SQL Server 2008 R2 Books Online (November CTP)
What's New (Database Engine)
[This topic is pre-release documentation and is subject to change in future releases. Blank topics are included as placeholders.]

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
Manageability Enhancements (Database Engine)	SQL Server 2008 R2 introduces the SQL Server Utility, which provides the ability to enroll data-tier applications and instances of SQL Server into the SQL Server Utility for centralized management and consolidation. Another addition is the data-tier application (DAC), which provides a single unit of deployment and management for the data-tier objects used by an application.
Availability Enhancements (Database Engine)	The availability of Microsoft SQL Server 2008 databases is improved by enhancements to database mirroring. Database mirroring enables the creation of hot standby servers that provide rapid failover support with no loss of data from committed transactions.
Programmability Enhancements (Database Engine)	Programmability enhancements in the Database Engine include new data storage features, new data types, new full-text search architecture, and numerous improvements and additions to Transact-SQL.
Scalability and Performance Enhancements (Database Engine)	Scalability and performance enhancements in the Database Engine include filtered indexes and statistics, new table and query hints, and new query performance and query processing features.
Security Enhancements (Database Engine)	Security enhancements in the Database Engine include new encryption functions, the transparent data encryption and extensible key management features, and a clarification of DES algorithms.

SQL Server 2008 R2 Books Online (November CTP)
Manageability Enhancements (Database Engine)
[This topic is pre-release documentation and is subject to change in future releases. Blank topics are included as placeholders.]

SQL Server 2008 R2 introduces the SQL Server Utility for managing multiple instances of the SQL Server Database Engine. It also introduces a unit of management called a data-tier application (DAC) that provides an application-based view for managing the data-tier objects in the SQL Server Utility or stand-alone instances of the Database Engine.

What's New in SQL Server 2008 R2 November CTP

Note:

Because SQL Server 2008 R2 is a minor version upgrade of SQL Server 2008, we recommend that you also review the content in the SQL Server 2008 section.

Connectivity to SQL Azure

SQL Server 2008 R2 introduces the ability to connect to SQL Azure Database from the client utilities:

- The Generate and Publish Scripts Wizard can use SQL Azure as both the source and destination for the scripts it publishes. For more information, see Using the Generate and Publish Scripts Wizard.

Data-tier Application Upgrade

The November CTP introduces several new features for the data-tier applications introduced in the August CTP:

- You can upgrade a deployed DAC from one version to another. For more information, see Upgrading Data-tier Applications.
- You can unpack a DAC package to review the contents before using the package to deploy or upgrade the DAC. Also, the DAC package file is now a zipped file containing multiple XML files. For more information, see Viewing and Comparing Data-tier Applications.
- You can now register a DAC for an existing database, creating a DAC instance. For more information, see How to: Register a DAC (SQL Server Management Studio).

SQL Server Utility

The November CTP provides enhanced functionality for SQL Server Utility features introduced in the August CTP:

- A Getting Started page in SQL Server Management Studio.
- Improved performance and scale.
- Improved workflow for removing a managed instance of SQL Server from the SQL Server Utility.
- Sample scripts for create UCP, enroll instance, and remove instance operations.

Network Connectivity

The VIA protocol is deprecated. This feature will be removed in a future version of Microsoft SQL Server. Avoid using this feature in new development work, and plan to modify applications that currently use this feature..

What's New in SQL Server 2008 R2 August CTP

SQL Server Utility

The SQL Server Utility enables:

- Creating a SQL Server utility control point (UCP): Install a single instance of the SQL Server 2008 R2 Database Engine and then promote it to be the UCP. The UCP is the central repository for configuration and performance data collected for all the instances enrolled in the SQL Server Utility. The UCP is the central reasoning point for the SQL Server Utility. It supports actions such as applying central policies, or analyzing resource utilization trends of an instance to predict when it might exceed central resource utilization polices.
- Using Utility Explorer in SQL Server Management Studio to enroll existing SQL Server 2008 R2 data-tier applications and instances of the Database Engine into the SQL Server Utility for centralized management.
- Setting central policies that tailor and control the operation of the SQL Server Utility and its dashboards.
- Using the Utility Explorer to display a dashboard and detailed viewpoints that report the resource utilization, resource health, and configuration information of all the instances enrolled in the SQL Server Utility. The dashboards let you quickly identify data-tier applications and instances of the Database Engine that are either underutilizing or overutilizing their underlying hardware resources. You can then develop a plan to maximize the use of your servers, such as consolidating underutilized instances or databases to a single server.

For more information about the SQL Server Utility, see:

- Overview of SQL Server Utility.
- How to: Create a SQL Server Utility Control Point (SQL Server Utility).
- How to: Enroll an Instance of SQL Server (SQL Server Utility).
- Monitoring Instances of SQL Server in the SQL Server Utility.

Data-Tier Application

A data-tier application simplifies the development, deployment, and management of the data-tier objects that support a multi-tier or client-server application. A DAC defines all of the Database Engine schema and instance objects, such as tables, views, and logins, required to support the application. The DAC operates as a single unit of management through the development, deployment, and management lifecycle of the associated application. The DAC also contains policies that define the deployment prerequisites for the DAC.

For more information about DACs, see:

- Designing and Implementing Data-tier Applications.
- Deploying Data-tier Applications.
- Managing Data-tier Applications.

SQL Server PowerShell Provider

SQL Server 2008 R2 introduces new SQLSERVER:\Utility and SQLSERVER:\DAC folders to support the SQL Server Utility and data-tier applications in PowerShell scripts. For more information, see Using the SQL Server PowerShell Provider.

What's New in SQL Server 2008

Database Administration

Auditing

SQL Server Audit is a new feature of SQL Server 2008 that lets you create customized audits of Database Engine events. SQL Server Audit uses extended events to record the information for the audit and provides the tools and processes you must have to enable, store, and view audits on various server and database objects. For more information, see Auditing (Database Engine).

Backup Compression

Backup compression was introduced in SQL Server 2008 Enterprise. Beginning in SQL Server 2008 R2, backup compression is supported by SQL Server 2008 R2 Standard and all higher editions. Every edition of SQL Server 2008 can restore a compressed backup. You can change the backup compression behavior for an individual backup, backup job, or log shipping configuration. For more information, see Backup Compression (SQL Server).

By default, backup compression significantly increases CPU usage, which can adversely impact concurrent operations. You can create low-priority compressed backups in a session whose CPU usage is limited by Resource Governor. For more information, see How to: Use Resource Governor to Limit CPU Usage by Backup Compression (Transact-SQL).

Change Data Capture

Change data capture offers an effective solution to the challenge of efficiently performing incremental loads from source tables to data marts and data warehouses. Change data captures insert, update, and delete activity applied to SQL Server tables, and makes the details of the changes available in an easily consumed relational format. The change tables used by change data capture contain columns that mirror the column structure of a tracked source table, along with the metadata needed to understand the changes that have occurred. For more information, see Basics of Change Data Capture.

Change Tracking

SQL Server change tracking allows applications to obtain incremental changes to user tables. Where two-way synchronization is required, change tracking also allows applications to

check for data conflicts. With change tracking integrated into SQL Server 2008, developers no longer have to create complicated custom change-tracking solutions.

Prior to the integration of change tracking capabilities into SQL Server, developers often created custom change tracking solutions that used a combination of triggers, timestamp columns, other additional columns, and additional tables. Now, developing synchronization applications is easier and faster.

For more information, see Change Tracking.

Data Collector

SQL Server 2008 introduces a data collector that you can use to obtain and save data that is gathered from several sources. The data collector provides data collection containers that you can use to determine the scope and frequency of data collection on a SQL Server server system. For more information, see Introducing the Data Collector.

Detecting Edition-Related Database Migration Problems

Use the sys.dm_db_persisted_sku_features (Transact-SQL) view to identify features that would prevent a database from moving to a different edition of SQL Server 2008.

Events and Performance Counters

Deprecation Counters

Performance counters and trace events are available to track the usage of deprecated features. For more information, see SQL Server, Deprecated Features Object and Deprecated Database Engine Features in SQL Server 2008 R2.

DDL Triggers and Event Notifications

The class of events on which you can create DDL triggers and event notifications is expanded to include numerous stored procedures that perform DDL-like operations. For a list of events, see DDL Events and DDL Event Groups. Additionally, the XML schema for events is installed with the Database Engine and is also available on the Internet. For more information, see EVENTDATA (Transact-SQL).

sp_configure Options

The **access check cache quota** and **access check cache bucket count** options control the number of entries and number of hash buckets used for **access check result cache**. For more information, see access check cache Options.

Server Administration

Central Management Servers

SQL Server 2008 introduces a new method of administering multiple servers by enabling you to designate Central Management Servers. An instance of SQL Server that is designated as a

Central Management Server maintains a list registered servers. For more information, see Administering Multiple Servers Using Central Management Servers.

Dynamic Management Views

There are five new dynamic management views to present memory information:

- sys.dm_os_memory_brokers (Transact-SQL)
- sys.dm_os_memory_nodes (Transact-SQL)
- sys.dm_os_nodes (Transact-SQL)
- sys.dm_os_process_memory (Transact-SQL)
- sys.dm_os_sys_memory (Transact-SQL)

The **sys.dm_os_sys_info** dynamic management view has discontinued the **cpu_ticks_in_ms** column, and has added two new columns, **sqlserver_start_time_ms_ticks** and **sqlserver_start_time**.

Hot Add CPU

SQL Server 2008 supports dynamically adding CPUs to a running system. Adding CPUs can occur physically by adding new hardware, logically by online hardware partitioning, or virtually through a virtualization layer. For more information, see Hot Add CPU.

Optimize for ad hoc workloads Option

The optimize for ad hoc workloads option is a new server configuration option used to improve the efficiency of the plan cache for workloads that contain many single use ad hoc batches. When this option is set to 1, the Database Engine stores a small compiled plan stub in the plan cache when a batch is compiled for the first time, instead of the full compiled plan. This helps to relieve memory pressure by not allowing the plan cache to become filled with compiled plans that are not reused. For more information, see Setting Server Configuration Options.

Resource Governor

Resource Governor is a feature that you can use to manage SQL Server workload and system resource consumption. Resource Governor enables you to limit the amount of CPU and memory that incoming application requests can use. For more information, see Managing SQL Server Workloads with Resource Governor.

SQL Server Extended Events

SQL Server 2008 introduces SQL Server Extended Events, an event infrastructure for server systems. This release of Extended Events enables you to open windows into the run time of the host process by using events as trace points. Those events can then be aggregated in memory, sent to a file, or output to Event Tracing for Windows (ETW). For more information, see Introducing SQL Server Extended Events.

Policy-Based Management Administration

As part of Microsoft's ongoing effort to reduce the total cost of ownership, SQL Server 2008 introduces Policy-Based Management, a new framework for managing SQL Server. A policy contains a check condition, which is the state that the policy is evaluating, and a filter condition, which is the target set that is being evaluated. For example, an administrator can set policies that SQL Mail is off for all instances of SQL Server, or that tables in the Marketing schema in the AdventureWorks database should have names that start with "mk_pr_".

Policy-Based Management delivers the following benefits:

- Ensures compliance with policies for system configuration.
- Prevents or monitors changes to the system by authoring policies against the configuration.
- Reduces total cost of ownership by simplifying administration tasks.
- Detects compliance issues in SQL Server Management Studio.
- Allows policies to run at the same time on multiple servers:
- Includes built-in functions and the ability to execute user-defined Transact-SQL queries and user-defined WMI/WQL queries for condition expressions that allow policies to apply to specified system states and target sets.
- Includes more than 40 out-of-the-box policies that correspond to the rules from the Best Practices Analyzer and default system state settings from Surface Area Configuration. Policies must be imported to an instance of the Database Engine. For more information, see How to: Export and Import a Policy-Based Management Policy.

In addition to importing out-of-the-box policies, new policies can be created directly from the **File/New** menu.

For more information and a tutorial about how to use Policy-Based Management, see Administering Servers by Using Policy-Based Management.

SQL Server Management Studio

Query Editor

Transact-SQL Debugger

The Database Engine Query Editor now includes a Transact-SQL debugger similar to the Visual Studio debuggers. The Transact-SQL debugger helps you find problems in your Transact-SQL code by pausing execution on specific statements and then displaying data values and system information, such as the Transact-SQL call stack and the values stored in variables and parameters. For more information, see Using the Transact-SQL Debugger.

IntelliSense

The Database Engine Editor now provides IntelliSense functionality such as word completion, error underlining, parameter help, colorizing, Quick Info, outlining, and syntax pair matching. IntelliSense is provided for frequently used Transact-SQL elements. It will be extended to other Transact-SQL elements in future releases. For more information, see Using IntelliSense.

Database Engine Error List Window

SQL Server Management Studio includes an Error List window that displays the syntax and semantic errors generated from the IntelliSense code in the Database Engine Query Editor. For more information, see Error List Window (Management Studio).

Object Explorer

The Object Explorer Details window of SQL Server Management Studio has been enhanced in the following ways:

- You can customize the columns that are displayed by the Object Explorer Details window. To see a list of possible columns for the selected object type, right-click the column headings. To reorder the display, you can drag the columns.
- The properties of a selected item appear at the bottom of the Object Explorer Details window. To see additional properties, make the status bar larger by dragging the topic border of the status bar upward.

The code editor for Transact-SQL and MDX has a color-coded status bar at the bottom. The status bar provides information about the editor connection, and changes color when a code editor has more than one connection.

You can add or remove the names of items from the title bar of the code editor windows. To customize the tab name, on the **Tools** menu, click **Options**, and then select the **Text Editor** and **Editor Tab and Status Bar Page**.

Management Studio includes an enhanced details display for items that are selected in the Object Explorer. Columns can be resized, sorted and reorganized; and new navigation buttons enable you to move to a parent object or the last object visited. You can select multiple objects in the details area to perform actions on them individually or as a group.

Synchronization from an object in Object Explorer Details now selects the individual object in Object Explorer instead of its parent object. In addition, you can select columns to display from a list of properties, and the sorting and selections stay constant as you navigate. Also, for a selected object, a list of properties is shown in a sizeable area at the bottom of the details area. For more information, see Using Object Explorer Details and SQL Server Object Search.

Service Broker Nodes in Object Explorer

Object Explorer has a new node for Service Broker conversation priorities. In addition, the other Service Broker nodes now display additional menu options, including **Properties**, when you right-click the nodes. For more information, see Service Broker Object Properties F1 Help.

Customer-Requested Improvements

SQL Server Management Studio has the following customer-requested improvements:

- In the Query Editor window, you can query multiple servers at the same time by opening query windows from registered server groups. The query results can be combined into a single results pane, or can be returned in separate results panes. For

more information, see Administering Multiple Servers Using Central Management Servers.

- You can access SQL Server Profiler from the Query Editor window, from the **Query** menu, the Query Editor toolbar, or by pressing CTRL+ALT+P.
- You can now open tables by using a Return Top *n* rows option.
- You can configure the number of rows that are returned when you are opening tables.
- On the **Tools/Option** menu, you can specify the action that results by double-clicking tables in Object Explorer.
- You can block the table designer from re-creating tables when you are implementing design changes.
- The **Object Details** page can be customized with additional columns and now supports search. For more information, see Using Object Explorer Details and SQL Server Object Search.
- Additional connection parameters can be added to the connection string when you are connecting by using SQL Server Management Studio. For more information, see Connect to Server (Additional Connection Parameters Page).

SQL Server PowerShell Provider and Cmdlets

SQL Server 2008 introduces two PowerShell snap-in dll files that implement the following:

- A SQL Server PowerShell provider that exposes the SQL Server Management Object models as paths similar to file system paths. Each node in the path is associated with a SQL Server Management Object class.
- A set of SQL Server PowerShell cmdlets for tasks such as running **sqlcmd** scripts, evaluating policies, or encoding delimited identifiers for use in PowerShell paths.

In addition, there is a new **sqlps** utility that launches a PowerShell environment configured with the SQL Server features such as the provider and snap-ins. A PowerShell subsystem has been added to SQL Server Agent so that you can run PowerShell scripts on a schedule or in response to Database Engine events. You can launch SQL Server PowerShell sessions by right-clicking Object Explorer nodes in SQL Server Management Studio. For more information, see SQL Server PowerShell Overview.