be taken into account for servers that use case-sensitive collation. Servers with case-insensitive collation, such as SQL\_Latin1\_General\_CP1\_CI\_AS, will treat "default" and "Default" as the same. |

MIN\_CPU\_PERCENT =value

Specifies the guaranteed average CPU bandwidth for all requests in the resource pool when there is CPU contention. value is an integer with a default setting of 0. The allowed range for value is from 0 through 100.

MAX\_CPU\_PERCENT =value

Specifies the maximum average CPU bandwidth that all requests in the resource pool will receive when there is CPU contention. value is an integer with a default setting of 100. The allowed range for value is from 1 through 100.

CAP\_CPU\_PERCENT =value

Specifies a hard cap on the CPU bandwidth that all requests in the resource pool will receive. Limits the maximum CPU bandwidth level to be the same as the specified value. value is an integer with a default setting of 100. The allowed range for value is from 1 through 100.

AFFINITY {SCHEDULER = AUTO | (Scheduler\_range\_spec) | NUMANODE = (NUMA\_node\_range\_spec)}

Attach the resource pool to specific schedulers. The default value is AUTO.

AFFINITY SCHEDULER = (Scheduler\_range\_spec) maps the resource pool to the SQL Server schedules identified by the given IDs. These IDs map to the values in the scheduler\_id column in [sys.dm\_os\_schedulers (Transact-SQL)](http://msdn.microsoft.com/en-us/library/ms177526).

When you use AFFINITY NAMANODE = (NUMA\_node\_range\_spec), the resource pool is affinitized to the SQL Server schedulers that map to the physical CPUs that correspond to the given NUMA node or range of nodes. You can use the following Transact-SQL query to discover the mapping between the physical NUMA configuration and the SQL Server scheduler IDs.

SELECT osn.memory\_node\_id AS [numa\_node\_id], sc.cpu\_id, sc.scheduler\_id

FROM sys.dm\_os\_nodes AS osn

INNER JOIN sys.dm\_os\_schedulers AS sc ON osn.node\_id = sc.parent\_node\_id AND sc.scheduler\_id < 1048576

MIN\_MEMORY\_PERCENT =value

Specifies the minimum amount of memory reserved for this resource pool that can not be shared with other resource pools. value is an integer with a default setting of 0. The allowed range for value is from 0 through 100.

MAX\_MEMORY\_PERCENT =value

Specifies the total server memory that can be used by requests in this resource pool. value is an integer with a default setting of 100. The allowed range for value is from 1 through 100.

[Remarks](javascript:void(0))

MAX\_CPU\_PERCENT and MAX\_MEMORY\_PERCENT must be greater than or equal to MIN\_CPU\_PERCENT and MIN\_MEMORY\_PERCENT, respectively.

CAP\_CPU\_PERCENT differs from MAX\_CPU\_PERCENT in that workloads associated with the pool can use CPU capacity above the value of MAX\_CPU\_PERCENT if it is available, but not above the value of CAP\_CPU\_PERCENT.

The total CPU percentage for each affinitized component (scheduler(s) or NUMA node(s)) should not exceed 100%.

When you are executing DDL statements, we recommend that you be familiar with Resource Governor states. For more information, see [Resource Governor](http://msdn.microsoft.com/en-us/library/bb933866).

[Permissions](javascript:void(0))

Requires CONTROL SERVER permission.

[Examples](javascript:void(0))

The following example keeps all the default resource pool settings on the default pool except for MAX\_CPU\_PERCENT, which is changed to 25.

ALTER RESOURCE POOL "default"

WITH

( MAX\_CPU\_PERCENT = 25)

GO

ALTER RESOURCE GOVERNOR RECONFIGURE

GO

In the following example, the CAP\_CPU\_PERCENT sets the hard cap to 80% and AFFINITY SCHEDULER is set to an individual value of 8 and a range of 12 to 16.

ALTER RESOURCE POOL Pool25

WITH(

MIN\_CPU\_PERCENT = 5,

MAX\_CPU\_PERCENT = 10,

CAP\_CPU\_PERCENT = 80,

AFFINITY SCHEDULER = (8, 12 TO 16),

MIN\_MEMORY\_PERCENT = 5,

MAX\_MEMORY\_PERCENT = 15

);

GO

ALTER RESOURCE GOVERNOR RECONFIGURE

GO

**CREATE RESOURCE POOL (Transact-SQL)**

**SQL Server 2012**

[Other Versions](javascript:;)

Description: http://i.msdn.microsoft.com/Areas/Epx/Content/Images/ImageSprite.png

* [SQL Server 2008 R2](http://msdn.microsoft.com/en-us/library/bb895329(d=printer,v=sql.105))
* [SQL Server 2008](http://msdn.microsoft.com/en-us/library/bb895329(d=printer,v=sql.100))

Creates a Resource Governor resource pool. Resource Governor is not available in every edition of Microsoft SQL Server. For a list of features that are supported by the editions of SQL Server, see [Features Supported by the Editions of SQL Server 2012](http://msdn.microsoft.com/en-us/library/cc645993).

[Transact-SQL Syntax Conventions](http://msdn.microsoft.com/en-us/library/ms177563).

[Syntax](javascript:void(0))

CREATE RESOURCE POOL pool\_name

[ WITH

    ( [ MIN\_CPU\_PERCENT = value ]

    [ [ , ] MAX\_CPU\_PERCENT = value ]

     [ [ , ] CAP\_CPU\_PERCENT = value ]

     [ [ , ] AFFINITY {SCHEDULER = AUTO | (Scheduler\_range\_spec) | NUMANODE = (NUMA\_node\_range\_spec)} ]

    [ [ , ] MIN\_MEMORY\_PERCENT = value ]

    [ [ , ] MAX\_MEMORY\_PERCENT = value ])

]

[;]

Scheduler\_range\_spec::=

{SCHED\_ID | SCHED\_ID TO SCHED\_ID}[,…n]

NUMA\_node\_range\_spec::=

{NUMA\_node\_ID | NUMA\_node\_ID TO NUMA\_node\_ID}[,…n]

[Arguments](javascript:void(0))

pool\_name

Is the user-defined name for the resource pool. pool\_name is alphanumeric, can be up to 128 characters, must be unique within an instance of SQL Server, and must comply with the rules for [identifiers](http://msdn.microsoft.com/en-us/library/ms175874).

MIN\_CPU\_PERCENT =value

Specifies the guaranteed average CPU bandwidth for all requests in the resource pool when there is CPU contention. value is an integer with a default setting of 0. The allowed range for value is from 0 through 100.

MAX\_CPU\_PERCENT =value

Specifies the maximum average CPU bandwidth that all requests in resource pool will receive when there is CPU contention. value is an integer with a default setting of 100. The allowed range for value is from 1 through 100.

CAP\_CPU\_PERCENT =value

Specifies a hard cap on the CPU bandwidth that all requests in the resource pool will receive. Limits the maximum CPU bandwidth level to be the same as the specified value. value is an integer with a default setting of 100. The allowed range for value is from 1 through 100.

AFFINITY {SCHEDULER = AUTO | (Scheduler\_range\_spec) | NUMANODE = (<NUMA\_node\_range\_spec>)}

Attach the resource pool to specific schedulers. The default value is AUTO.

AFFINITY SCHEDULER = (Scheduler\_range\_spec) maps the resource pool to the SQL Server schedules identified by the given IDs. These IDs map to the values in the scheduler\_id column in [sys.dm\_os\_schedulers (Transact-SQL)](http://msdn.microsoft.com/en-us/library/ms177526).

When you use AFFINITY NAMANODE = (NUMA\_node\_range\_spec), the resource pool is affinitized to the SQL Server schedulers that map to the physical CPUs that correspond to the given NUMA node or range of nodes. You can use the following Transact-SQL query to discover the mapping between the physical NUMA configuration and the SQL Server scheduler IDs.

[Copy](javascript:if%20(window.epx.codeSnippet)window.epx.codeSnippet.copyCode('CodeSnippetContainerCode_39bff252-faee-48f1-81e3-d64722be97d9');)

SELECT osn.memory\_node\_id AS [numa\_node\_id], sc.cpu\_id, sc.scheduler\_id

FROM sys.dm\_os\_nodes AS osn

INNER JOIN sys.dm\_os\_schedulers AS sc ON osn.node\_id = sc.parent\_node\_id AND sc.scheduler\_id < 1048576

MIN\_MEMORY\_PERCENT =value

Specifies the minimum amount of memory reserved for this resource pool that can not be shared with other resource pools. value is an integer with a default setting of 0 The allowed range for value is from 0 to 100.

MAX\_MEMORY\_PERCENT =value

Specifies the total server memory that can be used by requests in this resource pool. value is an integer with a default setting of 100. The allowed range for value is from 1 through 100.

[Remarks](javascript:void(0))

The values for MAX\_CPU\_PERCENT and MAX\_MEMORY\_PERCENT must be greater than or equal to the values for MIN\_CPU\_PERCENT and MIN\_MEMORY\_PERCENT, respectively.

CAP\_CPU\_PERCENT differs from MAX\_CPU\_PERCENT in that workloads associated with the pool can use CPU capacity above the value of MAX\_CPU\_PERCENT if it is available, but not above the value of CAP\_CPU\_PERCENT.

The total CPU percentage for each affinitized component (scheduler(s) or NUMA node(s)) should not exceed 100%.

[Permissions](javascript:void(0))

Requires CONTROL SERVER permission.

[Examples](javascript:void(0))

The following example shows how to create a resource pool named bigPool. This pool uses the default Resource Governor settings.

[Copy](javascript:if%20(window.epx.codeSnippet)window.epx.codeSnippet.copyCode('CodeSnippetContainerCode_07be6e54-587a-42f0-80f3-0c85f76e9e89');)

CREATE RESOURCE POOL bigPool;

GO

ALTER RESOURCE GOVERNOR RECONFIGURE;

GO

In the following example, the CAP\_CPU\_PERCENT sets the hard cap to 30% and AFFINITY SCHEDULER is set to a range of 0 to 63, 128 to 191.

[Copy](javascript:if%20(window.epx.codeSnippet)window.epx.codeSnippet.copyCode('CodeSnippetContainerCode_18a81cbb-02a5-454f-a168-f623d069e651');)

CREATE RESOURCE POOL PoolAdmin

WITH (

     MIN\_CPU\_PERCENT = 10,

MAX\_CPU\_PERCENT = 20,

CAP\_CPU\_PERCENT = 30,

AFFINITY SCHEDULER = (0 TO 63, 128 TO 191),

     MIN\_MEMORY\_PERCENT = 5,

     MAX\_MEMORY\_PERCENT = 15);

[Remarks](javascript:void(0))

The Resource Governor feature enables a database administrator to distribute server resources among resource pools, up to a maximum of 64 pools.

**sys.dm\_exec\_query\_resource\_semaphores (Transact-SQL)**

**SQL Server 2012**

[Other Versions](javascript:;)

Description: http://i.msdn.microsoft.com/Areas/Epx/Content/Images/ImageSprite.png

* [SQL Server 2008 R2](http://msdn.microsoft.com/en-us/library/ms366321(d=printer,v=sql.105))
* [SQL Server 2008](http://msdn.microsoft.com/en-us/library/ms366321(d=printer,v=sql.100))
* [SQL Server 2005](http://msdn.microsoft.com/en-us/library/ms366321(d=printer,v=sql.90))

Returns the information about the current query-resource semaphore status. **sys.dm\_exec\_query\_resource\_semaphores** provides general query-execution memory status and allows you to determine whether the system can access enough memory. This view complements memory information obtained from [sys.dm\_os\_memory\_clerks](http://msdn.microsoft.com/en-us/library/ms175019) to provide a complete picture of server memory status. **sys.dm\_exec\_query\_resource\_semaphores** returns one row for the regular resource semaphore and another row for the small-query resource semaphore.

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data type** | **Description** |
| **resource\_semaphore\_id** | smallint | Nonunique ID of the resource semaphore. 0 for the regular resource semaphore and 1 for the small-query resource semaphore. |
| **target\_memory\_kb** | bigint | Grant usage target in kilobytes. |
| **max\_target\_memory\_kb** | bigint | Maximum potential target in kilobytes. NULL for the small-query resource semaphore. |
| **total\_memory\_kb** | bigint | Memory held by the resource semaphore in kilobytes. If the system is under memory pressure or if forced minimum memory is granted frequently, this value can be larger than the **target\_memory\_kb** or **max\_target\_memory\_kb** values. Total memory is a sum of available and granted memory. |
| **available\_memory\_kb** | bigint | Memory available for a new grant in kilobytes. |
| **granted\_memory\_kb** | bigint | Total granted memory in kilobytes. |
| **used\_memory\_kb** | bigint | Physically used part of granted memory in kilobytes. |
| **grantee\_count** | int | Number of active queries that have their grants satisfied. |
| **waiter\_count** | int | Number of queries waiting for grants to be satisfied. |
| **timeout\_error\_count** | bigint | Total number of time-out errors since server startup. NULL for the small-query resource semaphore. |
| **forced\_grant\_count** | bigint | Total number of forced minimum-memory grants since server startup. NULL for the small-query resource semaphore. |
| **pool\_id** | int | ID of the resource pool to which this resource semaphore belongs. |

[Permissions](javascript:void(0))

Requires VIEW SERVER STATE permission on the server.

[Remarks](javascript:void(0))

Queries that use dynamic management views that include ORDER BY or aggregates might increase memory consumption and thus contribute to the problem they are troubleshooting.

Use **sys.dm\_exec\_query\_resource\_semaphores** for troubleshooting but do not include it in applications that will use future versions of SQL Server.

The Resource Governor feature enables a database administrator to distribute server resources among resource pools, up to a maximum of 64 pools. In SQL Server 2012, each pool behaves like a small independent server instance and requires 2 semaphores.

# sys.dm\_resource\_governor\_resource\_pool\_affinity (Transact-SQL)

**SQL Server 2012**

Tracks resource pool affinity.

|  |  |  |
| --- | --- | --- |
| **Colmn name** | **Data type** | **Description** |
| Pool\_id | int | The ID of the resource pool. Is not nullable. |
| Processor\_group | smallint | The ID of the Windows logical processor group. Is not nullable. |
| Scheduler\_mask | bigint | The binary mask representing the schedulers associated with this pool. Is not nullable. |

[Remarks](javascript:void(0))

Pools that are created with an affinity of AUTO will not appear in this view because they have no affinity. For more information, see the [CREATE RESOURCE POOL (Transact-SQL)](http://msdn.microsoft.com/en-us/library/bb895329) and [ALTER RESOURCE POOL (Transact-SQL)](http://msdn.microsoft.com/en-us/library/bb934024) statements.

**sys.dm\_resource\_governor\_resource\_pools (Transact-SQL)**

**SQL Server 2012**

[Other Versions](javascript:;)

Description: http://i.msdn.microsoft.com/Areas/Epx/Content/Images/ImageSprite.png

* [SQL Server 2008 R2](http://msdn.microsoft.com/en-us/library/bb934023(d=printer,v=sql.105))
* [SQL Server 2008](http://msdn.microsoft.com/en-us/library/bb934023(d=printer,v=sql.100))

Returns information about the current resource pool state, the current configuration of resource pools, and resource pool statistics.

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data type** | **Description** |
| pool\_id | int | The ID of the resource pool. Is not nullable. |
| name | sysname | The name of the resource pool. Is not nullable. |
| statistics\_start\_time | datetime | The time when statistics was reset for this pool. Is not nullable. |
| total\_cpu\_usage\_ms | bigint | The cumulative CPU usage in milliseconds. Is not nullable. |
| cache\_memory\_kb | bigint | The current total cache memory usage in kilobytes. Is not nullable. |
| compile\_memory\_kb | bigint | The current total stolen memory usage in kilobytes (KB). The majority of this usage would be for compile and optimization, but it can also include other memory users. Is not nullable. |
| used\_memgrant\_kb | bigint | The current total used (stolen) memory from memory grants. Is not nullable. |
| total\_memgrant\_count | bigint | The cumulative count of memory grants in this resource pool. Is not nullable. |
| total\_memgrant\_timeout\_count | bigint | The cumulative count of memory grant time-outs in this resource pool. Is not nullable. |
| active\_memgrant\_count | int | The current count of memory grants. Is not nullable. |
| active\_memgrant\_kb | bigint | The sum, in kilobytes (KB), of current memory grants. Is not nullable. |
| memgrant\_waiter\_count | int | The count of queries currently pending on memory grants. Is not nullable. |
| max\_memory\_kb | bigint | The maximum amount of memory, in kilobytes, that the resource pool can have. This is based on the current settings and server state. Is not nullable. |
| used\_memory\_kb | bigint | The amount of memory used, in kilobytes, for the resource pool. Is not nullable. |
| target\_memory\_kb | bigint | The target amount of memory, in kilobytes, the resource pool is trying to attain. This is based on the current settings and server state. Is not nullable. |
| out\_of\_memory\_count | bigint | The number of failed memory allocations in the pool. Is not nullable. |
| min\_cpu\_percent | int | The current configuration for the guaranteed average CPU bandwidth for all requests in the resource pool when there is CPU contention. Is not nullable. |
| max\_cpu\_percent | int | The current configuration for the maximum average CPU bandwidth allowed for all requests in the resource pool when there is CPU contention. Is not nullable. |
| min\_memory\_percent | int | The current configuration for the guaranteed amount of memory for all requests in the resource pool when there is memory contention. This is not shared with other resource pools. Is not nullable. |
| max\_memory\_percent | int | The current configuration for the percentage of total server memory that can be used by requests in this resource pool. Is not nullable. |
| cap\_cpu\_percent | int | Hard cap on the CPU bandwidth that all requests in the resource pool will receive. Limits the maximum CPU bandwidth level to the specified level. The allowed range for value is from 1 through 100. Is not nullable. |

[Remarks](javascript:void(0))

Resource Governor workload groups and Resource Governor resource pools have a many-to-one mapping. As a result, many of the resource pool statistics are derived from the workload group statistics.

This dynamic management view shows the in-memory configuration.To see the stored configuration metadata, use the sys.resource\_governor\_resource\_pools catalog view.

When ALTER RESOURCE GOVERNOR RESET STATISTICS is successfully executed, the following counters are reset: statistics\_start\_time, total\_cpu\_usage\_ms, and out\_of\_memory\_count. statistics\_start\_time is set to the current system date and time, the other counters are set to zero (0).

[Permissions](javascript:void(0))

Requires VIEW SERVER STATE permission.

**sys.dm\_exec\_query\_memory\_grants (Transact-SQL)**

**SQL Server 2012**

[Other Versions](javascript:;)

Description: http://i.msdn.microsoft.com/Areas/Epx/Content/Images/ImageSprite.png

* [SQL Server 2008 R2](http://msdn.microsoft.com/en-us/library/ms365393(d=printer,v=sql.105))
* [SQL Server 2008](http://msdn.microsoft.com/en-us/library/ms365393(d=printer,v=sql.100))
* [SQL Server 2005](http://msdn.microsoft.com/en-us/library/ms365393(d=printer,v=sql.90))

Returns information about the queries that have acquired a memory grant or that still require a memory grant to execute. Queries that do not have to wait on a memory grant will not appear in this view.

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data type** | **Description** |
| **session\_id** | smallint | ID (SPID) of the session where this query is running. |
| **request\_id** | int | ID of the request. Unique in the context of the session. |
| **scheduler\_id** | int | ID of the scheduler that is scheduling this query. |
| **dop** | smallint | Degree of parallelism of this query. |
| **request\_time** | datetime | Date and time when this query requested the memory grant. |
| **grant\_time** | datetime | Date and time when memory was granted for this query. NULL if memory is not granted yet. |
| **requested\_memory\_kb** | bigint | Total requested amount of memory in kilobytes. |
| **granted\_memory\_kb** | bigint | Total amount of memory actually granted in kilobytes. Can be NULL if the memory is not granted yet. For a typical situation, this value should be the same as **requested\_memory\_kb**. For index creation, the server may allow additional on-demand memory beyond initially granted memory. |
| **required\_memory\_kb** | bigint | Minimum memory required to run this query in kilobytes. **requested\_memory\_kb** is the same or larger than this amount. |
| **used\_memory\_kb** | bigint | Physical memory used at this moment in kilobytes. |
| **max\_used\_memory\_kb** | bigint | Maximum physical memory used up to this moment in kilobytes. |
| **query\_cost** | float | Estimated query cost. |
| **timeout\_sec** | int | Time-out in seconds before this query gives up the memory grant request. |
| **resource\_semaphore\_id** | smallint | Non-unique ID of the resource semaphore on which this query is waiting.   |  | | --- | | **Description: NoteNote** | | This ID is unique in versions of SQL Server that are earlier than SQL Server 2008. This change can affect troubleshooting query execution. For more information, see the "Remarks" section later in this topic. | |
| **queue\_id** | smallint | ID of waiting queue where this query waits for memory grants. NULL if the memory is already granted. |
| **wait\_order** | int | Sequential order of waiting queries within the specified **queue\_id**. This value can change for a given query if other queries get memory grants or time out. NULL if memory is already granted. |
| **is\_next\_candidate** | bit | Candidate for next memory grant.  1 = Yes  0 = No  NULL = Memory is already granted. |
| **wait\_time\_ms** | bigint | Wait time in milliseconds. NULL if the memory is already granted. |
| **plan\_handle** | varbinary(64) | Identifier for this query plan. Use **sys.dm\_exec\_query\_plan** to extract the actual XML plan. |
| **sql\_handle** | varbinary(64) | Identifier for Transact-SQL text for this query. Use **sys.dm\_exec\_sql\_text** to get the actual Transact-SQL text. |
| **group\_id** | int | ID for the workload group where this query is running. |
| **pool\_id** | int | ID of the resource pool that this workload group belongs to. |
| **is\_small** | tinyint | When set to 1, indicates that this grant uses the small resource semaphore. When set to 0, indicates that a regular semaphore is used. |
| **ideal\_memory\_kb** | bigint | Size, in kilobytes (KB), of the memory grant to fit everything into physical memory. This is based on the cardinality estimate. |

[Permissions](javascript:void(0))

Requires VIEW SERVER STATE permission on the server.

[Remarks](javascript:void(0))

A typical debugging scenario for query time-out may look like the following:

* Check overall system memory status using [sys.dm\_os\_memory\_clerks](http://msdn.microsoft.com/en-us/library/ms175019), [sys.dm\_os\_sys\_info](http://msdn.microsoft.com/en-us/library/ms175048), and various performance counters.
* Check for query-execution memory reservations in **sys.dm\_os\_memory\_clerks** where type = 'MEMORYCLERK\_SQLQERESERVATIONS'.
* Check for queries waiting for grants using **sys.dm\_exec\_query\_memory\_grants**.
* Further examine memory-intensive queries using [sys.dm\_exec\_requests](http://msdn.microsoft.com/en-us/library/ms177648).
* If a runaway query is suspected, examine the Showplan from [sys.dm\_exec\_query\_plan](http://msdn.microsoft.com/en-us/library/ms189747) and batch text from [sys.dm\_exec\_sql\_text](http://msdn.microsoft.com/en-us/library/ms181929).

Queries that use dynamic management views that include ORDER BY or aggregates may increase memory consumption and thus contribute to the problem they are troubleshooting.

The Resource Governor feature enables a database administrator to distribute server resources among resource pools, up to a maximum of 64 pools. Beginning with SQL Server 2008, each pool behaves like a small independent server instance and requires 2 semaphores. The number of rows that are returned from **sys.dm\_exec\_query\_resource\_semaphores** can be up to 20 times more than the rows that are returned in SQL Server 2005.

**sys.resource\_governor\_workload\_groups (Transact-SQL)**

**SQL Server 2012**

[Other Versions](javascript:;)

Description: http://i.msdn.microsoft.com/Areas/Epx/Content/Images/ImageSprite.png

* [SQL Server 2008 R2](http://msdn.microsoft.com/en-us/library/bb895237(d=printer,v=sql.105))
* [SQL Server 2008](http://msdn.microsoft.com/en-us/library/bb895237(d=printer,v=sql.100))

Returns the stored workload group configuration. Each workload group can subscribe to one and only one resource pool.

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data type** | **Description** |
| group\_id | int | Unique ID of the workload group. Is not nullable. |
| name | sysname | Name of the workload group. Is not nullable. |
| importance | sysname | Is the relative importance of a request in this workload group. Importance is one of the following, with MEDIUM being the default:   * LOW * MEDIUM * HIGH   Is not nullable.   |  | | --- | | **Note** | | Importance only applies to workload groups in the same resource pool. | |
| request\_max\_memory\_grant\_percent | int | Maximum memory grant, as a percentage, for a single request. The default value is 25. Is not nullable.   |  | | --- | | **Note** | | If this setting is higher than 50 percent, large queries will run one at a time. Therefore, there is greater risk of getting an out-of-memory error while the query is running. | |
| request\_max\_cpu\_time\_sec | int | Maximum CPU use limit, in seconds, for a single request. The default value, 0, specifies no limit. Is not nullable.   |  | | --- | | **Note** | | For more information, see [CPU Threshold Exceeded Event Class](http://msdn.microsoft.com/en-us/library/bb934148). | |
| request\_memory\_grant\_timeout\_sec | int | Memory grant time-out, in seconds, for a single request. The default value, 0, uses an internal calculation based on query cost. Is not nullable. |
| max\_dop | int | Maximum degree of parallelism for the workload group. The default value, 0, uses global settings. Is not nullable.   |  | | --- | | **Note** | | This setting will override the query option **maxdop**. | |
| group\_max\_requests | int | Maximum number of concurrent requests. The default value, 0, specifies no limit. Is not nullable. |
| pool\_id | int | ID of the resource pool that this workload group uses. |
| group\_min\_memory\_percent | int | Percentage of MIN\_MEMORY\_PERCENT, which is the minimum amount of memory reserved for the resource pool that cannot be shared with other resource pools. This value provides a guaranteed minimum for a group. |

[Remarks](javascript:void(0))

The catalog view displays the stored metadata. To see the in-memory configuration, use the corresponding dynamic management view.

The stored and in-memory configuration can be different if the Resource Governor configuration has been changed but the ALTER RESOURCE GOVERNOR RECONFIGURE statement has not been applied.

[Permissions](javascript:void(0))

Requires VIEW ANY DEFINITION permission to view contents, requires CONTROL SERVER permission to change contents.

**Backup and Restore Enhancements**

**SQL Server 2012**

The enhancements introduced in SQL Server 2012 SP1 Cumulative Update 2, enable backup and restore from the Windows Azure Blob storage service from SQL Server using TSQL or SMO. To request the update from Microsoft Customer Support, see [this article](http://go.microsoft.com/fwlink/?LinkId=271608).

PowerShell support for SQL Server backup and restore with the Windows Azure Blob storage service is added in SQL Server 2012 SP1 Cumulative Update 4. To request the update from Microsoft Customer Support, see [this knowledgebase article](http://go.microsoft.com/fwlink/?LinkId=296580).

For more information on using PowerShell to backup to Windows Azure Blob storage service, see [Use PowerShell to Backup Multiple Databases to Windows Azure Blob Storage Service](http://msdn.microsoft.com/en-us/library/dn223322).

[SQL Server Backup and Restore with the Windows Azure Blob Storage Service](javascript:void(0))

Expanding on current backup storage options, native SQL Server backups can now be stored in Windows Azure Blob storage. This enhancement can be used to backup SQL Server databases on an on-premises instance or an instance of SQL Server running a hosted environment such as Windows Azure Virtual Machine. With this new functionality, you can now back up directly to and restore from the Windows Azure Blob storage. Backing up to the Windows Azure Blob storage service has specific benefits. It can be an offsite storage option for the on-premises database backups, or used when you migrate an on-premises database to an instance of SQL Server running in a Windows Azure Virtual Machine. For more details on benefits, see [SQL Server Backup and Restore with Windows Azure Blob Storage Service](http://msdn.microsoft.com/en-us/library/jj919148).

The following are limitations specific to this release:

* The maximum backup size supported is 1 TB.
* In this implementation, you can issue backup or restore statements by using TSQL or SMO. A backup to or restoring from the Windows Azure Blob storage service by using SQL Server Management Studio Backup or Restore wizard is not currently enabled.
* For more information and other limitations, see [SQL Server Backup and Restore with Windows Azure Blob Storage Service](http://msdn.microsoft.com/en-us/library/jj919148).

[Links to more resources](javascript:void(0))

* [SQL Server Backup and Restore with Windows Azure Blob Storage Service](http://msdn.microsoft.com/en-us/library/jj919148)
* [Tutorial: Getting Started with SQL Server Backup and Restore to Windows Azure Blob Storage Service](http://msdn.microsoft.com/en-us/library/jj720558)
* [Backup and Restore Best Practices (Windows Azure Blob Storage Service)](http://msdn.microsoft.com/en-us/library/jj919149)

# Microsoft.SqlServer.Dac Namespace

**SQL Server 2012**

This namespace provides classes you can use in your code to perform operations on DACPAC and BACPAC packages.  These packages can be extracted and exported from existing databases or used to deploy and import new databases. DAPCAC packages can also be used to upgrade an existing database.

See also [Understanding Data-tier Applications](http://msdn.microsoft.com/en-us/library/ee240739.aspx).

[Classes](javascript:void(0))

|  |  |  |
| --- | --- | --- |
|  | **Class** | **Description** |
| Description: Public class | [BacPackage](http://msdn.microsoft.com/en-us/library/microsoft.sqlserver.dac.bacpackage.aspx) | Representation of the artifact that contains the definition and data of a data-tier application. |
| Description: Public class | [DacAzureDatabaseSpecification](http://msdn.microsoft.com/en-us/library/microsoft.sqlserver.dac.dacazuredatabasespecification.aspx) | Defines optional parameters specific to a Windows Azure SQL Database. |
| Description: Public class | [DacDataProgressMessage](http://msdn.microsoft.com/en-us/library/microsoft.sqlserver.dac.dacdataprogressmessage.aspx) | Data associated with an executing data operation to report status updates or errors related to progress. |
| Description: Public class | [DacDeployOptions](http://msdn.microsoft.com/en-us/library/microsoft.sqlserver.dac.dacdeployoptions.aspx) | Defines options that affect the behavior of package deployment to a database. |
| Description: Public class | [DacExceptionMessage](http://msdn.microsoft.com/en-us/library/microsoft.sqlserver.dac.dacexceptionmessage.aspx) | Data associated with an error with exception information. |
| Description: Public class | [DacExtractOptions](http://msdn.microsoft.com/en-us/library/microsoft.sqlserver.dac.dacextractoptions.aspx) | Defines options that affect the behavior of package extraction. |
| Description: Public class | [DacMessage](http://msdn.microsoft.com/en-us/library/microsoft.sqlserver.dac.dacmessage.aspx) | Data associated with an executing operation to report status updates or errors. |
| Description: Public class | [DacMessageEventArgs](http://msdn.microsoft.com/en-us/library/microsoft.sqlserver.dac.dacmessageeventargs.aspx) | Data associated with event raised as an executing operation reports status updates or errors. |
| Description: Public class | [DacOperationProgressMessage](http://msdn.microsoft.com/en-us/library/microsoft.sqlserver.dac.dacoperationprogressmessage.aspx) | Progress message associated with the overall progress of a DAC operation. |
| Description: Public class | [DacPackage](http://msdn.microsoft.com/en-us/library/microsoft.sqlserver.dac.dacpackage.aspx) | Representation of the artifact that contains the definition of a data-tier application. |
| Description: Public class | [DacPackageExtensions](http://msdn.microsoft.com/en-us/library/microsoft.sqlserver.dac.dacpackageextensions.aspx) | Provides methods to support manipulation of DacPackage objects. These methods work on an existing [DacPackage](http://msdn.microsoft.com/en-us/library/microsoft.sqlserver.dac.dacpackage.aspx). |
| Description: Public class | [DacProfile](http://msdn.microsoft.com/en-us/library/microsoft.sqlserver.dac.dacprofile.aspx) | Contains settings related to a specific deployment configuration. |
| Description: Public class | [DacProgressEventArgs](http://msdn.microsoft.com/en-us/library/microsoft.sqlserver.dac.dacprogresseventargs.aspx) | Data associated with event raised as an executing operation completes individual steps of processing. |
| Description: Public class | [DacRetryMessage](http://msdn.microsoft.com/en-us/library/microsoft.sqlserver.dac.dacretrymessage.aspx) | Indicates that the DacFX API retried a command or connection due to a transient error returned by the server. |
| Description: Public class | [DacServices](http://msdn.microsoft.com/en-us/library/microsoft.sqlserver.dac.dacservices.aspx) | Provides operations to create and consume DAC packages represented by [DacPackage](http://msdn.microsoft.com/en-us/library/microsoft.sqlserver.dac.dacpackage.aspx) instances. |
| Description: Public class | [DacServicesException](http://msdn.microsoft.com/en-us/library/microsoft.sqlserver.dac.dacservicesexception.aspx) | Exception thrown for DAC service errors. |
| Description: Public class | [PackageMetadata](http://msdn.microsoft.com/en-us/library/microsoft.sqlserver.dac.packagemetadata.aspx) | Metadata information that describes a package. |
| Description: Public class | [PackageOptions](http://msdn.microsoft.com/en-us/library/microsoft.sqlserver.dac.packageoptions.aspx) | Defines advanced options and additional artifacts for package creation. |

[Enumerations](javascript:void(0))

|  |  |  |
| --- | --- | --- |
|  | **Enumeration** | **Description** |
| Description: Public enumeration | [DacAzureEdition](http://msdn.microsoft.com/en-us/library/microsoft.sqlserver.dac.dacazureedition.aspx) | List of possible editions for a Windows Azure SQL Database. |
| Description: Public enumeration | [DacMessageType](http://msdn.microsoft.com/en-us/library/microsoft.sqlserver.dac.dacmessagetype.aspx) | Specifies the type (or severity) of message associated with a given event. |
| Description: Public enumeration | [DacOperationStatus](http://msdn.microsoft.com/en-us/library/microsoft.sqlserver.dac.dacoperationstatus.aspx) | Specifies the state of a service operation. |
| Description: Public enumeration | [DacSchemaModelStorageType](http://msdn.microsoft.com/en-us/library/microsoft.sqlserver.dac.dacschemamodelstoragetype.aspx) | Specifies the type of backing storage for a schema model. |

# Microsoft.SqlServer.Dac.Extensions Namespace

**SQL Server 2012**

This namespace provides classes containing extension methods you can use in your code to retrieve information from DACPAC and BACPAC packages or utilize the expanded functionality not currently present in the [Microsoft.SqlServer.Dac](http://msdn.microsoft.com/en-us/library/microsoft.sqlserver.dac.aspx) namespace. Classes in this namespace may change over time as new classes are added or existing classes are removed and incorporated into the [Microsoft.SqlServer.Dac](http://msdn.microsoft.com/en-us/library/microsoft.sqlserver.dac.aspx) namespace.

See also [Understanding Data-tier Applications](http://msdn.microsoft.com/en-us/library/ee240739.aspx).

[Classes](javascript:void(0))

|  |  |  |
| --- | --- | --- |
|  | **Class** | **Description** |
| Description: Public class | [DacExtensions](http://msdn.microsoft.com/en-us/library/microsoft.sqlserver.dac.extensions.dacextensions.aspx) | Provides extensions for interacting with DAC packages represented by [DacPackage](http://msdn.microsoft.com/en-us/library/microsoft.sqlserver.dac.dacpackage.aspx) or [BacPackage](http://msdn.microsoft.com/en-us/library/microsoft.sqlserver.dac.bacpackage.aspx) instances. |