# logo-sql08.gif

Overview: Upgrading from SQL Server 2000 to   
SQL Server 2008

SQL Server Technical Article

**Published:** February 2009

**Applies to:** SQL Server 2008

**Summary:** Upgrading to Microsoft® SQL Server® 2008 provides organizations with improved scalability, security, and manageability for their databases built on the trusted fundamentals of the Microsoft® SQL Server® platform. This white paper discusses the factors that companies should take into consideration as they plan to upgrade to SQL Server 2008 as well as the tools that Microsoft and Microsoft partners provide to help make upgrading easier and more successful.

Copyright

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This white paper is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in, or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

© 2009 Microsoft Corporation. All rights reserved.

Microsoft, Access, Active Directory, ActiveX, Excel, Forefront, Internet Explorer, MSDN, SharePoint, SQL Server, Visual Studio, Windows, Windows PowerShell, Windows Server, and Windows Vista are trademarks of the Microsoft group of companies.

All other trademarks are property of their respective owners.

Contents

[Introduction: Why Upgrade to SQL Server 2008? 1](#_Toc222633091)

[Return on Investment 2](#_Toc222633092)

[Microsoft Data Platform Vision 2](#_Toc222633093)

[Trusted Database Platform 3](#_Toc222633094)

[Increased Productivity for Database Workloads 3](#_Toc222633095)

[Intelligent Data Platform 4](#_Toc222633096)

[Feature Changes in SQL Server 2008 5](#_Toc222633097)

[Support 5](#_Toc222633098)

[Key Considerations in Upgrading from SQL Server 2000 to SQL Server 2008 6](#_Toc222633099)

[Upgrade Options 6](#_Toc222633100)

[Side-by-Side Upgrade vs. In-Place Upgrade 7](#_Toc222633101)

[In-Place Upgrade 7](#_Toc222633102)

[Side-by-Side Upgrade 8](#_Toc222633103)

[Comparing In-Place and Side-by-side Upgrade Methods 9](#_Toc222633104)

[Upgrade Strategy Overview 10](#_Toc222633105)

[Functionality Considerations 14](#_Toc222633106)

[Minimizing Downtime 14](#_Toc222633107)

[Application Backward Compatibility 14](#_Toc222633108)

[Discontinued Features 15](#_Toc222633109)

[Deprecated Features 15](#_Toc222633110)

[Breaking Changes 15](#_Toc222633111)

[Behavior Changes 16](#_Toc222633112)

[Other SQL Server Services Affected by Upgrade 16](#_Toc222633113)

[SQL Server Upgrade Tools 16](#_Toc222633114)

[Primary Tools 16](#_Toc222633115)

[SQL Server 2008 Upgrade Advisor 16](#_Toc222633116)

[DTS xChange 17](#_Toc222633117)

[Secondary Tools 18](#_Toc222633118)

[Microsoft Assessment and Planning Toolkit 3.2 18](#_Toc222633119)

[SQL Server 2008 Upgrade Assistant 19](#_Toc222633120)

[Best Practices Analyzer for SQL Server 2000 19](#_Toc222633121)

[System Configuration Checker (SQL Server 2008 Setup During In-Place Upgrade) 19](#_Toc222633122)

[SQL Server Profiler 20](#_Toc222633123)

[SQL Server: Deprecated Features Object Counter in System Monitor 21](#_Toc222633124)

[SQL Server Data Backup 21](#_Toc222633125)

[Other Tools 21](#_Toc222633126)

[Conclusion 22](#_Toc222633127)

[Appendix A: Discontinued Features in SQL Server 2008 23](#_Toc222633128)

[Appendix B: Deprecated Features in SQL Server 2008 26](#_Toc222633129)

[See Also – Breaking Changes: 35](#_Toc222633130)

[See Also – Behavior Changes 35](#_Toc222633131)

[Related Links 37](#_Toc222633132)

# Introduction: Why Upgrade to SQL Server 2008?

Upgrading to Microsoft® SQL Server® 2008 provides organizations with next-generation database capabilities built on the trusted fundamentals of SQL Server. Upgrading to SQL Server 2008 provides these benefits through a return on your upgrade investment that can pay for itself in as little as six months.

More than ever, organizations rely on data storage and analysis for critical business operations. Moreover, to take advantage of new opportunities in today's fast-moving business world, companies need the ability to create and deploy data-driven solutions quickly for end users throughout the enterprise. SQL Server 2008 provides a trusted, productive, and intelligent data platform that enables you to run your most demanding mission-critical applications, reduce time and cost of development and management of applications, and deliver actionable insight to your entire organization. While conceptually an upgrade to Microsoft SQL Server 2008 may seem trivial, one may run into unexpected issues or barriers if an appropriate level of planning is skipped. This white paper will explore some of the reasons to upgrade from Microsoft® SQL Server®2000 to SQL Server 2008 as well as common upgrade strategies, considerations, and tools provided by Microsoft and Microsoft partners to make upgrading easier and more successful.

At the most fundamental level, a database must protect the data stored within it. SQL Server 2008 equips organizations with a highly secure, reliable, and scalable platform on which to run their mission-critical applications. New security tools like transparent data encryption, read-based auditing, and policy-based management allow organizations to control and protect data. Transparent data encryption permits organizations to maintain regulatory compliance and store confidential information with greater peace of mind, and policy-based permissions allow much more modular control over access to data.

Microsoft has taken significant care to make the database more productive and reduce the cost of managing your data infrastructure while streamlining development of database applications. Policy-based permissions management reduces the cost and effort of administering your data infrastructure. Organizations will be able to benefit from more sophisticated data types and programming models within the entity framework that streamline development to realize new business opportunities. Features like Resource Governor allow complete control over database load by allowing prioritization of all database activity. This allows organizations to fine-tune and scale the database with much more control than in the past.

SQL Server 2008 enjoys a number of intelligent performance enhancements made throughout the technology stack, including enhancements within Analysis Services, Reporting Services, and Integration Services. SQL Server 2008 Analysis Services provides a comprehensive and scalable analysis platform for analyzing performance indicators and business metrics through the intuitive interface of Microsoft® Office Excel® 2007. SQL Server 2008 Reporting Services empowers users to produce reports and visualizations in Microsoft® Office Word 2007 and Excel 2007 without involving IT staff. SQL Server 2008 Integration Services can provide record-setting extract, transform, and load (ETL) performance and can integrate growing volumes of data from disparate systems.

## Return on Investment

In an increasingly competitive business environment, every IT investment decision needs to support itself through reduced costs and increased productivity. Upgrading to SQL Server 2008 is no different and can provide surprising returns for your business.

In a September 2008 Forrester® Total Economic Impact™ study commissioned by Microsoft, Forrester interviewed a business that had upgraded to SQL Server 2008. Focusing only on readily quantifiable benefits such as avoiding additional SQL Server licenses and related hardware, removing third-party software, and employee savings and productivity, Forrester discovered that the business interviewed had realized a three-year return on investment between 162 and 181 percent. For this company, that equaled a payback period on their upgrade between four and six months. Considering the additional benefits provided by upgrading to SQL Server 2008 not covered in this study, the benefits to this company were even greater. Download the full report to get a better sense of the economic returns of upgrading to SQL Server 2008 for your company: <http://download.microsoft.com/download/d/1/1/d11349b8-af33-45c4-a89c-f0dc64bbd431/TEI%20of%20SQL%20Server%202008%20Upgrade.pdf>.

## Microsoft Data Platform Vision

The amount and variety of information that organizations need to work with continues to explode. Myriad new forms of information are becoming integral to business operations, from digitized images and video to sensor information from radio-frequency identification (RFID) tags. At the same time, growing regulatory compliance in a globalized business world requires that organizations store more information securely and keep it available at all times. In parallel with these driving needs, the cost of disk storage has dramatically decreased, enabling businesses to store more data per dollar invested. And fundamentally, users and decision makers must be able to sift quickly through mountains of data to find relevant information to gain business insight.

Microsoft created its Data Platform Vision to frame these needs faced by businesses and to provide a solution for organizations to meet these needs. The driving goal of the Data Platform Vision is to provide businesses with tools so they can store and manage disparate types of data—including XML, e-mail, time/calendar, file, document, and geospatial—while providing a rich set of services to interact with the data: search, query, data analysis, reporting, data integration, and robust synchronization.

SQL Server 2008 delivers on the Microsoft Data Platform Vision. It is more than simply a database solution. SQL Server 2008 extends users’ reach so that they can access information from creation to archiving on virtually any device, from the desktop to a mobile device. It also deepens the usability of data, affording users rich analytical and reporting capabilities through applications they are already familiar with like Word 2007 and Excel 2007. SQL Server 2008 delivers a trusted, productive, intelligent data platform to help businesses meet the requirements and seize the opportunities caused by the evolution in business data needs.

## Trusted Database Platform

Trust SQL Server 2008 to run your most mission-critical applications on a highly secure, reliable, and scalable platform. Encrypting information on your database is essential to maintaining the security of your data. Building on the data encryption capabilities introduced in Microsoft® SQL Server® 2005, transparent data encryption in SQL Server 2008 saves time for both database administrators and developers. In comparison to SQL Server 2000, SQL Server 2008 offers much more robust encryption and requires far fewer administrative resources. Moreover, SQL Server 2008 audits all access to private database information, further securing data and simplifying compliance.

*“With SQL Server 2008 we have transparent encryption, so we can easily enforce the encryption of the information in the database itself without making any changes on the application side”.*

*Ayad Shammout, CareGroup HealthCare System*

Database availability and data continuity have been improved in SQL Server 2005 and SQL Server 2008. SQL Server 2005 added support for database mirroring and made 99.999 percent uptime possible. SQL Server 2008 has built on this foundation by adding enhancements to clustering and support for virtualization. In addition, SQL Server 2008 also supports hot-add processor support (on server hardware that supports it) and automatic data corruption recovery on mirrored databases.

Resource Governor is a new utility in SQL Server 2008 that allows complete control over database load by allowing prioritization of all database activity. Database administrators identify and set priorities for workloads and groups then allocate shared CPU and memory resources as they are requested, based on specified limits. This allows organizations to fine-tune and scale the database with much more control and stability than what was available in the past.

## Increased Productivity for Database Workloads

SQL Server 2008 offers many performance enhancements throughout the technology stack to reduce the cost of managing your data infrastructure while streamlining development of data applications. These include improvements within the core Database Engine, SQL Server Analysis Services (SSAS), SQL Server Reporting Services (SSRS), and SQL Server Integration Services (SSIS).

SQL Server 2005 introduced support for access permission management. Database administrators can manage their enterprise data infrastructure easily with SQL Server 2008 Policy-Based Management. This reduces the time they spend on daily maintenance operations by defining a common set of policies for most database operations like query optimizations, naming conventions, backup and restore operations, and index management. Administrators can also apply policies to many servers for consistent management across the enterprise. SQL Server 2008 also enables Performance System Analysis by collecting performance data from your system and storing it in a centralized data repository. Administrators can view reports to benchmark and analyze system performance and health.

With SQL Server 2008, database administrators can also use Management Studio to troubleshoot, tune, and monitor instances across the enterprise; define configuration policies for the enterprise; and use built-in facets and policies to manage surface area configuration and apply best practices—all of which reduces the time and costs of managing enterprise information infrastructure.

SQL Server 2008 has new, built-in compression for database files and transaction log files, row-level and page-level compression, and compression at the backup level. These new compression types free up space for live data. Not only do the SQL Server 2008 data compression features reduce hardware, space, and cooling costs, they can also reduce the operating footprint and enhance processing speeds as a result of smaller amounts of data being retrieved and saved to the database.

*“We’ve seen tables reduced in size by 80 percent using SQL Server 2008 Backup Compression. With 100 terabytes of data, we’re very happy to be able to reduce our backup footprint”.*

*Thomas Grohser Senior Database Engineer, bwin*

As you extend your applications, you get additional development enhancements such as a more sophisticated set of data types that will streamline the development process. Integration with Microsoft® Visual Studio® 2008 and the Microsoft® .NET Framework can speed up development and dramatically reduce the lines of code for new applications. More specifically, programmers are able to access data by defining business entities instead of tables and columns with the ADO.NET Entity Framework. Additionally, they are able to query and retrieve these entities natively within any .NET language with Language Integrated Query (LINQ). These features enable developers to work with the logical entity model while administrators are still able to define the physical implementation of the model as tables and columns. Plus, by upgrading from SQL Server 2000, developers can access types and sources of data that were previously impossible to store, providing additional analysis opportunities to users. The new FILESTREAM data type has been introduced to facilitate easier manipulation of unstructured data such as documents and images residing outside the database. XML data is stored efficiently and is readily accessible with XQuery. Additionally SQL Server 2008 supports geometry and geography data types for storing spatial data. These types support methods and properties that allow for the creation, comparison, analysis, and retrieval of spatial data.

Lastly, SQL Server 2008 now supports multiple options for virtual server consolidation, providing organizations with the flexibility to choose the consolidation approach that best meets their requirements. Capabilities such as centralized management, auditing, and monitoring make it easy to manage multiple databases and data services on virtual appliances, significantly reducing administrative overhead in large enterprises.

## Intelligent Data Platform

SQL Server 2008 drives business intelligence throughout your organization, manages reports and analysis of any size or complexity, and empowers users by providing powerful visualization and integration with the Microsoft® Office system.

SQL Server 2008 enables organizations to import, store, and deliver almost any data as well as manage reports and analyze huge amounts of data. Administrators are able to scale and manage large numbers of users and data with improved query performance on large tables. For example, Unisys and Microsoft recently set a new ETL performance record by loading one terabyte of data in less than 30 minutes using SQL Server 2008 Integration Services (<http://www.microsoft.com/sqlserver/2008/en/us/benchmarks.aspx>).

Obtaining information *from* the database is improved over SQL Server 2000 as well. Business users can create complex reports and share them internally and externally with colleagues, customers, and partners. SQL Server Analysis Services provides a consistent set of key performance indicators and business metrics to all users with its comprehensive and scalable analysis platform. SQL Server 2008 Analysis Services provides the capability to perform complex data analysis and reporting on real-time data: Users can receive immediate feedback about key metrics by creating dashboards and scorecards and tracking key performance indicators that provide current information about business operations.

In SQL Server 2008, these analysis and additional reporting capabilities have been integrated with familiar Microsoft Office applications like Word 2007 and Excel 2007, as well as Microsoft® Office SharePoint® Server 2007. In contrast to SQL Server 2000, SQL Server 2008 Reporting Services enables business users to create their own ad hoc reports when they want them. The IT department no longer needs to be involved when new reports are needed, freeing them to work on other high-priority tasks. In addition to self-service reporting, the new report designer application in SQL Server 2008 allows users to create enterprise-class reports without the need for Developer Studio.

Integrated Data Mining enables predictive analysis so you can investigate common issues like forecasting and identifying key influencers for decisions. Enhanced designers assist in developing scalable analysis models that incorporate best practices into the design experience. The scale and performance of analysis models have been increased by optimizations made to analytical capabilities as well as optimizations to complex computations and aggregations.

SQL Server 2008 introduces sparse columns, which allow NULL values to be stored without taking up any physical space on the disk. SQL Server 2008 also includes a new mechanism called Change Data Capture that captures updated, deleted, and inserted data in an easily consumed storage schema and allows for incremental loading of data warehouses from those tables.

## Feature Changes in SQL Server 2008

SQL Server 2008 contains improvements and additional features in nearly every area of the product. In fact, any one of these enhanced features can be a compelling case for upgrading, depending on the need for high availability, performance, and added functionality. Additionally, upgrading to the latest release of the product extends the Microsoft support life cycle to the maximum degree possible, in accordance with the software support policy.

To better understand the SQL Server 2008 features that make upgrading advantageous, see the white paper “SQL Server 2008 Product Overview” (<http://www.microsoft.com/sql/techinfo/whitepapers/sql2008Overview.mspx>).

## Support

With the latest version of SQL Server, companies can benefit from a long-term and current support path. As of April 2008, SQL Server 2000 has moved off mainstream support to an extended support path. As a result, Microsoft will not accept requests for warranty support, design changes, or new features during the extended support phase.

Microsoft provides upgrade tools to help manage upgrading from prior versions. Compatibility has been maintained with the majority of functionality, which should enable most applications to upgrade seamlessly.

# Key Considerations in Upgrading from SQL Server 2000 to SQL Server 2008

Upgrading to SQL Server 2008 from SQL Server 2000 presents organizations that are undertaking this change with a number of things to consider. Organizations must decide what route to take in upgrading their SQL Server 2000 instances as well as evaluate how the upgrade will affect the functionality of their database applications and other SQL Server services on which their database workloads rely.

## Upgrade Options

Table 1 summarizes four methods for managing an upgrade to SQL Server 2008 from SQL Server 2000. An in-place upgrade modifies the existing database, with the end result that the new version replaces the previous version. A side-by-side upgrade creates a new instance of the database, after which all of the data and many other database objects must be manually copied to the new version before integration with the of the database application. In general, the automatic upgrade path is administratively less intensive but less flexible, while a side-by-side upgrade offers more control with a higher administrative cost.

A full list of upgrade paths may be accessed at MSDN®, the Microsoft® Developer Network (<http://msdn.microsoft.com/en-us/library/ms143393.aspx>).

Table 1: Upgrade Methods and Considerations

|  |  |  |
| --- | --- | --- |
|  | **SQL Server 2000 to  SQL Server 2008  (direct upgrade)** | **SQL Server 2000 to  SQL Server 2008 (with SQL Server 2005 as interim step)** |
| **In-Place Upgrade** | **Advantages**   * Single occurrence of database downtime * Entire operation automated   **Disadvantages**   * Existing applications may require significant modification to retain functionality * Less control over upgrade process | **Advantages**   * Automated * Software release schedules can be coordinated with database upgrades * Application changes can be addressed incrementally   **Disadvantages**   * Administratively more intensive * Rollback strategy potentially more complex |
| **Side-by-Side Upgrade** | **Advantages**   * More granular control over the upgrade process * Original database left unchanged; allows for testing of new database * Single occurrence of database downtime   **Disadvantages**   * Administratively more intensive * Requires more hardware resources | **Advantages**   * More granular control over the upgrade process * Software release schedules can be coordinated with database upgrades * Original database left unchanged; allows for testing of new database * Allows flexible integration of the updated database * Potential for a single occurrence of database downtime   **Disadvantages**   * Administratively more intensive * Requires more hardware resources |

## Side-by-Side Upgrade vs. In-Place Upgrade

There are two fundamental strategies for upgrading:

* **In-place upgrade:** uses the SQL Server 2008 Setup program to directly upgrade a SQL Server 2000 instance to SQL Server 2008. The older SQL Server instance is replaced.
* **Side-by-side upgrade:** performs operations to move all or data and other database components from SQL Server 2000 to a separate SQL Server 2008 instance.

### In-Place Upgrade

Using an in-place upgrade strategy, the SQL Server 2008 Setup program directly replaces a SQL Server 2000 instance with a new SQL Server 2008 instance on the same x86 or x64 platform; the upgraded instance of SQL Server 2000 is replaced by the new SQL Server 2008 instance. There is no need to copy database-related data from the older instance to SQL Server 2008 because the old data files are automatically converted to the new format. When the process is complete, the old SQL Server 2000 instance is removed from the server, with only retained backups able to restore it to its previous state.

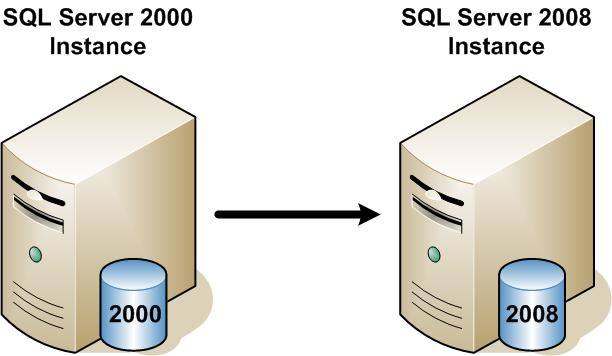


Figure 1: Direct upgrade of SQL Server 2000 to SQL Server 2008

### Side-by-Side Upgrade

Conversely, in a *side-by-side* upgrade, database structure and component data are transferred from the SQL Server 2000 instance to a new, distinct SQL Server 2008 instance; the new SQL Server 2008 instance runs alongside the legacy SQL Server 2000 either by using two servers or a single server.

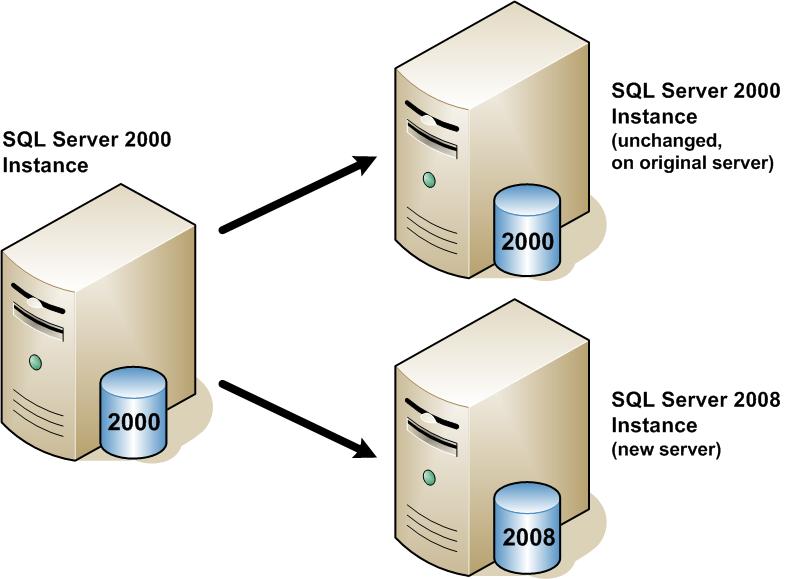


Figure 2: Side-by-side upgrade to SQL Server 2008 using two servers

You may also use the side-by-side method to upgrade to SQL Server 2008 on a single server. Figure 3 shows a side-by-side upgrade on a single server.

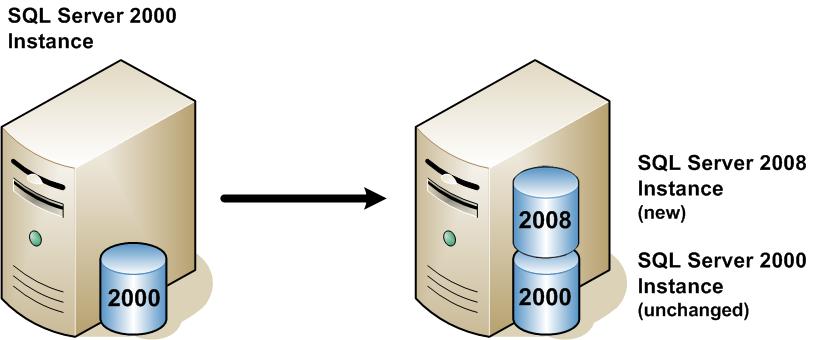


Figure 3: A side-by-side upgrade on a single server, leaving both instances running

Regardless of whether a side-by-side upgrade is performed using one or two servers, data and other database objects must be transferred using other utilities.

Objects requiring other transfer methods include:

* Data files
* Database objects
* SSAS cubes
* Configuration settings
* Security settings
* SQL Server Agent jobs
* SSIS packages

A side-by-side upgrade to a new server offers the most flexibility and control: organizations can take advantage of a new and potentially more powerful server and platform, but the legacy server remains as a fallback if they encounter compatibility issues. This method allows for rigorous testing of the new database before transitioning it into the production environment. The downside of a side-by-side upgrade is that increased manual interventions are required, so it might take more up-front preparation and planning, but, in most cases, the benefits of this degree of control merits the extra effort.

## Comparing In-Place and Side-by-side Upgrade Methods

Table 2 summarizes the distinction between the two upgrade strategies:

Table 2: Characteristics of an In-Place Upgrade vs. a Side-by-Side Upgrade

|  |  |  |
| --- | --- | --- |
| Process | In-Place Upgrade | Side-by-Side Upgrade |
| Number of resulting instances | One only | Two |
| Number of physical servers involved | One | One or more |
| Data file transfer | Automatic | Manual |
| SQL Server instance configuration | Automatic | Manual |
| Supporting tool | SQL Server Setup | Various data transfer methods |

Note that the main distinction between an in-place upgrade and a side-by-side upgrade hinges on the resulting instances. An in-place upgrade replaces the old instance, so that only one instance remains.

Another way to look at the distinctions between an in-place upgrade and a side-by-side upgrade is to focus on how much of the legacy instance you want to upgrade. Table 3 shows how you can use the component level of the upgrade, combined with the resulting number of instances, to determine what upgrade strategies are available for your needs.

Table 3: Upgrade Strategies and Components

|  |  |  |
| --- | --- | --- |
| Component Level | Single Resulting SQL Server 2008 Instance | Two Resulting Instances |
| All components | In-place | Side-by-side |
| Single component | In-place | Side-by-side |
| Single database | Not available | Side-by-side |

### Upgrade Strategy Overview

Expediency, disk space, new server hardware, and high availability are all factors that will determine which upgrade strategy to use. Because of database complexity and the diversity of implementation methods, there are no simple rules to follow.

#### Rolling Back an Upgrade

When evaluating which upgrade strategy to use, take into account the risk that an in-place upgrade or side-by-side upgrade may need to be rolled back. The complexity and effort required to roll back is an important factor in choosing which method to use.

Rolling back an in-place upgrade can be complex and time-consuming. The new data file structures for SQL Server 2008 are not compatible with SQL Server 2000. To roll back an upgraded instance, you must uninstall the SQL Server 2008 instance, remove the data files and other components, reinstall the legacy SQL Server 2000 or SQL Server 2005 instance, and restore the original data. Having a backup or image of the initial system may enable you to shorten the time required to restore the original system on the server. One option is to copy the legacy data files from a backup location to the appropriate disk volume, and then integrate the SQL Server 2000 database in the previous environment.

In a side-by-side upgrade, the new SQL Server 2008 instance resides alongside the legacy SQL Server instance, either on the same server or on a different server. As a result, the legacy instance remains available for a rollback scenario.

However, after the upgraded SQL Server 2008 instance goes into production and starts capturing new data, there will come a point in time when enough new data has been captured that a rollback is no longer realistic. For an in-place upgrade, if you encounter problems after the system is in production, making adjustments or “patches” to the new application would be a better option than attempting a rollback. For a side-by-side upgrade, you could employ SSIS to transfer new data from the SQL Server 2008 instance to the legacy SQL Server 2000 to bring it current. Depending on the complexity of the data, this could be a difficult process.

#### Choosing an Upgrade Strategy

The upgrade method available for your specific needs depends on numerous factors, including the components you want to upgrade and the editions you want to use.

* **Components:** A certain upgrade strategy might not be possible because the component does not support it. For example, there is no in-place upgrade for SSIS from SQL Server 2000; Microsoft recommends that you upgrade most SQL Server 2000 SSAS components.
* **Versions and Editions:** The in-place upgrade strategy does not support all paths between versions and editions. For example, to upgrade a SQL Server 2000 Enterprise Edition instance to SQL Server 2008 Standard Edition, you must perform a side-by-side upgrade because SQL Server Setup does not support an in-place upgrade path.
* **Partial upgrading:** To transition only a few databases on a server to SQL Server 2008 and leave the rest on the legacy version, you must use a side-by-side upgrade.
* **Upgrading over time:** To transition databases gradually, a few databases at a time, from a legacy instance to SQL Server 2008, you can only use a side-by-side upgrade.
* **Effect on applications:** If your organization requires minimal disturbance to the existing applications and users, you may want to choose an in-place upgrade if possible.
* **Availability:** Both an in-place upgrade and a side-by-side upgrade require that the databases be unavailable for a certain amount of time. The amount of downtime required depends primarily on the size of the data sets. At first, it might seem that an in-place upgrade would be faster than a side-by-side upgrade because the data is not transferred from one server to another. However, an in-place upgrade also requires time for the installation of SQL Server 2008. In a side-by-side upgrade, SQL Server 2008 is already installed on another instance. If the data transfer proceeds quickly and few changes are needed on the new instance, a side-by-side upgrade might be faster than an in-place upgrade.
* **Rollback:** For many database systems in production, it is impossible to justify a change without a rollback strategy in case the results are not acceptable. The side-by-side upgrade strategy supports rollback at the time of acceptance testing because the legacy instance can still be made available. However, after users update the databases in the new instance, rollback might no longer be feasible.

Some of these factors alone may dictate one strategy over another. Regardless of which method is employed, a successful upgrade to SQL Server 2008 should be smooth and trouble free. To achieve that smooth transition, you must devote sufficient planning for the upgrade and match the complexity of your database application. Otherwise, you risk costly and stressful errors and upgrade problems.

Part of the upgrade planning process should also include consideration of how new SQL Server 2008 components can be used to enhance your particular implementation. Significant improvements have been made in the areas of relational database operations, specifically higher availability, database engine enhancements, and security and auditing. Analysis Services, Data Mining, Integration Services, and Reporting Services have all been improved. Close analysis of enhancements made in these areas in advance of the upgrade itself will help ensure that organizations derive the greatest benefit from the new version.

Like all IT projects, planning for every contingency and then testing your plan gives you confidence that you will succeed. However, if you ignore the planning process, you increase the chances of running into difficulties that can derail and delay your upgrade. The flowchart below is an example of a typical upgrade process plan. Creating a similar plan tailored to your particular needs is highly recommended.

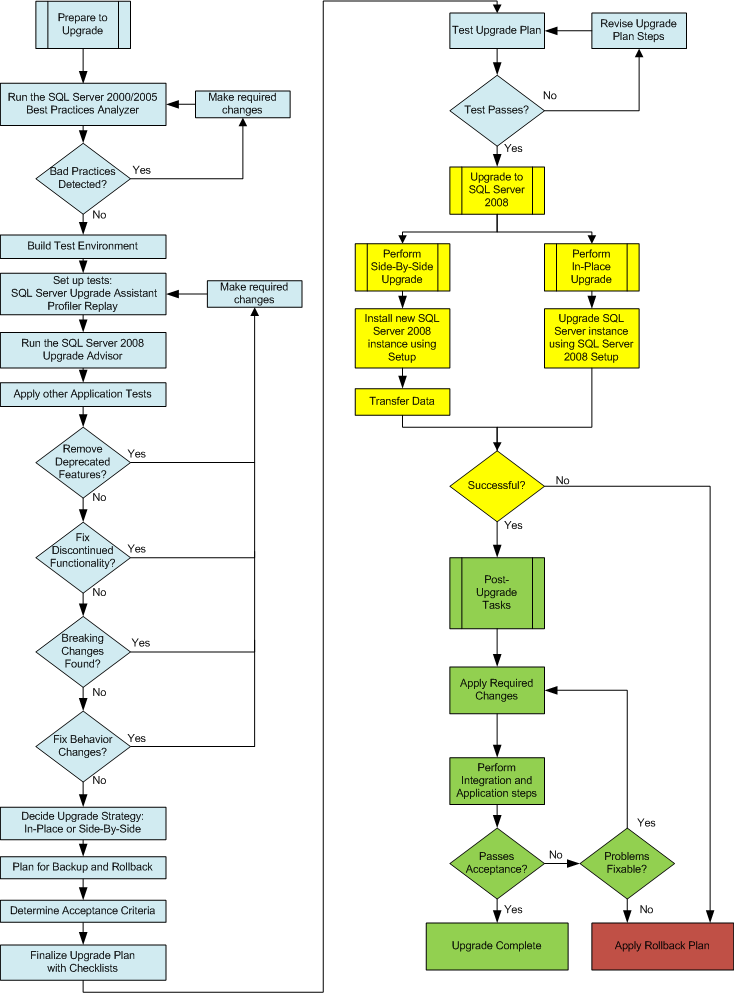


Figure 4 An example upgrade process plan

For more in depth coverage of this topic, you may download the SQL Server 2008 Upgrade Technical Reference Guide (<http://www.microsoft.com/downloads/details.aspx?FamilyID=66d3e6f5-6902-4fdd-af75-9975aea5bea7&displaylang=en>).

## Functionality Considerations

Regardless of your upgrade strategy, you should consider the functionality changes between SQL Server 2000 and SQL Server 2008—some SQL Server 2000 features on which your applications rely may be discontinued in SQL Server 2008 or in future versions of SQL Server.

Application compatibility testing is a good practice to identify and resolve potential issues that may arise after upgrading the database. A typical set of Application Compatibility Testing tasks should include the following:

* Determine how much Application Compatibility Testing is appropriate.
* Ensure testing of all relational database management system (RDBMS) queries.
* Test databases with the current compatibility level. In the case of SQL Server 2008, it is level 10.
* Test application setup with SQL Server 2008.
* Confirm that any third-party vendor software is certified on SQL Server 2008.

### Minimizing Downtime

Often when working with complex systems, or applications considered to be mission critical, it is desirable to minimize the amount of time these systems are unavailable while performing upgrade tasks. There are several tasks that can be carried out in advance of the upgrade itself, resulting in a shorter maintenance event. The following tasks can be performed without any negative impact or outage to the databases that will be upgraded:

* Upgrade any databases older than SQL Server 2000 to an upgradable version (at least SQL Server 2008).
* Ensure installation requirements are met. Run the SQL Server Upgrade Advisor for a listing of potential issues that will prevent the setup routine from completing. Resolve each of the issues enumerated in the report.
* Preinstall the Microsoft® .NET Framework 3.5 Service Pack 1 (SP1).
* Preinstall SQL Server 2008 common components (Simple Network Access Controller, Management tools).
* Select efficient media and transfer methods (Fibre Channel, Cat-5/6, network-attached storage (NAS), storage area network (SAN), removable media) if performing a side-by-side upgrade.
* Use DBCC CHECKDB to examine the continuity of data as well as check for other anomalies.
* Back up data before, during, and after the upgrade procedure.

### Application Backward Compatibility

When planning for an upgrade to SQL Server 2008, you need to understand which features have been deprecated, discontinued, or changed in the new version. Being aware of these changes ahead of an upgrade can help prevent both performance problems and compatibility issues.

The majority of SQL Server 2008 functionality and behavior is backward compatible with SQL Server 2000. However, you should examine feature changes during the planning process. The most serious backward-compatibility issues that can affect planning are those that can block an in-place upgrade and prevent an installation of SQL Server 2008.

If the SQL Server 2008 Setup program detects these issues in the process of an in-place upgrade, it will abort the install, leaving the legacy instance unchanged. The SQL Server 2008 Upgrade Advisor is the best tool for finding these types of blocking issues ahead of time. You can find comprehensive information about changes in the SQL Server 2008 Books Online (BOL) topic, “SQL Server Backward Compatibility” at <http://msdn.microsoft.com/en-us/library/cc707787.aspx>.

### Discontinued Features

While the vast majority of backwards compatibility has been retained, it is possible that certain components of SQL Server 2000 have been discontinued. These features functioned in earlier versions of SQL Server but have been removed from SQL Server 2008. Although some references to these features might not block an in-place upgrade, you should remove those references anyway—if the references are not removed, the application might not behave correctly.

Use the [Upgrade Advisor](#_SQL_Server_2008) to detect whether your application is using discontinued features. For more information about such features, see “Discontinued SQL Server Features in SQL Server 2008” at <http://msdn.microsoft.com/en-us/library/cc707782.aspx>. See Appendix A: Discontinued Features in SQL Server 2008 for a list of features that have been discontinued.

### Deprecated Features

Features that are deprecated in SQL Server 2008 still operate the same as in the legacy versions, but they will be removed in a future version of SQL Server. Access to these features does not necessarily need to be removed to complete an upgrade, but you should eventually address them because they might cause problems with upgrades after SQL Server 2008. For details, see the SQL Server 2008 BOL topic: “Deprecated SQL Server Features in SQL Server 2008” at <http://msdn.microsoft.com/en-us/library/cc707789.aspx>.

**Note:** Your upgrade will not be blocked if you use deprecated features. However, it is advised that you decide how or when you want to deal with any of these to give yourself plenty of time to resolve the issues before they are discontinued in some future SQL Server release.

See Appendix B: Deprecated Features in SQL Server 2008 for a list of features that will not be supported in the next version of SQL Server or future versions of SLQ Server.

### Breaking Changes

Breaking changes to SQL Server 2008 are those that might require changes to the applications because the features in question now have a different behavior. If you do not use the feature, there is no impact on you, but if you do use the feature, your application might be affected.

The best tool for discovering this type of issue is Upgrade Advisor, which analyzes a legacy system and reports on all potential breaking changes and how to resolve them. For more information about this type of change, see “Breaking Changes to SQL Server Features in SQL Server 2008” at <http://msdn.microsoft.com/en-us/library/cc707784.aspx>.

### Behavior Changes

Behavior changes may not visibly affect your database code or applications, but you need to be aware of them. Database operations that depend on features with modified behaviors may be adversely affected. For more information about behavior changes, see “Behavior Changes to SQL Server Features in SQL Server 2008” at <http://msdn.microsoft.com/en-us/library/cc707785.aspx>.

### Other SQL Server Services Affected by Upgrade

For information on backward compatibility for SQL Server 2008 components, see “Backward Compatibility” at <http://msdn.microsoft.com/en-us/library/cc280407.aspx>.

# SQL Server Upgrade Tools

Microsoft and Microsoft partners offer myriad tools to help automate and better ensure the success of the upgrade process to SQL Server 2008. Each tool has its own purpose and timing, so it is best to become familiar with all the tools and then use those most appropriate for each phase of your upgrade.

## Primary Tools

The principal tools for planning and executing your SQL Server 2008 upgrade are the SQL Server 2008 Upgrade Advisor and DTS xChange from Pragmatic Works.

### SQL Server 2008 Upgrade Advisor

Perhaps the most important tool of the several tools commonly used for upgrade planning is SQL Server 2008 Upgrade Advisor. Upgrade Advisor can help ease the transition to SQL Server 2008 by detecting potential incompatibility issues in your legacy SQL Server 2000 instance. It analyzes objects and code within legacy instances and produces reports detailing upgrade issues. The resulting reports show detected issues and provide guidance about how to resolve the issues or work around them. The reports are stored on disk, and you can review them by using Upgrade Advisor or export them to Microsoft® Office Excel® for further analysis.

In addition to analyzing data and database objects, SQL Server 2008 Upgrade Advisor can analyze Transact-SQL (T-SQL) scripts and SQL Server Profiler/SQL Trace traces. Upgrade Advisor examines SQL code for syntax that is no longer valid in SQL Server 2008. It generates a report listing the code in question along with links to where you can find more information to help resolve the questionable code.

Whether you choose an in-place upgrade or a side-by-side upgrade, run Upgrade Advisor on your legacy systems. You can run Upgrade Advisor from a local or remote server, and you can execute it from the Command Prompt window by using a configuration filename as an input parameter.

#### Requirements

The Upgrade Advisor requires the following to run:

* Windows Server® 2008, Windows Server® 2003 Service Pack 2 (SP2), Windows Vista® SP1, or Windows® XP SP3
* The Microsoft® .NET Framework 2.0 (the same version of the .NET Framework included with SQL Server 2008 and Microsoft® Visual Studio® 2005)
* Windows® Installer 4.5
* SQL Server 2000 Decision Support Objects (DSO) if analyzing SSAS (you can use SQL Server 2000 Setup to install DSO)
* SQL Server 2000 client components if analyzing Data Transformation Services (DTS) (you can use SQL Server 2000 Setup to install the SQL Server 2000 client components)
* Pentium III-compatible processor or higher, with a processor speed of at least 500 megahertz (MHz)
* 15 megabytes (MB) of available hard disk space

#### Availability

Upgrade Advisor is a separate download. The most recent downloadable version is available as part of the Microsoft SQL Server 2008 Feature Pack available at <http://www.microsoft.com/downloads/details.aspx?FamilyId=C6C3E9EF-BA29-4A43-8D69-A2BED18FE73C&displaylang=en>.

You can find more information about this valuable tool in the [Upgrade Advisor Guide in SQL Server 2008 BOL](ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10sq_GetStart/html/d85b0833-ddeb-42e3-9397-97ea60d521b7.htm); also see “Using Upgrade Advisor to Prepare for Upgrades” at <http://msdn.microsoft.com/en-us/library/ms144256.aspx>.

### DTS xChange

Although SQL Server 2008 Integration Services may be considered the fourth iteration of an ETL tool in SQL Server, it is dramatically different from SQL Server 2000 Data Transformation Services (DTS). The architecture of SSIS has changed drastically to support an in-memory ETL method that can efficiently support a load of millions of rows. To take advantage of this new architecture, organizations must migrate their DTS packages into SSIS and apply new best practices to the migrated packages.

SQL Server 2008 comes with the DTS Package Migration Wizard, a built-in means of migrating simple DTS packages to SSIS. However, this method will not meet the needs of all organizations. For organizations in need of a more robust solution that can handle thousands of packages with little administrator intervention, DTS xChange is an enterprise solution, offered by a Microsoft partner Pragmatic Works, that migrates DTS packages to Integration Services while applying a series of best practices rules to the packages.

DTS xChange is broken into three components:

1. **Profile**: DTS xChange Profiler helps organizations estimate their migration project in hours and dollar cost whether they choose to use an automation tool or not.
2. **Convert**: DTS xChange migrates packages, applying rules to each DTS package as it migrates them to enforce best practices.
3. **Monitor**: The SSIS Performance Warehouse is a software development kit (SDK) to help organizations get the most out of their new Integration Services environment. It contains a series of reports and a data warehouse to monitor administrators’ Integration Services package execution.

## Secondary Tools

There are multiple additional tools that fit specialized needs in the upgrade planning and execution process, including:

* Microsoft® Assessment and Planning Toolkit 3.2
* SQL Server 2008 Upgrade Assistant
* SQL Server Best Practices Analyzer
* System Configuration Checker
* SQL Server Profiler
* SQL Server: Deprecated Features Object Counter
* Other tools

### Microsoft Assessment and Planning Toolkit 3.2

For enterprise users of SQL Server, the number and versions of all SQL databases may not be readily available. For these occurrences, the Microsoft Assessment and Planning Toolkit 3.2 (MAP) can be used to ascertain details about hardware and software running SQL Server databases. MAP is a scalable and agent-less assessment platform designed to make it easier for our customers to adopt the latest Microsoft technologies. In this version, MAP has expanded its assessment capabilities to include SQL Server 2008, Microsoft® Forefront™/ Network Access Protection (NAP), and Microsoft® Online Services migration.

#### Requirements

Supported operating systems: Windows Server® 2003, Windows Server® 2008, Windows Vista, Windows Vista Service Pack 1, Windows XP Professional Edition

Hardware Requirements:

1.6-gigahertz (GHz) or faster processor minimum (dual-core for Windows Vista)

1.5 GB of RAM minimum (2.0 GB for Windows Vista)

1 GB of available hard-disk space required

10/100 megabits per second (Mbps) network adapter required

Software Requirements: SQL Server 2005 or Microsoft® SQL Server® 2008 Express for storing inventory and assessment data. Microsoft® Office Word 2003 SP2 or Word 2007 and Microsoft® Office Excel® 2003 SP2 or Excel 2007 for generating reports. The .NET Framework v3.5 SP1, Windows Installer v4.5

#### Availability

For more information and download instructions, see the Microsoft Assessment and Planning Toolkit on the Microsoft Web site: <http://www.microsoft.com/downloads/details.aspx?FamilyID=67240b76-3148-4e49-943d-4d9ea7f77730&DisplayLang=en>.

### SQL Server 2008 Upgrade Assistant

The SQL Server 2008 Upgrade Assistant is an external tool that lets you determine in a test environment how an application currently running on SQL Server 2000 will run on SQL Server 2008. This tool uses Upgrade Advisor, along with baseline and trace replay in a test environment, to help identify compatibility issues.

#### Requirements

Requirements for using Upgrade Assistant are:

* Windows Server 2008, Windows Server 2003 R2, Windows Vista, or Windows XP SP2 or later
* SQL Server 2000 SP4 or later
* Microsoft .NET Framework 2.0 SP1 or later

#### Availability

For more information and download instructions, see SQL Server 2008 Upgrade Assistant on the Scalability Experts Web site: <http://www.scalabilityexperts.com/default.asp?action=article&ID=43>.

### Best Practices Analyzer for SQL Server 2000

Before installing SQL Server 2008, you should also run the SQL Server Best Practices Analyzer (BPA) against the SQL Server 2000 instance. If bad or questionable practices exist, you may address them before the upgrade, moving the fixes through test and into production. Using best practices on the legacy SQL Server systems first will help ensure a smoother upgrade.

You can download the SQL Server 2000 version of BPA at Best Practices Analyzer Tool for Microsoft SQL Server 2000 1.0: <http://www.microsoft.com/downloads/details.aspx?FamilyID=b352eb1f-d3ca-44ee-893e-9e07339c1f22&DisplayLang=en>.

### System Configuration Checker (SQL Server 2008 Setup During In-Place Upgrade)

An in-place upgrade uses SQL Server 2008 Setup to directly upgrade a SQL Server 2000 instance. SQL Server 2008 installs required prerequisites such as the .NET Framework and Windows PowerShell™ 1.0. It also scans the target computer for minimum hardware and software requirements, as well as a compatible SQL Server edition.

To do this, the SQL Server 2008 Setup program contains a utility called the System Configuration Checker (SCC), which performs a scan of the computer in preparation for an install. For comprehensive information about SCC, see the SQL Server 2008 BOL topic “Check Parameters for the System Configuration Checker” at <http://msdn.microsoft.com/en-us/library/ms143753.aspx>.

The Setup SCC looks for conditions that can prevent a successful SQL Server installation or upgrade. These checks occur before Setup starts the SQL Server 2008 Installation Wizard and report any issues that would block an install along with advice about how to address the blocking issues. The Setup SCC uses rules from the following categories (for more information about any of these categories, see the related link from SQL Server 2008 BOL):

* Feature Installation Rules: <http://msdn.microsoft.com/en-us/library/cc646015.aspx>
* Upgrade and Repair Rules Check: <http://msdn.microsoft.com/en-us/library/cc281843.aspx>
* Edition Upgrade Rules: <http://msdn.microsoft.com/en-us/library/cc645998.aspx>
* Uninstallation Rules: <http://msdn.microsoft.com/en-us/library/cc645979.aspx>

The SCC will prevent an upgrade if the following conditions are not met:

* The target computer must be connected to the Internet while the .NET Framework security check validates a certificate.
* The SQL Server registry keys must be consistent.
* The CPU architecture of the installation program must match the CPU architecture of features intended for upgrading.
* If the computer is clustered, the cluster service must be online.
* Windows PowerShell must be installed. (Setup will do this automatically when installing prerequisites.)
* SQL Server Setup must be supported on this operating system platform.
* SCC checks whether a pending computer restart is required.
* The existing performance counter registry hive must be consistent.
* SCC checks that neither Microsoft® SQL Server® 7.0 nor SQL Server 7.0 OLAP Services is installed on the server. SQL Server 2008 is not supported running on the same server with SQL Server 7.0.

Additional checks that the SCC performs are:

* Database features that are not supported in the SQL Server version and edition to which you are upgrading.
* Restrictions on restarting of the SQL Server Service.
* SQL Server service is not set to **Disabled**.
* Analysis of whether or not the SQL Server version meets the upgrade matrix requirements.
* SASS upgrade validity.
* Evaluation of whether the edition of the selected instance of SQL Server is supported in a given scenario.

### SQL Server Profiler

SQL Server Profiler can record a running workload and then replay that same activity from a given SQL Server instance, making it a valuable tool for preparing an upgrade.

Profiler is useful for simulating an upgrade to determine performance and correct behavior. For example, you can use SQL Server 2008 Profiler to trace a SQL Server 2000 database under load and save the trace. You can then restore the SQL Server 2000 database to two instances on equivalent hardware: a SQL Server 2000 instance and a SQL Server 2008 instance. Run the replay on each (but at different times if on the same server), and while running the replay, also run a Profiler trace on each of the two runs, capturing errors and query durations. By comparing the results, you can determine whether the upgrade behaves correctly (without error) and performs well.

Using Profiler to test upgrade results is made much easier by using the SQL Server 2008 Upgrade Assistant. Upgrade Assistant helps automate the process and reports for comparing performance and behavior of an upgraded SQL Server.

For more information about using Profiler for replay, see the SQL Server 2008 BOL topic “Replaying Traces” at <http://technet.microsoft.com/en-us/library/ms190995.aspx>.

### SQL Server: Deprecated Features Object Counter in System Monitor

SQL Server 2008 provides a new System Monitor (Perfmon) counter called SQL Server: Deprecated Features Object to monitor whether your application is submitting commands to the SQL Server 2008 database engine that have been scheduled for removal from SQL Server in future releases. You should remove such deprecated commands from SQL Server 2008 applications after they are detected. You can use this counter to help plan modifications to your application code so that when you upgrade to the next version of SQL Server after SQL Server 2008, the upgrade process will go more smoothly.

Choose which type of feature to monitor by using the Instance selection box for the counter. System Monitor records the total number of times the deprecated feature was encountered since SQL Server 2008 was last started. For details about using this tool, see the SQL Server 2008 BOL topic, “SQL Server, Deprecated Features Object” at <http://technet.microsoft.com/en-us/library/bb510662.aspx>.

### SQL Server Data Backup

It will be important to assess the point at which it is sensible to perform backup operations and to assess which data should be backed up. User databases for example should be backed up after users are not using the system and before the upgrade process has started. Another point which may be considered for backup might be once the upgrade has been complete but before any system parameterization has occurred. One final backup should be made once the entire system is up and running but before users are permitted to start using the application.

### Other Tools

#### Analysis Services Migration Wizard

Use the Migration Wizard to migrate the metadata, and optionally the data, from an existing Microsoft SQL Server 2000 Analysis Services database into a Microsoft SQL Server Analysis Services database. Additionally, the migration process can be saved to a script file for later migration.

#### DTS Package Migration Wizard

Installing SSIS 2008 also installs the DTS Package Migration Wizard, which aids in the migration of DTS packages to SSIS.

Also, SQL Server 2008 provides support for running DTS packages. For details, see “Support for Data Transformation Services (DTS) in SQL Server 2008” at <http://msdn.microsoft.com/en-us/library/bb500440.aspx>.

For information about upgrading DTS to SSIS and support for DTS, see “SQL Server Integration Services” at <http://technet.microsoft.com/en-us/library/ms141026.aspx>.

# Conclusion

Upgrading from SQL Server 2000 to SQL Server 2008 can provide organizations with broad benefits to the scalability, manageability, security, and performance of their database applications. However, there are numerous considerations that companies must bear in mind as they plan and execute their upgrade. Among these are not only the means and the path by which they will affect their upgrade, but also features that have changed or will change in the future that will impact their database.

Microsoft and Microsoft partners like Pragmatic Works provide a broad array of tools to help companies successfully upgrade to SQL Server 2008. Effective planning, testing, and proper use of tools can help make the upgrade a smooth experience that provides new capabilities to your organization.

# Appendix A: Discontinued Features in SQL Server 2008

The following features are not supported in SQL Server 2008; for the most up-to-date list of discontinued features, see <http://msdn.microsoft.com/en-us/library/cc280407.aspx>.

|  |  |
| --- | --- |
| **Database Engine** | **Discontinued feature** |
| Aliases | sp\_addalias |
| APIs | Registered Servers API  SQL-DMO based Windows® Management Instrumentation (WMI) provider  SQL Namespace API (SQL-NS) |
| Backup and restore | Named pipe backup devices  DUMP statement  LOAD statement  BACKUP LOG WITH NO\_LOG  BACKUP LOG WITH TRUNCATE\_ONLY  BACKUP TRANSACTION |
| Command prompt utilities | **isql** utility |
| Compatibility level | **60, 65, and 70 compatibility levels** |
| Configuration options | **'allow updates'** option of **sp\_configure**  **'open objects'** option of **sp\_configure**  **'set working set size'** option of **sp\_configure** |
| Database creation | DISK INIT  DISK RESIZE |
| Database creation | FOR LOAD option of CREATE DATABASE |
| DBCC | DBCC CONCURRENCYVIOLATION  DBCC DBREPAIR  DBCC NEWALLOC  DBCC PINTABLE, DBCC UNPINTABLE  DBCC ROWLOCK  DBCC TEXTALL  DBCC TEXTALLOC |
| Extended store procedure programming | Use of SRV\_PWD field in the SRV\_PFIELD structure when there has been an impersonation context switch from the original login |
| Groups | **sp\_addgroup**  **sp\_changegroup**  **sp\_dropgroup**  **sp\_helpgroup** |
| Network protocols | The following protocols: NWLink IPX/SPX, AppleTalk, Banyan Vines, Multiprotocol |
| Rebuild master | Rebuildm.exe |
| Sample databases | **Northwind** and **pubs** |
| Setup.exe | Remote Setup - the TARGETCOMPUTER parameter - is not supported |
| Tools | Surface Area Configuration Tool |
| Transact-SQL | **\*=** and **=\*** outer join operators |
| Virtual tables | **Syslocks** |
| Web Assistant | **sp\_makewebtask**  **sp\_dropwebtask**  **sp\_runwebtask**  **sp\_enumcodepages** |
| **Analysis Services** | **Discontinued Feature** |
| Connection string properties | Mining Execution Location  Mining Location  Log File  Execution Location  Distinct Measures by Key  Large Level Threshold |
| Features | Aggregated Providers  Linked Cubes  Custom Level Formulas  Cube and Database Role Commands |
| MDX | **CreateVirtualDimension**  **CreatePropertySet**  **Ignore**  **With Cache**  **Create Cache** |
| Other | Active Directory® Registration  Skipped levels in parent-child hierarchies  Notification Services |
| **Reporting Services** | **Discontinued Feature** |
| Rendering | HTML 3.2 Rendering Extension  HTML OWC Rendering Extension  SQL Server 2000 Report Server Web Service Endpoint |
| Report server initialization | Rsactivate.exe  Windows® Internet Explorer® 5.5 Support  Report Builder Runs in Full Trust Mode Only |
| Tools | Surface Area Configuration Tool |
| **Replication** | **Discontinued Feature** |
| All types of replications | * Creating push subscriptions without a connection to the Subscriber in the New Subscription Wizard * Using file transfer protocol (FTP) to initialize Subscribers running SQL Server version 7.0 * Creating subscriptions in Windows® Synchronization Manager * Subscribing to a publication by locating it in Active Directory * Snapshot ActiveX® control * Remote agent activation * Microsoft® Office Access® (Microsoft® Jet 4.0) Subscribers |
| Transactional replication | Message Queuing option for queued updating subscriptions |
| Merge replication | **IVBCustomResolver** interface |
| **Other Tools and Features** | **Discontinued Feature** |
| Setup command-line parameters | ADDLOCAL  DISABLENETWORKPROTOCOLS  DISABLENETWORKPROTOCOLS  INSTALLSQLDATADIR  REINSTALL  REINSTALLMODE  REMOVE  SAMPLEDATABASE  SAVESYSDB  SKUUPGRADE  UPGRADE  USESYSDB |

# Appendix B: Deprecated Features in SQL Server 2008

The following features will not be supported in the next version of SQL Server:

|  |  |
| --- | --- |
| **Database Engine** | **Feature Not Supported in the Next Version of SQL Server** |
| Backup and restore | BACKUP { DATABASE | LOG } WITH PASSWORD  BACKUP { DATABASE | LOG } WITH MEDIAPASSWORD  RESTORE { DATABASE | LOG } WITH DBO\_ONLY  RESTORE { DATABASE | LOG } WITH PASSWORD  RESTORE { DATABASE | LOG } WITH MEDIAPASSWORD |
| Compatibility levels | 80 compatibility level and upgrade from version 80 |
| Database objects | WITH APPEND clause on triggers |
| Database options | **sp\_dboption** |
| Instance options | Default setting of **disallow results from triggers** option = 0 |
| Metadata | DATABASEPROPERTY |
| Query hints | FASTFIRSTROW hint |
| Remote servers | **sp\_addremotelogin**  **sp\_addserver**  **sp\_dropremotelogin**  **sp\_helpremotelogin**  **sp\_remoteoption**  @@remserver  SET REMOTE\_PROC\_TRANSACTIONS |
| Security | **sp\_dropalias** |
| SET options | SET DISABLE\_DEF\_CNST\_CHK  SET ROWCOUNT for INSERT, UPDATE, and DELETE statements |
| System tables | **sys.database\_principal\_aliases** |
| Transact-SQL syntax | Use of \*= and =\*  COMPUTE / COMPUTE BY  RAISERROR syntax |
| Tools | sqlmaint Utility |
| **Analysis Services** | **Feature Not Supported in the Next Version of SQL Server** |
| Connection string properties | Mining Execution Location, Mining Location, Log File, Execution Location, Distinct Measures by Key, Large Level Threshold |
| Features | Aggregated Providers, Linked Cubes, Custom Level Formulas, Cube and Database Role Commands |
| MDX | **CreateVirtualDimension, CreatePropertySet, Ignore** |
| Other | Active Directory Registration, Skipped levels in parent-child hierarchies, Surface Area Configuration Tool |
| **Replication** | **Feature Not Supported in the Next Version of SQL Server** |
| All types of replications | Attachable subscriptions  Adding publications to Active Directory  Checksum validation  Schema changes usingsp\_repladdcolumn andsp\_repldropcolumn  SQL Distributed Management Objects (SQL-DMO)  Subscriber registration  -UseInprocLoader parameter |
| Transactional replication | Distribution ActiveX control  "No sync" subscriptions to transactional publications  ODBC Subscribers  Replicating to Oracle 8 subscribers and from Oracle 8 publishers  Replication Distributor Interface  Subscription expiration for transactional publications  Transformable subscriptions  Updatable subscriptions including immediate updating and queued updating with snapshot and transactional publications |
| Merge replication | Alternate synchronization partners  **@allow\_partition\_realignmentproperty** in **sp\_addmergepublication**  **@delete\_tracking** property in **sp\_addmergearticle**  **-ExchangeType**parameter  Logical Records  Merge ActiveX control  Multicolumn UPDATE option  "No sync" subscriptions to merge publications  **-ParallelUploadDownloadparameter** |
| **Other Tools and Features** | **Feature Not Supported in the Next Version of SQL Server** |
| Setup command-line parameters | ADDLOCAL  DISABLENETWORKPROTOCOLS  INSTALLSQLDATADIR  REINSTALL  REINSTALLMODE  REMOVE  SAMPLEDATABASE  SKUUPGRADE  UPGRADE  USESYSDB |

The following features will not be supported in a future version of SQL Server:

|  |  |
| --- | --- |
| **Database Engine** | **Feature Not Supported in a Future Version of SQL Server** |
| Backup and restore | BACKUP { DATABASE | LOG } TO TAPE  BACKUP { DATABASE | LOG } TO *device\_that\_is\_a\_tape*  **sp\_addumpdevice 'tape'**  **sp\_helpdevice** |
| Collations | Azeri\_Latin\_90  Azeri\_Cyrilllic\_90  Hindi  Korean\_Wansung\_Unicode  Lithuanian\_Classic  Macedonian  SQL\_AltDiction\_CP1253\_CS\_AS |
| Compatibility levels | **sp\_dbcmptlevel**  Database compatibility level 90 |
| Configuration | * SET ANSI\_NULLS OFF and ANSI\_NULLS OFF database option * SET ANSI\_PADDING OFF and ANSI\_PADDING OFF database option * SET CONCAT\_NULL\_YIELDS\_NULL OFF and CONCAT\_NULL\_YIELDS\_NULL OFF database option * SET OFFSETS |
| Data types | **sp\_addtype**  **sp\_droptype**  **timestamp** syntax for **rowversion** data type  Ability to insert null values into **timestamp** columns  **'text in row'** table option  Data types:   * text * ntext * image |
| Database management | **sp\_attach\_db**  **sp\_attach\_single\_file\_db** |
| Database objects | CREATE DEFAULT  DROP DEFAULT  **sp\_bindrule**  **sp\_bindefault**  **sp\_change\_users\_login**  **sp\_depends**  **sp\_renamedb**  **sp\_renamedb**  **sp\_getbindtoken**  **sp\_unbindrule**  **sp\_unbindefault** |
| Database options | Ability to return result sets from triggers  **sp\_bindsession**  **sp\_bindsession**  **sp\_resetstatus**  **sp\_resetstatus**  TORN\_PAGE\_DETECTION option of ALTER DATABASE |
| DBCC | DBCC DBREINDEX  DBCC INDEXDEFRAG  DBCC SHOWCONTIG  DBCC PINTABLE  DBCC UNPINTABLE |
| Extended properties | **Level0type** = 'type' and **Level0type** = 'USER' to add extended properties to level-1 or level-2 type objects |
| Extended stored procedures | **xp\_grantlogin**  **xp\_revokelogin**  **xp\_loginConfig** |
| Extended stored procedure programming | **sp\_addextendedproc**  **sp\_dropextendedproc**  **sp\_helpextendedproc**  **srv\_alloc**  **srv\_convert**  **srv\_describe**  **srv\_getbindtoken**  **srv\_got\_attention**  **srv\_message\_handler**  **srv\_paramdata**  **srv\_paraminfo**  **srv\_paramlen**  **srv\_parammaxlen**  **srv\_paramname**  **srv\_paramnumber**  **srv\_paramset**  **srv\_paramsetoutput**  **srv\_paramstatus**  **srv\_paramtype**  **srv\_pfield**  **srv\_pfieldex**  **srv\_rpcdb**  **srv\_rpcname**  **srv\_rpcnumber**  **srv\_rpcoptions**  **srv\_rpcowner**  **srv\_rpcparams**  **srv\_senddone**  **srv\_sendmsg**  **srv\_sendrow**  **srv\_setcoldata**  **srv\_setcollen**  **srv\_setutype**  **srv\_willconvert**  **srv\_wsendmsg** |
| Functions | **fn\_get\_sql** |
| Index options | CREATE TABLE, ALTER TABLE, or CREATE INDEX syntax without parentheses around the options  **sp\_indexoption** |
| Instance options | **sp\_configure** option **'allow updates'**  **sp\_configure** options:   * **'locks'** * **'open objects'** * **'set working set size'**   **sp\_configure** options **'user instances enabled'**  **sp\_configure** option **'priority boost'**  **sp\_configure** option **'remote proc trans'** |
| Linked servers | A linked server uses OLEDB |
| Locking | **sp\_lock** |
| Mail | SQL Mail |
| Metadata | FILE\_ID  INDEXKEY\_PROPERTY |
| Native XML Web Services | The CREATE ENDPOINT or ALTER ENDPOINT statement with the FOR SOAP option  EXT\_endpoint\_webmethods  EXT\_soap\_endpoints |
| Programmability | SQL Server Database Management Objects (SQL-DMO) |
| Removable databases | **sp\_certify\_removable**  **sp\_create\_removable**  **sp\_dbremove** |
| Security | The ALTER LOGIN WITH SET CREDENTIAL syntax  **sp\_addapprole**  **sp\_dropapprole**  **sp\_addlogin**  **sp\_droplogin**  **sp\_adduser**  **sp\_dropuser**  **sp\_grantdbaccess**  **sp\_revokedbaccess**  **sp\_addrole**  **sp\_droprole**  **sp\_approlepassword**  **sp\_password**  **sp\_changeobjectowner**  **sp\_defaultdb**  **sp\_defaultlanguage**  **sp\_denylogin**  **sp\_grantlogin**  **sp\_revokelogin**  **sp\_srvrolepermission**  **sp\_dbfixedrolepermission**  **sp\_srvrolepermission**  **sp\_dbfixedrolepermission**  GRANT ALL  DENY ALL  REVOKE ALL  USER\_ID  SETUSER  PERMISSIONS intrinsic function  RC4 encryption algorithm |
| SMO classes | **Microsoft.SQLServer.Management.Smo.Information** class  **Microsoft.SQLServer.Management.Smo.Settings** class  **Microsoft.SQLServer.Management.Smo.DatabaseOptions** class  **Microsoft.SqlServer.Management.Smo.DatabaseDdlTrigger.NotForReplication** property |
| SQL Server Agent | **net send** notification  Pager notification  ActiveX subsystem |
| SQL Server Management Studio | Solution Explorer integration in SQL Server Management Studio  Source Control integration in SQL Server Management Studio |
| System functions | **fn\_virtualservernodes**  **fn\_servershareddrives** |
| System tables | **sysaltfiles**  **syscacheobjects**  **syscolumns**  **syscomments**  **sysconfigures**  **sysconstraints**  **syscurconfigs**  **sysdatabases**  **sysdepends**  **sysdevices**  **sysfilegroups**  **sysfiles**  **sysforeignkeys**  **sysfulltextcatalogs**  **sysindexes**  **sysindexkeys**  **syslockinfo**  **syslogins**  **sysmembers**  **sysmessages**  **sysobjects**  **sysoledbusers**  **sysopentapes**  **sysperfinfo**  **syspermissions**  **sysprocesses**  **sysprotects**  **sysreferences**  **sysremotelogins**  **sysservers**  **systypes**  **sysusers**  **sys.numbered\_procedures**  **sys.numbered\_procedure\_parameters** |
| System views | **sys.sql\_dependencies** |
| Table compression | Use of the vardecimal storage format  Use of the **sp\_db\_vardecimal\_storage\_format** procedure  Use of the **sp\_estimated\_rowsize\_reduction\_for\_vardecimal** procedure |
| Table hints | Specifying NOLOCK or READUNCOMMITTED in the FROM clause of an UPDATE or DELETE statement  Specifying table hints without using the WITH keyword |
| Textpointers | TEXTPTR()  TEXTVALID()  READTEXT  UPDATETEXT  WRITETEXT |
| Transact-SQL syntax | * :: function-calling sequence * Three-part and four-part column references in SELECT list * A string enclosed in quotation marks used as a column alias for an expression in a SELECT list: for example, *'string\_alias' = expression* * Numbered procedures * *table\_name.index\_name* syntax in DROP INDEX * Not ending Transact-SQL statements with a semicolon * Use of #, ## as temporary table and temporary stored procedure names * Use of @, @@, or @@ as Transact-SQL identifiers * Use of DEFAULT keyword as default value * Use of a space as a separator between table hints * The select list of an aggregate indexed view must contain COUNT\_BIG (\*) in 90 compatibility mode * The indirect application of table hints to an invocation of a multi-statement table-valued function (TVF) through a view * GROUP BY ALL * ROWGUIDCOL as a column name in DML statements * IDENTITYCOL as a column name in DML statements * ALTER DATABASE syntax:   + MODIFY FILEGROUP READONLY   + MODIFY FILEGROUP READWRITE |
| XML | Inline XDR Schema Generation |
| Other | DB-Library  Embedded SQL for C |
| **Analysis Services** | **Feature Not Supported in a Future Version of SQL Server** |
| Connection string properties | InsertInto CreateCube |
| Features | SQL Server 2000 PMML |
| Multidimensional Expressions (MDX) | **Create Action**  CalculationPassValue  CalculationCurrentPass  NON\_EMPTY\_BEHAVIOR query optimizer hint was turned on by default |
| Other | CELL\_EVALUATION\_LIST intrinsic cell property |
| Objects | COM assemblies |
| **Replication** | **Feature Not Supported in a Future Version of SQL Server** |
| Replication | Adding publications to Active Directory  Attachable subscriptions  Checksum validation  PublisherAddress, PublisherNetwork, DistributorNetwork, and DistributorAddress parameters in Distribution and Merge Agents  Schema changes using **sp\_repladdcolumn** and **sp\_repldropcolumn**  SQL Distributed Management Objects (SQL-DMO)  Subscriber registration  **-UseInprocLoader** parameter |
| Transactional Replication | "No sync" subscriptions to transactional publications  Distribution ActiveX control  ODBC Subscriber  Replicating to Oracle 8 subscribers and from Oracle 8 publishers  Replication Distributor Interface  Subscription expiration for transactional publications  Transformable subscriptions  Updatable subscriptions including immediate updating and queued updating with snapshot and transactional publications |
| Merge Replication | @allow\_partition\_realignment property in sp\_addmergepublication  @delete\_tracking property in sp\_addmergearticle  Alternate synchronization partners  -ExchangeType parameter  Logical Records  Merge ActiveX control  Multicolumn UPDATE option  No sync subscriptions to merge publications  -ParallelUploadDownload parameter |
| **Other Tools and Features** | **Feature Not Supported in a Future Version of SQL Server** |
| Management Tools | Net send notifications (SQL Server Agent)  osql.exe  Pager notifications (SQL Server Agent)  SMO Class: Microsoft.SqlServer.Management.Smo.DatabaseDdlTrigger.NotForReplication property  SMO Class: Microsoft.SQLServer.Management.Smo.DatabaseOptions class  SMO class: Microsoft.SQLServer.Management.Smo.Information class  SMO class: Microsoft.SQLServer.Management.Smo.Settings class  SQL Server 2005 Registered Server API  SQL-DMO  SQLMail  The ActiveX subsystem (SQL Server Agent)  The Database Project System, including source-control integration, in SSMS |

## See Also – Breaking Changes:

Breaking Changes to SQL Server Features in SQL Server 2008:  
<http://msdn.microsoft.com/en-us/library/cc707784.aspx>

Breaking Changes to Analysis Services Features in SQL Server 2008:  
<http://msdn.microsoft.com/en-us/library/ms143742.aspx>

Breaking Changes in SQL Server Reporting Services:  
<http://msdn.microsoft.com/en-us/library/ms143380.aspx>

Breaking Changes to Integration Services Features in SQL Server 2008:  
<http://msdn.microsoft.com/en-us/library/aa337086.aspx>

Breaking Changes in SQL Server Replication:  
<http://msdn.microsoft.com/en-us/library/ms143470.aspx>

Breaking Changes to Management Tools Features in SQL Server 2008:  
<http://msdn.microsoft.com/en-us/library/cc879338.aspx>

Breaking Changes to Management Tools Features in SQL Server 2008:  
<http://msdn.microsoft.com/en-us/library/cc879338.aspx>

## See Also – Behavior Changes

Behavior Changes to Database Engine Features in SQL Server 2008:  
<http://msdn.microsoft.com/en-us/library/ms143359.aspx>

Behavior Changes to Analysis Services Features in SQL Server 2008:  
<http://msdn.microsoft.com/en-us/library/ms143682.aspx>

Behavior Changes in SQL Server Reporting Services:  
<http://msdn.microsoft.com/en-us/library/ms143200.aspx>

Behavior Changes to Integration Services Features in SQL Server 2008:  
<http://msdn.microsoft.com/en-us/library/bb500430.aspx>

Behavior Changes in SQL Server Replication:  
<http://msdn.microsoft.com/en-us/library/ms143733.aspx>

Behavior Changes to Management Tools Features in SQL Server 2008:  
<http://msdn.microsoft.com/en-us/library/cc879340.aspx>)

# Related Links

SQL Server 2008 Web site:

<http://www.microsoft.com/sqlserver/2008/en/us/default.aspx>

SQL Server 2008 Upgrade Technical Reference Guide:

<http://www.microsoft.com/downloads/details.aspx?FamilyID=3d5e96d9-0074-46c4-bd4f-c3eb2abf4b66&DisplayLang=en>

MSDN SQL Server Developer Center

<http://msdn.microsoft.com/en-us/sqlserver/default.aspx>

TechNet SQL Server Tech Center

<http://technet.microsoft.com/en-us/sqlserver/default.aspx>

Upgrading to SQL Server 2008

<http://msdn.microsoft.com/en-us/library/bb677622.aspx>

Did this paper help you? Please give us your feedback. Tell us on a scale of 1 (poor) to 5 (excellent), how would you rate this paper and why have you given it this rating? For example:

* Are you rating it high due to having good examples, excellent screen shots, clear writing, or another reason?
* Are you rating it low due to poor examples, fuzzy screen shots, or unclear writing?

This feedback will help us improve the quality of white papers we release.

[Send feedback](mailto:sqlfback@microsoft.com?subject=White%20Paper%20Feedback:%20[Title%20of%20White%20Paper]).