

SQL Server Setup Help

Installing SQL Server 2005



[See Also](#)



The SQL Server 2005 Installation Wizard is Microsoft Windows Installer-based, and provides a single feature tree for installation of all Microsoft SQL Server 2005 components:

- SQL Server Database Engine
- Analysis Services
- Reporting Services
- Notification Services
- Integration Services
- Management Tools
- Documentation and Samples

Note SQL Server 2005 is available in 32-bit and 64-bit editions. The 64-bit editions of SQL Server 2005 are installed using the same methods as the 32-bit editions: through the Installation Wizard, or via the command prompt. For more information on SQL Server 2005 components, see [Editions and Components of SQL Server 2005](#). Regardless of whether you use the SQL Server 2005 Installation Wizard or the command prompt to install SQL Server 2005, the Setup process involves the following steps.

▣ **Step 1: Prepare Your Computer to Install SQL Server 2005**

To prepare your computer for SQL Server 2005, review hardware and software requirements, System Configuration Checker requirements and blocking issues, and security considerations.

For more information: [Preparing to Install SQL Server 2005](#)

▣ **Step 2: Install SQL Server 2005**

To install SQL Server 2005, run Setup using the SQL Server 2005 Installation Wizard or install from the command prompt. You can also add components to an instance of SQL Server 2005, or upgrade to SQL Server 2005 from a previous SQL Server version.

For more information: [Installing SQL Server 2005 Components](#)

▣ **Step 3: Configure Your SQL Server 2005 Installation**

After Setup completes the installation of SQL Server 2005, you can configure SQL Server using graphical and command prompt utilities.

For more information: [Configuring a SQL Server Installation](#)

▣ **See Also**

Tasks

[How to: Upgrade to SQL Server 2005 \(Setup\)](#)

Other Resources

[Configuring High Availability](#)

[Database Engine Installation How-to Topics](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Editions and Components of SQL Server 2005



[See Also](#)



[Installing SQL Server 2005](#) >

Updated: **14 April 2006**

Installation requirements can vary widely, depending on your application needs. The different editions of SQL Server 2005 accommodate the unique performance, runtime, and price requirements of organizations and individuals. Which of the various SQL Server 2005 components you install will also depend on your organization or individual needs. The following sections help you understand how to make the best choice among the editions and components available in SQL Server 2005.

▣ **Deciding Among SQL Server 2005 Editions**

You can use all editions of SQL Server 2005 in production environments except for SQL Server 2005 Developer Edition and SQL Server 2005 Evaluation Edition. The following paragraphs describe the editions of SQL Server 2005.

SQL Server 2005 Enterprise Edition (32-bit and 64-bit)

Enterprise Edition scales to the performance levels required to support the largest enterprise online transaction processing (OLTP), highly complex data analysis, data warehousing systems, and Web sites. Enterprise Edition's comprehensive business intelligence and analytics capabilities and its high availability features such as failover clustering allow it to handle the most mission critical enterprise workloads. Enterprise Edition is the most comprehensive edition of SQL Server and is ideal for the largest organizations and the most complex requirements.

SQL Server 2005 Evaluation Edition (32-bit and 64-bit)

SQL Server 2005 is also available in a 180-day Evaluation Edition for 32-bit or 64-bit platforms. SQL Server Evaluation Edition supports the same feature set as SQL Server 2005 Enterprise Edition. You can upgrade SQL Server Evaluation Edition for production use.

SQL Server 2005 Standard Edition (32-bit and 64-bit)

SQL Server 2005 Standard Edition is the data management and analysis platform for small- and medium-sized organizations. It includes the essential functionality needed for e-commerce, data warehousing, and line-of-business solutions. Standard Edition's integrated business intelligence and high availability features provide organizations with the essential capabilities needed to support their operations. SQL Server 2005 Standard Edition is ideal for the small- to medium-sized organization that needs a complete data management and analysis platform.

SQL Server 2005 Workgroup Edition (32-bit only)

SQL Server 2005 Workgroup Edition is the data management solution for

small organizations that need a database with no limits on size or number of users. SQL Server 2005 Workgroup Edition can serve as a front-end Web server, or for departmental or branch office operations. It includes the core database features of the SQL Server product line, and is easily upgradeable to SQL Server 2005 Standard Edition or SQL Server 2005 Enterprise Edition. SQL Server 2005 Workgroup Edition is an ideal entry-level database that is reliable, robust, and easy-to-manage.

SQL Server 2005 Developer Edition (32-bit and 64-bit)

SQL Server 2005 Developer Edition lets developers build any type of application on top of SQL Server. It includes all of the functionality of SQL Server 2005 Enterprise Edition, but is licensed for use as a development and test system, not as a production server. SQL Server 2005 Developer Edition is an ideal choice for independent software vendors (ISVs), consultants, system integrators, solution providers, and corporate developers who build and test applications. You can upgrade SQL Server 2005 Developer Edition for production use.

SQL Server 2005 Express Edition (32-bit only)

The SQL Server Express database platform is based on SQL Server 2005. It is also a replacement for Microsoft Desktop Engine (MSDE). Integrated with Microsoft Visual Studio 2005, SQL Server Express makes it easy to develop data-driven applications that are rich in capability, secure in storage, and fast to deploy.

SQL Server Express is free and can be redistributed (subject to agreement), and functions as the client database, as well as a basic server database. SQL Server Express is an ideal choice for independent software vendors (ISVs), server users, non-professional developers, Web application developers, Web site hosts, and hobbyists building client applications. If you need more advanced database features, SQL Server Express can be seamlessly upgraded to more sophisticated versions of SQL Server.

SQL Server Express also offers additional components that are available as part of SQL Server 2005 Express Edition with Advanced Services (SQL Server Express). In addition to the features of SQL Server Express, SQL Server Express with Advanced Services contains the following features:

- SQL Server Management Studio Express (SSMSE), a subset of SQL Server Management Studio.
- Support for full-text catalogs.
- Support for viewing reports via Reporting Services.

SQL Server 2005 Compact Edition (32-bit only)

SQL Server Compact Edition is the compact database that extends enterprise data management capabilities to devices. SQL Server Compact Edition is capable of replicating data with SQL Server 2005 and SQL Server 2000, letting users maintain a mobile data store that is synchronized with the primary database. SQL Server Compact Edition is the only SQL Server edition that provides relational database management capabilities for smart devices.

SQL Server 2005 Runtime Edition (32-bit and 64-bit)

SQL Server 2005 Runtime Edition is offered through the Microsoft ISV Royalty Program. Under the end user license agreement for SQL Server 2005 Runtime Edition, an independent software vendor (ISV) may embed SQL Server code into their solution provided that the customer does not use the SQL Server code to run any other application, or use the SQL Server code in any other context. For more information about SQL Server Runtime Edition, see the Microsoft Knowledge Base article “[How to obtain the SQL Server Runtime license](http://support.microsoft.com/kb/917400)” (<http://support.microsoft.com/kb/917400>).

Using SQL Server 2005 with an Internet Server

On an Internet server, such as a server running Internet Information Services (IIS), you will typically install the SQL Server 2005 client tools. Client tools include the client connectivity components used by an application connecting to an instance of SQL Server.

Note:

Although you can install an instance of SQL Server on a computer running IIS, this is typically done only for small Web sites that have a single server computer. Most Web sites have their middle-tier IIS systems on one server or cluster of servers, and their databases on a separate server or federation of servers. SQL Server 2005 Reporting Services is typically installed on the

middle-tier. For more information about software requirements for Reporting Services, see [Hardware and Software Requirements for Installing SQL Server 2005](#). For more information about federations, see the **Federated Database Servers** topic in SQL Server 2005 Books Online.

Using SQL Server 2005 with Client/Server Applications

You can install just the SQL Server 2005 client components on a computer running client/server applications that connect directly to an instance of SQL Server. A client components installation is also a good option if you administer an instance of SQL Server on a database server, or if you plan to develop SQL Server applications.

The client components option installs the following SQL Server features: Command prompt tools, Reporting Services tools, connectivity components, programming models, management tools, development tools, Books Online, sample databases, and sample applications. For more information, see [How to: Install SQL Server 2005 \(Setup\)](#).

▣ Deciding Among SQL Server 2005 Components

Use the **Feature Selection** page of the SQL Server Installation Wizard to select the components to include in an installation of SQL Server 2005. None of the features in the tree are selected by default.

Use the following descriptions to determine the set of features that meets your needs.

Server components	Description
SQL Server Database Engine	Database Engine includes the Database Engine, the core service for storing, processing, and securing data, Replication, Full-text Search, and tools for managing relational and XML data.
Analysis Services	Analysis Services includes the tools for creating and managing online analytical processing (OLAP) and data mining applications.
Reporting Services ^{1,2}	Reporting Services includes server and client components for creating, managing, and deploying tabular, matrix, graphical, and free-form reports. Reporting Services is also an extensible platform that you can use to develop report applications.
Notification Services	Notification Services is a platform for developing and deploying applications that send personalized, timely information to subscribers on a variety of devices.
Integration Services	Integration Services is a set of graphical tools and programmable objects for moving, copying, and transforming data.

¹ Internet Information Services (IIS) 5.0 or higher is required for Reporting Services installations.

² Microsoft Internet Explorer 6.0 Service Pack (SP) 1 is required for the Report Designer component of Reporting Services.

Client components	Description
Connectivity	Installs components for communication between clients and servers, and network libraries for DB-Library, ODBC, and OLE

Components	DB.
------------	-----

Management tools	Description
SQL Server Management Studio ¹	SQL Server Management Studio (SSMS), new in Microsoft SQL Server 2005, is an integrated environment for accessing, configuring, managing, administering, and developing all components of SQL Server. SSMS combines the features of Enterprise Manager, Query Analyzer, and Analysis Manager, included in previous releases of SQL Server, into a single environment that provides SQL Server access to developers and administrators of all skill levels.
SQL Server Configuration Manager	SQL Server Configuration Manager provides basic configuration management for SQL Server services, server protocols, client protocols, and client aliases.
SQL Server Profiler	SQL Server Profiler provides a graphical user interface for monitoring an instance of the Database Engine or an instance of Analysis Services.
Database Engine Tuning Advisor	Database Engine Tuning Advisor helps create optimal sets of indexes, indexed views, and partitions.

¹ Internet Explorer 6.0 SP1 is required for SQL Server Management Studio installations.

Development Tools	Description
Business Intelligence Development Studio ¹	The Business Intelligence Development Studio is an integrated development environment for Analysis Services, Reporting Services, and Integration Services solutions.

¹ Internet Explorer 6.0 SP1 is required for Business Intelligence Development Studio installations.

Documentation and Samples	Description
SQL Server Books Online	Core documentation for SQL Server 2005.
SQL Server Samples	Provides sample code and sample applications for the Database Engine, Analysis Services, Reporting Services, and Integration Services.

▣ Change History

Release	History
17 July 2006	Changed content: <ul style="list-style-type: none">• Added SQL Server Runtime to the list of SQL Server 2005 editions.
14 April 2006	Changed content: <ul style="list-style-type: none">• Updated the length of the trial period for SQL Server 2005 Evaluation Edition from 120 to 180 days.• Revised the list of SQL Server 2005 editions that can be used in production environments: only Developer Edition and Evaluation Edition cannot be used in production environments.

▣ **See Also**

Reference

[Components to Install](#)

[Feature Selection](#)

[Version and Edition Upgrades](#)

Concepts

[SQL Server Surface Area Configuration](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

SQL Server 2005 Privacy Statement



[Installing SQL Server 2005](#) >

For information about user data collection components in Microsoft SQL Server 2005, see this [Microsoft Web Site](#).

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Legal Notice



[Installing SQL Server 2005](#) >

Updated: **17 July 2006**

This documentation, including sample applications herein, is provided for information purposes only, and this documentation is provided without any warranties, either express or implied. Information in this documentation, including URL and other Internet Web site references, is subject to change without notice. The entire risk of the use or the results of the use of this documentation remains with the user.

The primary purpose of a sample contained within this documentation is to illustrate a concept, or a reasonable use of a particular statement or clause. Most samples do not include all of the code that may normally be found in a full production system, as some of the usual data validation and error handling is removed to focus the sample on a particular concept or statement. Technical support is not available for these samples or for any included source code.

Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, people, places, and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, e-mail address, person, place, or event is intended or should be inferred. Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this documentation 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 documentation. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this documentation does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

© 2006 Microsoft Corporation. All rights reserved.

Microsoft, Windows, Windows NT, Windows Server, Active Directory, ActiveX, BackOffice, bCentral, BizTalk, DirectX, Excel, Hotmail, IntelliSense, J/Direct, Jscript, Microsoft Press, MSDN, MS-DOS, Outlook, PivotChart, PivotTable, PowerPoint, SharePoint, Visual Basic, Visual C#, Visual C++, Visual FoxPro, Visual InterDev, Visual J#, Visual J++, Visual SourceSafe, Visual Studio, Win32, Win32s, Windows Mobile, Windows Server System, SQL Server, and WinFX are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

SAP NetWeaver is the registered trademark of SAP AG in Germany and in several other countries.

All other trademarks are property of their respective owners.

□ **Documentation Policy for SQL Server 2005 Support and Upgrade**

Content that appears in SQL Server documentation is published only after it has been tested sufficiently. Product documentation – SQL Server Books Online, readme files, known issues documents, and Knowledge Base articles - contains content regarding SQL Server features and functionality that is robust enough to be safe for general use by all customers. This policy applies to all SQL Server documentation, including readme files for releases and services packs; a readme file is considered an extension of Books Online.

In some cases, a particular feature is not something that customers should use directly and, therefore, it is not documented. Unless a feature is also discussed in SQL Server documentation published by Microsoft, content from third party books or Web sites is not supported by Microsoft Customer Support Services (CSS) and should not be used in production databases or applications.

Customers must not use undocumented APIs, including but not limited to: stored procedures, extended stored procedures, functions, views, tables, columns, properties, or metadata. Microsoft Customer Support Services will not support databases or applications that leverage or use undocumented entry points.

Server and database upgrades to future versions of SQL Server are not guaranteed for applications and databases that leverage and use undocumented entry points. Use of SQL Server features and functionality must be limited to those that are included in Microsoft SQL Server documentation. If a feature is not documented in Microsoft SQL Server documentation, it is not a supported part of SQL Server.

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

About the SQL Server 2005 End User License Agreement



[Installing SQL Server 2005](#) >

Updated: **17 July 2006**

The Microsoft End User License Agreement (EULA) is copied to the local computer when SQL Server 2005 is installed. The files are written to:
%ProgramFiles%\Microsoft SQL Server\90\EULA\ or
%ProgramFiles(x86)%\Microsoft SQL Server\90\EULA\.

When multiple instances of the same SQL Server 2005 edition and language are installed on the same computer, a single copy of the EULA will apply to all instances of that edition and language.

The EULA file name indicates the edition of SQL Server 2005 as well as the SQL Server 2005 language edition. For example:

- License_DEV_ENU.txt for SQL Server 2005 Developer Edition, English language edition.
- License_STD_ITA.txt for SQL Server 2005 Standard Edition, Italian language edition.
- License_UA_JPN.txt for SQL Server 2005 Upgrade Advisor, Japanese language edition.
- License_SQLNCLI_ENU.txt for SQL Native Client, English language edition.
- License_SQLXML4_KOR.txt for SQLXML, Korean language edition.

The following file name segments are used to identify SQL Server 2005 editions:

- ENT - Enterprise Edition
- DEV - Developer Edition
- STD - Standard Edition
- WKGP - Workgroup Edition
- EXPR - Express Edition
- UA - Upgrade Advisor

The following file name segments are used to identify SQL Server 2005 languages:

- ENU - English
- JPN - Japanese
- FRA - French
- ITA - Italian
- KOR - Korean
- DEU - German
- ESN - Spanish
- CHS - Simplified Chinese
- CHT - Traditional Chinese
- NLA - Dutch
- PTB - Portuguese (Brazilian)
- SVE - Swedish
- RUS - Russian

▣ **SQL Server 2005 Licensing Options**

For information about SQL Server 2005 pricing and licensing options, see this [Microsoft Web site](#).

▣ Change History

Release	History
17 July 2006	New content: <ul style="list-style-type: none">• Added Russian to the list of file name segments for supported languages.

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Preparing to Install SQL Server 2005



[See Also](#)



Updated: **5 December 2005**

Depending on your requirements, Microsoft SQL Server 2005 Setup installs a new instance of SQL Server, upgrades an existing instance, or provides maintenance options. The following topics address important considerations for the computer where SQL Server 2005 will be installed. Before you install SQL Server 2005, review the following information.

▣ All SQL Server 2005 Installations

- Be sure the computer meets the system requirements for SQL Server 2005. For more information, see [Hardware and Software Requirements for Installing SQL Server 2005](#).
- Review [Security Considerations for a SQL Server Installation](#).
- Setup includes a preinstallation check that identifies unsupported configurations on installation computers and guides users to remedy blocking issues. For more information, see [Check Parameters for the System Configuration Checker](#).
- Make sure you have administrator permissions on the computer where SQL Server will be installed. If you install SQL Server from a remote share, you must use a domain account that has read and execute permissions on the remote share. To create a failover cluster, you must be a local administrator with permissions to logon as a service, and to act as part of the operating system on all nodes of the virtual server.

Security Note:

If you modify permissions on a domain user account to run SQL Server Setup, be sure to revoke those permissions after Setup completes.

- Create one or more domain user accounts if you are installing SQL Server 2005 on a computer running Microsoft Windows XP or Windows 2003, and you want SQL Server 2005 to communicate with other clients and servers. For more information, see [Setting Up Windows Service Accounts](#).
- Do not install SQL Server 2005 on a domain controller. For more information, see the "Isolation of Services" section in [Security Considerations for a SQL Server Installation](#).
- Back up your current installation of SQL Server if you are running an instance of SQL Server on the computer where SQL Server 2005 is to be installed.
- Verify that the disk where SQL Server will be installed is uncompressed. If you attempt to install SQL Server to a compressed drive, Setup will fail.
- Exit antivirus software while installing SQL Server.
- Stop all services that depend on SQL Server, including any service using Open Database Connectivity (ODBC), such as Internet Information Services (IIS). Exit Event Viewer and registry editors

(Regedit.exe or Regedt32.exe).

- Review all SQL Server installation options, and prepare to make the appropriate selections when running Setup. For more information, see [Editions and Components of SQL Server 2005](#), [Setup User Interface Reference](#), and [How to: Install SQL Server 2005 \(Setup\)](#).

▣ **Upgrading to SQL Server 2005**

- For a list of supported upgrade scenarios, see [Upgrading to SQL Server 2005](#).
- For information on running SQL Server 2005 along with previous SQL Server versions, see [Working with Multiple Versions of SQL Server](#) and [Backward Compatibility](#).
- For a procedural topic with step-by-step upgrade instructions, see [How to: Upgrade to SQL Server 2005 \(Setup\)](#).
- For known issues and backward compatibility considerations, see [Version and Edition Upgrades](#), [Using Upgrade Advisor to Prepare for Upgrades](#), and [Backward Compatibility](#).

▣ **Failover Cluster Installations**

If you are installing a failover cluster, see [Before Installing Failover Clustering](#).

▣ **International Users**

If you are using an operating system with regional settings other than English (United States), or if you are customizing character set or sort order settings, see [Collation Options and International Support](#).

For default collations assigned by Setup, see [Collation Settings in Setup](#).

▣ **See Also**

Concepts

[Installing SQL Server 2005](#)

[Failover Clustering](#)

Other Resources

[Installing SQL Server 2005 Components](#)

[Upgrading to SQL Server 2005](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Hardware and Software Requirements for Installing SQL Server 2005

 [See Also](#)



[Preparing to Install SQL Server 2005](#) >

Updated: **14 April 2006**

The following sections list the minimum hardware and software requirements for running Microsoft SQL Server 2005.

 **Note:**

Requirements for running SQL Server on the 32-bit platform are different from requirements for the 64-bit platform.

▣ **Hardware and Software Requirements (32-Bit and 64-Bit)**

Monitor

SQL Server graphical tools require VGA or higher resolution: at least 1,024x768 pixel resolution.

Pointing Device

A Microsoft mouse or compatible pointing device is required.

CD or DVD Drive

A CD or DVD drive, as appropriate, is required for installation from CD or DVD media.

Cluster Hardware Requirements

On 32-bit and 64-bit platforms, eight-node cluster installations (that is, the maximum number of nodes supported by Microsoft Windows Server 2003) are supported. For more information on high availability solutions, see [Configuring High Availability](#) and [Before Installing Failover Clustering](#).

Network Software Requirements

Network software requirements for the 64-bit versions of SQL Server 2005 are the same as the requirements for the 32-bit versions. Windows Server 2003, Windows XP, and Windows 2000 have built-in network software.

Note:

SQL Server 2005 does not support the Banyan VINES Sequenced Packet protocol (SPP), Multiprotocol, AppleTalk, or NWLink IPX/SPX network protocols. Clients previously connecting with these protocols must select a different protocol to connect to SQL Server 2005.

Stand-alone named and default instances support the following network protocols:

- Shared memory
- Named pipes
- TCP/IP
- VIA

Note Shared memory is not supported on failover clusters.

Internet Requirements

Internet requirements for both the 32-bit and 64-bit versions of SQL Server 2005 are the same. The following table lists the Internet requirements for SQL Server 2005.

Component	Requirement
Internet software ¹	Microsoft Internet Explorer 6.0 SP1 or later is required for all installations of SQL Server 2005, as it is required for Microsoft Management Console (MMC) and HTML Help. A minimal installation of Internet Explorer is sufficient, and Internet Explorer is not required to be the default browser. However, if you are installing client components only and you will not connect to a server that requires encryption, Internet Explorer 4.01 with Service Pack 2 is sufficient.
Internet Information Services (IIS)	IIS 5.0 or later is required for Microsoft SQL Server 2005 Reporting Services (SSRS) installations. For more information about how to install IIS, see How to: Install Microsoft Internet Information Services . For more information about how to install IIS 7.0 on Windows Vista, see this Microsoft Web site .
ASP.NET 2.0 ²	ASP.NET 2.0 is required for Reporting Services. When installing Reporting Services, SQL Server Setup will enable ASP.NET if it is not already enabled.

¹Microsoft Internet Explorer 6.0 SP1 or later is required for SQL Server Management Studio, Business Intelligence Development Studio, and the Report Designer component of Reporting Services.

²For Reporting Services (64-bit) installations on 64-bit servers, the 64-bit version of ASP.NET must be installed. For Reporting Services (32-bit) installations on the 32-bit subsystem (WOW64) of a 64-bit server, the 32-bit version of ASP.NET must be installed. Reporting Services is not supported in side-by-side configurations on the 64-bit platform and on the 32-bit subsystem (WOW64) of a 64-bit server at the same time. For more information, see [How to: Configure IIS to Run 32-bit Reporting Services in WOW64 Mode](#).

Software Requirements

SQL Server Setup requires Microsoft Windows Installer 3.1 or later and Microsoft Data Access Components (MDAC) 2.8 SP1 or later. You can download MDAC 2.8 SP1 from this [Microsoft Web site](#).

SQL Server Setup installs the following software components required by the product:

- Microsoft .NET Framework 2.0
- Microsoft SQL Server Native Client
- Microsoft SQL Server Setup support files

 **Note:**

SQL Server 2005 Express Edition does not install the .NET Framework 2.0. Before installing SQL Server 2005 Express Edition, you must download and install the .NET Framework 2.0 from this [Microsoft Web site](#).

If not already installed, SQL Server Setup installs each of these components separately; only the Microsoft SQL Server Setup support files are automatically removed when you uninstall SQL Server 2005. For more information on uninstalling this release, see [How to: Uninstall an Existing Instance of SQL Server 2005 \(Setup\)](#).

 **Note:**

The .NET Framework 2.0 Software Development Kit (SDK) is not installed by

SQL Server 2005. The SDK contains documentation, a C++ compiler, and other tools that are useful when using the .NET Framework for SQL Server development. You can download the .NET Framework SDK from this [Microsoft Web site](#).

After installing required components, SQL Server Setup will verify that the computer where SQL Server will be installed also meets all of the other requirements for a successful installation. For more information, see [Check Parameters for the System Configuration Checker](#).

SQL Server 2005 failover clusters require Microsoft Cluster Server (MSCS) on at least one node of your server cluster. MSCS is only supported if it is installed on a hardware configuration that has been tested for compatibility with the MSCS software. For more information, see [Before Installing Failover Clustering](#).

Cross-Language Support

For upgrades:

- English-language versions of SQL Server can be upgraded to any localized version of SQL Server 2005.
- Localized versions of SQL Server can be upgraded to localized versions of SQL Server 2005 of the same language.
- Localized version of SQL Server cannot be upgraded to the English-language version of SQL Server 2005.
- Localized versions of SQL Server cannot be upgraded to localized SQL Server 2005 versions of a different localized language.

All localized upgrades must follow supported upgrade paths. For specific information on supported upgrade paths, see [Version and Edition Upgrades](#).

Additional cross-language support:

- The English-language version of SQL Server 2005 is supported on all localized versions of supported operating systems.
- Localized versions of SQL Server 2005 are supported on localized operating systems that are the same language as the localized SQL Server

version.

- Localized versions of SQL Server 2005 are also supported on English-language versions of supported operating systems through the use of Windows Multilingual User Interface Pack (MUI) settings. However, you must verify certain operating system settings before installing a localized version of SQL Server 2005 on a server that is running an English-language operating system with a non-English MUI setting. You must verify that the following operating system settings match the language of the localized SQL Server to be installed:
 - The operating system user interface setting
 - The operating system user locale setting
 - The system locale setting

If these operating system settings do not match the language of the localized SQL Server, then you must correctly set these operating system settings as described in [How to: Change Operating System Settings to Support Localized Versions](#).

Virtual Operating System Support

You can install SQL Server 2005 on Microsoft Virtual Server and on Microsoft Virtual PC on supported operating systems and hardware. For information about Virtual Server, see this [Microsoft Web site](#). For information about Virtual PC, see this [Microsoft Web site](#).

Caution:

Do not install SQL Server 2005 on Microsoft Virtual Server or Microsoft Virtual PC on hardware that is not listed in the Hardware and Software requirements sections below.

▣ Hardware and Software Requirements (32-Bit Only)

Hardware Requirements (32-Bit)

This table shows hardware requirements for installing and running SQL Server 2005 on the 32-bit platform.

SQL Server 2005 (32-bit)	Processor type ¹	Processor speed ²	Memory (RAM) ³
SQL Server 2005 Enterprise Edition ⁴ SQL Server 2005 Developer Edition SQL Server 2005 Standard Edition	Pentium III-compatible processor or higher	Minimum: 600 MHz Recommended: 1 GHz or higher	Minimum: 512 MB Recommended: 1 GB or more Maximum: Operating system maximum
SQL Server 2005 Workgroup Edition	Pentium III-compatible processor or higher	Minimum: 600 MHz Recommended: 1 GHz or higher	Minimum: 512 MB Recommended: 1 GB or more Maximum: Operating system maximum
SQL Server 2005 Express Edition	Pentium III-compatible processor or higher	Minimum: 500 MHz Recommended: 1 GHz or higher	Minimum: 192 MB Recommended: 512 MB or more Maximum: Operating system maximum

SQL Server 2005 Express Edition with Advanced Services	Pentium III-compatible processor or higher	Minimum: 600 MHz Recommended: 1 GHz or higher	Minimum: 512 MB Recommended: 1 GB or more Maximum: Operating system maximum
--	--	--	---

¹System Configuration Checker (SCC) will block Setup if the requirement for processor type is not met.

²SCC will warn the user but will not block Setup if the minimum or recommended processor speed check is not met. No warning will appear on multiprocessor machines.

³SCC will warn the user but will not block Setup if the minimum or recommended RAM check is not met. Memory requirements are for this release only, and do not reflect additional memory requirements of the operating system. SCC verifies the memory available when Setup starts.

⁴ SQL Server Evaluation Edition supports the same feature set as SQL Server 2005 Enterprise Edition.

 **Note:**

SQL Server 2005 software works on the hardware certified for use with the Microsoft Windows operating system. For more information about hardware certified for use with the Windows operating system, see this Microsoft Web site: [Products Designed for Microsoft Windows - Windows Catalog and HCL](#).

Hard Disk Space Requirements (32-Bit and 64-Bit)

During installation of SQL Server 2005, Windows Installer creates temporary files on the system drive. Before you run Setup to install or upgrade to SQL Server 2005, verify that you have 2.0 GB of available disk space on the system drive for these files. This requirement applies even if you install all SQL Server

components to a non-default (system) drive. Many files will be installed to the system drive, usually drive C:. For example, Setup log files will be written to the system drive, and can require 80 MB for a stand-alone installation, and more for a failover cluster installation. For more information, see [File Locations for Default and Named Instances of SQL Server 2005](#).

Actual hard disk space requirements depend on your system configuration and the applications and features you choose to install. The following table provides disk space requirements for SQL Server 2005 components.

Feature	Disk space requirement
Database Engine and data files, Replication, and Full-Text Search	280 MB
Analysis Services and data files	90 MB
Reporting Services and Report Manager	120 MB
Notification Services engine components, client components, and rules components	50 MB
Integration Services	120 MB
Client Components	850 MB
SQL Server Books Online and SQL Server 2005 Compact Edition Books Online	240 MB
Samples and sample databases. Note that samples and sample databases are not installed by default.	410 MB

Operating System Requirements (32-Bit)

This table shows the operating systems that run the server software for each 32-bit version of SQL Server 2005.

	Enterprise Edition¹	Developer Edition	Standard Edition	Workgroup Edition	Express Edition and Express with Advanced Services	Evaluation Edition

Windows 2000	No	No	No	No	No	No
Windows 2000 Professional Edition SP4 ^{2, 4}	No	Yes	Yes	Yes	Yes	Yes
Windows 2000 Server SP4 ²	Yes	Yes	Yes	Yes	Yes	Yes
Windows 2000 Advanced Server SP4 ²	Yes	Yes	Yes	Yes	Yes	Yes
Windows 2000 Datacenter Edition SP4 ²	Yes	Yes	Yes	Yes	Yes	Yes
Windows XP Embedded	No	No	No	No	No	No
Windows XP Home Edition SP2	No	Yes	No	No	Yes	No
Windows XP Professional Edition SP2 ⁴	No	Yes	Yes	Yes	Yes	Yes
Windows XP Media Edition SP2	No	Yes	Yes	Yes	Yes	Yes
Windows XP Tablet Edition SP2	No	Yes	Yes	Yes	Yes	Yes
Windows						

Server 2003 Server SP1 ⁶	Yes	Yes	Yes	Yes	Yes	Yes
Windows Server 2003 Enterprise Edition SP1 ⁶	Yes	Yes	Yes	Yes	Yes	Yes
Windows Server 2003 Datacenter Edition SP1 ⁶	Yes	Yes	Yes	Yes	Yes	Yes
Windows Server 2003 Web Edition SP1	No	No	No	No	Yes	No
Windows Small Business Server 2003 Standard Edition SP1	Yes	Yes	Yes	Yes	Yes	Yes
Windows Small Business Server 2003 Premium Edition SP1	Yes	Yes	Yes	Yes	Yes	Yes
Windows Vista Starter Edition	No	No	No	No	No	No
Windows Vista Home Basic Edition	No	Yes ⁷	No	No	Yes ⁸	Yes ⁷
Windows						

Vista Home Premium Edition	No	Yes ⁷	No	No	Yes ⁸	Yes ⁷
Windows Vista Ultimate Edition	Yes ⁷	Yes ⁷	Yes ⁷	Yes ⁷	Yes ⁸	Yes ⁷
Windows Vista Business Edition	Yes ⁷	Yes ⁷	Yes ⁷	Yes ⁷	Yes ⁸	Yes ⁷
Windows Vista Enterprise Edition	Yes ⁷	Yes ⁷	Yes ⁷	Yes ⁷	Yes ⁸	Yes ⁷
Windows Server 2003 64-Bit Itanium Datacenter Edition SP1	No	No	No	No	No	No
Windows Server 2003 64-Bit Itanium Enterprise Edition SP1	No	No	No	No	No	No
Windows Server 2003 64-Bit x64 Standard Edition SP1 ⁶	WOW64 ³	WOW6				
Windows Server 2003 64-Bit x64 Datacenter Edition	WOW64 ³	WOW6				

SP1 ⁶						
Windows Server 2003 64-Bit x64 Enterprise Edition SP1 ⁶	WOW64 ³	WOW64 ³	WOW64 ³	WOW64 ³	WOW64 ³	WOW6
Windows XP x64 Professional 2003	No	WOW64 ³	WOW64 ³	WOW64 ³	WOW64 ³	No
Windows Vista Home Basic 64-Bit x64 Edition	No	WOW64 ^{3,7}	No	No	WOW64 ^{3,8}	WOW6
Windows Vista Home Premium 64-Bit x64 Edition	No	WOW64 ^{3,7}	No	No	WOW64 ^{3,8}	WOW6
Windows Vista Ultimate 64-Bit x64 Edition	No	WOW64 ^{3,7}	WOW64 ^{3,7}	WOW64 ^{3,7}	WOW64 ^{3,8}	WOW6
Windows Vista Business 64-Bit x64 Edition	No	WOW64 ^{3,7}	WOW64 ^{3,7}	WOW64 ^{3,7}	WOW64 ^{3,8}	WOW6
Windows Vista Enterprise 64-Bit x64 Edition	No	WOW64 ^{3,7}	WOW64 ^{3,7}	WOW64 ^{3,7}	WOW64 ^{3,8}	WOW6

¹ SQL Server 2005 Evaluation Edition supports the same feature set as SQL Server 2005 Enterprise Edition, but SQL Server 2005 Enterprise Edition is not supported on all of the operating systems that support Evaluation Edition.

²You can download Windows 2000 SP4 from this [Microsoft Web site](#). Service packs for Windows 2000 Datacenter Edition must be obtained through the original equipment manufacturer (OEM).

³These editions of SQL Server 2005 can be installed to the Windows on Windows (WOW64) 32-bit subsystem of a 64-bit server.

⁴You can install Microsoft SQL Server Books Online, client tools and some legacy tools for SQL Server 2005 Enterprise Edition on Windows 2000 Professional SP4, and Windows XP SP2. Client tools include SQL Server Management Studio, and Business Intelligence Development Studio, SQL Server 2005 software development kit. Legacy tools include the Data Transformation Services Runtime and SQL-DMO.

⁵ Reporting Services, which is installed as part of SQL Server Express with Advanced Services, will not install on operating systems that do not include Internet Information Services (IIS).

⁶ SQL Server 2005 support on Windows Server 2003 R2 is the same as support on Windows Server 2003 SP1.

⁷Requires SQL Server 2005 SP2 on Windows Vista.

⁸ SQL Server 2005 Express Edition requires SP1 on Windows Vista. SQL Server 2005 Express Edition with Advanced Services requires SP2 on Windows Vista.

The following limitations or issues affect installations on supported operating systems:

- Native Web Service (SOAP/HTTP) support is only available for instances of SQL Server 2005 running on Windows 2003.
- Individual topics in Microsoft SQL Server 2005 Integration Services (SSIS) programming, Analysis Management Objects (AMO), and ADOMD.NET documentation may indicate support for earlier versions of Windows, such

as Windows 98, Windows ME, or Windows NT 4.0. However, for this release, these three programming interfaces are only supported on Windows XP, Windows 2000, and Windows 2003.

- SQL Server 2005 failover clusters require Microsoft Cluster Server (MSCS) on at least one node of your server cluster. MSCS is only supported if it is installed on a hardware configuration that has been tested for compatibility with the MSCS software. For more information, see [Before Installing Failover Clustering](#).

Supported Clients (32-Bit)

SQL Server 2005 32-bit client components can be installed on Windows 2000 Professional SP4 or later.

 **Note:**

This release supports Tabular Data Stream (TDS) 4.2 client connectivity through legacy MDAC/DB-Library, not by using new SQL Server 2005 features.

Hardware and Software Requirements (64-Bit Only)

Hardware Requirements (64-Bit)

This table shows hardware requirements for installing and running SQL Server 2005 on the 64-bit platform.

SQL Server 2005 (64-bit)	Processor type ¹	Processor speed ²	Memory (RAM) ³
SQL Server 2005 Enterprise Edition 4	IA64 minimum: Itanium processor or higher	IA64 minimum: 1 GHz	IA64 minimum: 512 MB IA64 recommended: 1 GB or more IA64 maximum: 32 TB
SQL Server 2005 Developer Edition	x64 minimum: AMD Opteron, AMD Athlon 64, Intel Xeon with Intel EM64T support, Intel Pentium IV with EM64T support	IA64 recommended: 1 GHz or more x64 minimum: 1 GHz	Operating system maximum Minimum: 512 MB
SQL Server 2005 Standard Edition		x64 recommended: 1 GHz or more	x64 recommended: 1 GB or more x64 maximum: Operating system maximum

¹System Configuration Checker (SCC) will block Setup if the processor type requirement is not met.

²SCC will warn the user but will not block Setup if the minimum or recommended processor speed check is not met.

³SCC will warn the user but will not block Setup if the minimum or recommended RAM check is not met. Memory requirements are for this release only, and do not reflect additional memory requirements of the operating system. SCC verifies the memory available when Setup starts.

⁴ SQL Server 2005 Evaluation Edition supports the same feature set as SQL Server 2005 Enterprise Edition.

 **Note:**

SQL Server 2005 software works on the hardware certified for use with the Microsoft Windows operating system. For more information about hardware certified for use with the Windows operating system, see this Microsoft Web site: [Products Designed for Microsoft Windows - Windows Catalog and HCL](#). For information on installing a high availability solution, see [Configuring High Availability](#).

Operating System Requirements (64-Bit)

This table shows the operating systems that run the server software from each 64-bit version of SQL Server 2005.

	Enterprise Edition ¹ (IA64)	Enterprise Edition ¹ (X64)	Developer Edition (IA64) ²	Developer Edition (X64) ³	Standard Edition (IA64)	Standard Edition (X64)
Windows Server 2003 64-Bit						

Itanium Datacenter Edition SP1	Yes ⁴	No	Yes ⁴	No	Yes ⁴	No
Windows Server 2003 64-Bit Itanium Enterprise Edition SP1	Yes ⁴	No	Yes ⁴	No	Yes ⁴	No
Windows Server 2003 64-Bit x64 Standard Edition SP1 ⁵	No	Yes ⁴	No	Yes ⁴	No	Yes ⁴
Windows Server 2003 64-Bit x64 Datacenter Edition SP1 ⁵	No	Yes ⁴	No	Yes ⁴	No	Yes ⁴
Windows Server 2003 64-Bit x64 Enterprise Edition SP1 ⁵	No	Yes ⁴	No	Yes ⁴	No	Yes ⁴
Windows XP x64 Professional 2003	No	Yes ⁴	No	Yes ⁴	No	Yes ⁴
Windows Vista Home Basic 64- Bit x64 Edition	No	No	No	Yes ⁶	No	No
Windows						

Vista Home Premium 64-Bit x64 Edition	No	No	No	Yes ⁶	No	No
Windows Vista Ultimate 64-Bit x64 Edition	No	No	No	Yes ⁶	No	Yes ⁶
Windows Vista Business 64-Bit x64 Edition	No	No	No	Yes ⁶	No	Yes ⁶
Windows Vista Enterprise 64-Bit x64 Edition	No	No	No	Yes ⁶	No	Yes ⁶

¹ SQL Server 2005 Evaluation Edition supports the same feature set as SQL Server 2005 Enterprise Edition, but Enterprise Edition is not supported on all of the operating systems that support Evaluation Edition.

²IA64 = Intel Itanium architecture.

³X64 = AMD architecture / Intel Extended Systems architecture.

⁴Tools native/WOW64. For more information on WOW64, see Extended System Support.

⁵ SQL Server 2005 support on Windows Server 2003 R2 is the same as support on Windows Server 2003 SP1.

⁶Requires SQL Server 2005 SP2 on Windows Vista.

⁷ SQL Server 2005 Express Edition requires SQL Server 2005 SP1 on Windows Vista. SQL Server 2005 Express Edition with Advanced Services requires SQL

Server 2005 SP2 on Windows Vista.

Extended System Support

SQL Server 2005 64-bit versions include support for extended systems, also known as Windows on Windows (WOW64). WOW64 is a feature of 64-bit editions of Microsoft Windows that allows 32-bit applications to execute natively in 32-bit mode. Applications function in 32-bit mode even though the underlying operating system is running on the 64-bit platform.

Supported Clients (64-Bit)

SQL Server 2005 64-bit client components can be installed on Windows Server 2003 (64-bit).

▣ Change History

Release	History
12 December 2006	Changed content: <ul style="list-style-type: none">• Added Windows Vista support information.• Updated hard disk space requirements.

▣ **See Also**

Reference

[Editions and Components of SQL Server 2005](#)

[Features Supported by the Editions of SQL Server 2005](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Check Parameters for the System Configuration Checker



[See Also](#)



[Preparing to Install SQL Server 2005](#) >

Updated: **14 April 2006**

As part of SQL Server 2005 Setup, the System Configuration Checker (SCC) scans the computer where Microsoft SQL Server 2005 will be installed. The SCC checks for conditions that prevent a successful SQL Server installation. Before Setup starts the SQL Server 2005 Installation Wizard, the SCC retrieves the status of each check item, compares the result with required conditions, and provides guidance for removal of blocking issues. All of the SCC check items are network enabled; checks can run on a local computer, as well as in remote and cluster situations.

The following table describes check parameters for the SCC, requirements, and remedies that will allow Setup to successfully install SQL Server 2005.

Check item	Description	Remedy for failure
Windows Management Instrumentation (WMI) service requirement	WMI service must be available. A failed check on this item will block Setup.	The SCC will report the incompatibility.
Microsoft XML Core Services (MSXML) requirement	MSXML is required for some SQL Server 2005 components. If required, it will be installed by SQL Server Setup.	This is not a blocking issue for Setup.
Operating	The SCC checks the	For more information, see

System minimum level requirement	operating system for minimum requirements.	Hardware and Software Requirements for Installing SQL Server 2005.
Operating System version and service pack (SP) level requirement	<p>Microsoft Windows 2000 SP4, Microsoft Windows XP SP2, or Microsoft Windows Server 2003 is required. Any unsupported operating system blocks Setup.</p> <div data-bbox="487 777 852 1312" style="border: 1px solid black; padding: 5px;"> <p> Note:</p> <p>Users cannot apply a service pack to Datacenter operating system editions without going through the Original Equipment Manufacturer (OEM).</p> </div>	<p>The user is pointed to online Help to see the operating system requirement, or to Windows Update Component for installing Windows service packs on the destination computer. In cases where computers are running Datacenter editions, users must contact the OEM for service pack installation. For more information, see Hardware and Software Requirements for Installing SQL Server 2005.</p>
SQL Server compatibility with operating system requirement	<p>The operating system type must support the given SQL Server edition. For example, SQL Server Enterprise Edition can only be installed on Windows 2000 or later. A failed check on this item will block Setup.</p>	<p>The user is pointed to online help to see the matrix of operating system and SQL Server editions. For more information, see Hardware and Software Requirements for Installing SQL Server 2005</p>

<p>Minimum hardware requirement</p>	<p>Hardware must satisfy minimum CPU and memory requirements. SCC will warn the user but will not block Setup if the minimum or recommended processor speed check is not met. No warning will appear on multiprocessor machines.</p> <p>SCC verifies the memory available when Setup starts. SCC will warn the user but will not block Setup if the minimum or recommended RAM check is not met. Memory requirements are for this release only, and do not reflect additional memory requirements of the operating system.</p>	<p>For more information, see Hardware and Software Requirements for Installing SQL Server 2005.</p>
<p>Feature availability^{1,2}</p>	<p>The SQL Server feature selection tree will check for prerequisites before displaying SQL Server components available to install.</p>	<p>The SCC displays results in the SCC report, including features that will not be available and their dependencies.</p>
<p>Pending reboot</p>	<p>SQL Server Setup cannot run if files required by Setup are locked by other services, processes, or</p>	<p>The SCC will recommend that users reboot the operating system</p>

requirement	applications. A failed check on this item will block Setup.	and then run Setup again.
Performance monitor counter requirement	The SCC checks registry key values to verify proper incrementing for installation of SQL Server perfmon counters. A failed check on this item will block Setup.	Users must update the registry. To manually update the registry key, see How to: Increment the Counter Registry Key for Setup in SQL Server 2005 .
Default installation path permission requirement	The SCC will verify drive formatting and read/write permissions. Setup also requires that SQL Server be installed to an uncompressed drive. A failed check on this item will block Setup.	If the drive is unformatted, read-only, or compressed, the SCC will generate an error.
COM+ catalog requirement	The SCC will verify the current COM+ catalog configuration.	The SCC will identify issues and make recommendations. For more information, see the readme file or Setup Known Issues document on the SQL Server 2005 CD.
System administrator privilege requirement	The user running Setup must have administrator privileges on all destination computers: local, remote, and all cluster nodes where SQL Server will be installed.	The SCC will tell the user to log off and log on again either after adding administrator privileges to the account, or using another account that has administrator privileges.

Enable default administrative share directory requirement (Remote/Cluster only)	All cluster nodes must have the default administrative share admin\$ enabled, so that cluster installation processes can access its system files. A failed check on this item will block Setup.	The SCC will ask to fix the problem by enabling this default share directory.
---	---	---

¹ Microsoft Internet Information Services (IIS) 5.0 or later is required for Reporting Services installations.

² Microsoft Internet Explorer 6.0 SP1 is required for installations of the Report Designer component of Reporting Services.

▣ **See Also**

Reference

[Hardware and Software Requirements for Installing SQL Server 2005](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Security Considerations for a SQL Server Installation



[See Also](#)



[Preparing to Install SQL Server 2005](#) >

Security is important, not just for Microsoft SQL Server and Microsoft, but for every product and every business. By following simple best practices, you can avoid many security vulnerabilities. This topic discusses some security best practices that you should employ both before you install SQL Server and after you install SQL Server. Security guidance for specific features is included in the reference topics for those features.

▣ **Before Installing SQL Server**

Follow these best practices when you set up the server environment:

- [Enhance physical security](#)
- [Use firewalls](#)
- [Isolate services](#)
- [Create service accounts with least privileges](#)
- [Disable NetBIOS and server message block](#)

Enhance Physical Security

Physical and logical isolation make up the foundation of SQL Server security. To enhance the physical security of the SQL Server installation, do the following tasks:

- Place the server in a room inaccessible to unauthorized persons.
- Place computers that host a database in a physically protected location, ideally a locked computer room with monitored flood detection and fire detection or suppression systems.
- Install databases in the secure zone of the corporate intranet and never directly connected to the Internet.
- Back up all data regularly and store copies in a secure off-site location.

Use Firewalls

Firewalls are integral to securing the SQL Server installation. Firewalls will be most effective if you follow these guidelines:

- Put a firewall between the server and the Internet.
- Divide the network into security zones separated by firewalls. Block all traffic, and then selectively admit only what is required.
- In a multitier environment, use multiple firewalls to create screened subnets.
- When you are installing the server inside a Windows domain, configure interior firewalls to permit Windows Authentication.
- In a Windows domain in which all versions of Windows are Windows XP

- or Windows Server 2003 or higher, disable NTLM Authentication. .
- If your application uses distributed transactions, you may have to configure the firewall to allow Microsoft Distributed Transaction Coordinator (MS DTC) traffic to flow between separate MS DTC instances, and between the MS DTC and resource managers such as SQL Server.

Isolate Services

Isolating services reduces the risk that one compromised service could be used to compromise others. To isolate services, use the following guidelines:

- Whenever possible, do not install SQL Server on a domain controller.
- Run separate SQL Server services under separate Windows accounts.
- In a multitier environment, run Web logic and business logic on separate computers.

Create Service Accounts That Have Least Privileges

SQL Server setup automatically configures the service account or accounts with the specific permissions required by SQL Server. When modifying or configuring the Windows services used by SQL Server 2005, you should grant only the permissions they require. For more information, see [Setting Up Windows Service Accounts](#).

Disable NetBIOS and Server Message Block

Servers in the perimeter network should have all unnecessary protocols disabled including NetBIOS and server message block (SMB).

NetBIOS uses the following ports:

- UDP/137 (NetBIOS name service)
- UDP/138 (NetBIOS datagram service)
- TCP/139 (NetBIOS session service)

SMB uses the following ports:

- TCP/139

- TCP/445

Web servers and Domain Name System (DNS) servers do not require NetBIOS or SMB. On these servers, disable both protocols to mitigate the threat of user enumeration. For more information about how to disable these protocols, see [How to: Disable NetBIOS over TCP/IP](#) and [How to: Disable Server Message Block](#).

▣ **After Installing SQL Server**

After installation, you can enhance the security of the SQL Server installation by following these best practices regarding accounts and authentication modes:

Service accounts

- Run SQL Server services with the lowest possible privileges.
- Associate SQL Server services with Windows accounts.

Authentication mode

- Require Windows Authentication for connections to SQL Server.

Strong passwords

- Always assign a strong password to the **sa** account.
- Always enable password policy checking.
- Always use strong passwords for all SQL Server logins.

▣ **See Also**

Other Resources

Password Policy

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Considerations for Running SQL Server 2005 on Windows Vista



[Preparing to Install SQL Server 2005](#) >

Microsoft SQL Server 2005 requires SP2 to run on Windows Vista. For more information, see [Hardware and Software Requirements for Installing SQL Server 2005](#).

The following considerations apply when you install and running SQL Server 2005 SP2 on Windows Vista.

▣ Issues Caused by User Account Control in Windows Vista

Windows Vista includes a new feature, User Account Control (UAC) that helps administrators manage their use of elevated permissions. By default, on Windows Vista, administrators do not use their administrative rights. Instead, they perform most actions as standard (non-administrative) users, temporarily assuming their administrative rights only when it is necessary.

UAC causes some known issues. For more information, see the following Web pages on TechNet:

- [Windows Vista: User Account Control](#)
- [Windows Vista User Account Control Step by Step Guide](#)

Administrator Rights Not Inherited from Windows

On versions earlier than Windows Vista, members of the local Administrators group do not need their own SQL Server logins and they do not have to be granted administrative rights inside SQL Server. They connect to SQL Server as the built-in server principal BUILTIN\Administrators, and they have administrative rights inside SQL Server because BUILTIN\Administrators is a member of the **sysadmin** fixed server role.

On Windows Vista, these mechanisms are available only to administrative users who are running with elevated Windows permissions, which is not recommended. Instead, you should create a SQL Server login (using the FROM WINDOWS option) for each administrative user, and add that login to the **sysadmin** fixed server role. You should also do this for Windows accounts that are used to run SQL Server agent jobs. These include replication agent jobs.

To add a new Login to the sysadmin fixed server role while logged in as machinename\Administrator

1. Click **Start**, point to **All Programs**, point to **SQL Server 2005**, and then click **SQL Server Management Studio**.

2. Connect to SQL Server.
3. To add the Windows user to the **sysadmin** fixed server role, follow these steps:
 1. Click **Security**.
 2. Right-click **Logins**, and then click **New Login**.
 3. Type the user name in the **Login name** text box.
 4. Click **Server Roles**.
 5. Select the **sysadmin** check box, and then click **OK**.

To add a new Login to the sysadmin fixed server role while logged in as any user other than the administrator

1. Click **Start**, point to **All Programs**, point to **SQL Server 2005**, right-click **SQL Server Management Studio**, and then click **Run As Administrator**.

 **Note:**

Run as Administrator option elevates the user permissions.

2. You will see a **User Account Control** dialog box. You might have to provide the administrator credentials. Click **Continue**.
3. In **SQL Server Management Studio**, connect to SQL Server.
4. To add the Windows user to the **sysadmin** fixed server role, follow these steps:
 1. Click **Security**.
 2. Right-click **Logins**, and then click **New Login**.
 3. Type the user name in the **Login name** text box.
 4. Click **Server Roles**.
 5. Select the **sysadmin** check box, and then click **OK**.

Administrator Access Denied to a Report Server Deployment

UAC can prevent administrative access to a Report Server deployment. To gain administrative access, connect to SQL Server using the workaround described

earlier in this topic and create role assignments on Home and at the system-level for your account. You have to run SQL Server Management Studio with elevated permissions only for creating the role assignments for your account, and for gaining access to a report server if your role assignments have been deleted. After creating role assignments for your account, you can close SQL Server Management Studio and then reopen it with standard permissions. The role assignments that you created for your account provide sufficient rights for you to create additional role assignments for other users.

Local Access Denied to Report Server Virtual Directories

If you try to access Report Manager or the report server on a local computer using Internet Explorer, you will get an access denied error. To resolve this error, add Report Manager and the report server URL to Trusted Sites in Internet Explorer.

Notification Services Command Prompt Fails to Open

If you try to open Notification Services Command Prompt, it will fail with an "Access is denied" error. This is because accessing the Notification Services folder under *<drive>\Program Files\Microsoft SQL Server\90* requires administrative permissions.

To run Notification Services Command Prompt, either log on to the computer by using the Administrator account, or run Notification Services Command Prompt with elevated permissions.

To use the Administrator account

1. Log on to the computer by using the local Administrator account.
2. On the **Start** menu, click **All Programs**, click **Microsoft SQL Server 2005**, click **Configuration Tools**, right-click **Notification Services Command Prompt**, and then click **Run as administrator**.

To use elevated permissions

1. Log on to the computer by using an account that is a member of the local administrator group.

2. On the **Start** menu, click **Computer**.
3. Locate <drive>\Program Files\Microsoft SQL Server\90\NotificationServices.
4. When you see a message that states "You don't currently have permission to access this folder," click **Continue**.

 **Note:**

If you do not see this message, you already have permission to access the folder.

5. Right-click **Notification Services Command Prompt**, and then click **Run as administrator**.

Notification Services Commands in SQL Server Management Studio Return Errors

If you try to run Notification Services commands in SQL Server Management Studio, an "Object reference not set to an instance of an object" error might occur. This is because running some Notification Services commands requires administrative permissions.

To run SQL Server Management Studio with elevated permissions

1. Log on to the computer by using an administrator account.
2. On the **Start** menu, click **All Programs**, click **Microsoft SQL Server 2005**, right-click **SQL Server Management Studio**, and then click **Run as administrator**.

If you do not want to run SQL Server Management Studio with elevated permissions, use Notification Services Command Prompt to deploy and administer Notification Services.

Permissions to the Replication Snapshot Share Must Be Explicitly Granted

UAC can prevent administrative access to the snapshot share. You must explicitly grant snapshot share permissions to the Windows accounts that are used by the Snapshot Agent, Distribution Agent, and Merge Agent. You must do this even if the Windows accounts are members of the Administrators group. For more information, see "Replication Agent Security Model" in SQL Server Books Online.

▣ **SQL Server Performance Counters Are Not Visible in the Operating System Performance Monitor**

In this release of SQL Server 2005 SP2, SQL Server performance counters are not visible in the operating system performance monitor when you are running Windows Vista. However, SQL Server performance counters are provided in the **sys.dm_os_performance_counters** dynamic management view.

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Setting Up Windows Service Accounts

 [See Also](#)



[Preparing to Install SQL Server 2005](#) >

Updated: 12 December 2006

Each service in SQL Server represents a process or a set of processes to manage authentication of SQL Server operations with Microsoft Windows. This topic presents the default configuration of services in this release of SQL Server, and configuration options for SQL Server services that can be set during SQL Server installation.

Security Note Always run SQL Server services by using the lowest possible user rights.

Depending on the Microsoft SQL Server 2005 components you choose to install, SQL Server 2005 Setup installs the following services:

- SQL Server Database Services - The service for the SQL Server relational Database Engine.
- SQL Server Agent - Executes jobs, monitors SQL Server, fires alerts, and allows automation of some administrative tasks.

 **Note:**

For SQL Server and SQL Server Agent to run as services in Windows, SQL Server and SQL Server Agent must be assigned a Windows user account. Typically, both SQL Server and SQL Server Agent are assigned the same user account—either the local system or a domain user account. However, you can customize the settings for each service during the installation process. For more information about how to customize account information for each service, see [Service Account](#).

- Analysis Services - Provides online analytical processing (OLAP) and

- data mining functionality for business intelligence applications.
- Reporting Services - Manages, executes, renders, schedules, and delivers reports.
- Notification Services - A platform for developing and deploying applications that generate and send notifications.
- Integration Services - Provides management support for Integration Services package storage and execution.
- Full-Text Search - Quickly creates full-text indexes on content and properties of structured and semi-structured data to allow fast linguistic searches on this data.
- SQL Server Browser - The name resolution service that provides SQL Server connection information for client computers.
- SQL Server Active Directory Helper - Publishes and manages SQL Server services in Active Directory.
- SQL Writer - Allows backup and restore applications to operate in the Volume Shadow Copy Service (VSS) framework.

The remainder of this topic is divided into the following sections:

- [Configuring Services Exposed in SQL Server Setup](#)
- [Using Startup Accounts for SQL Server Services](#)
- [Identifying Instance-Aware and Instance-Unaware Services](#)
- [Reviewing NT Rights and Privileges Granted for SQL Server Service Accounts](#)
- [Reviewing Access Control Lists Created for SQL Server Service Accounts](#)
- [Reviewing Windows Permissions for SQL Server Services](#)
- [Reviewing Additional Considerations](#)
- [Localized Service Names](#)

▣ **Configuring Services Exposed in SQL Server Setup**

During SQL Server Setup, you can configure some SQL Server services with the startup account and whether the service auto-starts. The following table shows the SQL Server services that can be configured during installation. For unattended installations, you can use the switches in an installation file or from the command prompt.

SQL Server service name	Configurable in Installation Wizard?	Switches for unattended installations¹
MSSQLSERVER	Yes	SQLACCOUNT, SQLPASSWORD, SQLAUTOSTART
SQLServerAgent	Yes	AGTACCOUNT, AGTPASSWORD, AGTAUTOSTART
MSSQLServerOLAPService	Yes	ASACCOUNT, ASPASSWORD, ASAUTOSTART
ReportServer	Yes	RSACCOUNT, RSPASSWORD, RSAUTOSTART
SQLBrowser	Yes	SQLBROWSERACCOUNT, SQLBROWSERPASSWORD, SQLBROWSERAUTOSTART ²

¹For more information and sample syntax about remote and unattended installations, see [How to: Install SQL Server 2005 from the Command Prompt](#).

²SQLBROWSERAUTOSTART can be specified even if SQL Server Browser is already installed.

The following services cannot be configured at install time. They are installed with default settings:

- Notification Services
- Integration Services

- Full-Text Search
- Active Directory Helper
- SQL Writer

▣ Using Startup Accounts for SQL Server Services

To start and run, each service in SQL Server 2005 must have a user account. User accounts can be built-in system accounts or domain user accounts.

In addition to having user accounts, every service has three possible startup states that users can control:

- **Disabled** The service is installed but not currently running.
- **Manual** The service is installed but will start only when another service or application needs its functionality.
- **Automatic** The service is started by the operating system after device drivers are loaded at boot time.

The following table shows default and optional accounts for each SQL Server service, and the startup states for each service.

SQL Server service name	Default account	Optional accounts	Startup type	Default state following Setup
SQL Server Express Edition on Windows 2000:	Local system	SQL Server Express Edition: Domain User, Local System, Network Service ¹	Automatic ²	Started
SQL Server Express Edition on all other supported operating systems:	Network service	All other editions: Domain User, Local System, Network Service ¹		Stopped only if user chooses not to autostart
All other editions on all supported operating systems:	Domain user ¹			
SQL Server		Domain User, Local System, Network	Disabled	Stopped
			Automatic	Started

Agent	Domain user ³	Service ^{1,6}	only if user chooses to autostart	only if user chooses to autostart
Analysis Services	Domain user ³	Domain User, Local System, Network Service, Local Service	Automatic	Started Stopped only if user chooses not to autostart
Reporting Services	Domain user ³	Domain User, Local System, Network Service, Local Service	Automatic	Started Stopped only if user chooses not to autostart
Notification Services ⁴	N/A	N/A	N/A	N/A
Integration Services	Windows 2000: Local system All other supported operating systems: Network service	Domain User, Local System, Network Service, Local Service	Automatic	Started Stopped only if user chooses not to autostart.
Full-Text Search	Same account as SQL Server	Domain User, Local System, Network Service, Local Service	Manual	Stopped Started only if user chooses to autostart.
SQL Server Browser	SQL Server Express Edition on Windows 2000: Local system SQL Server Express Edition on all other supported operating systems: Local service	Domain User, Local System, Network Service, Local Service	Disabled ⁵ Automatic only if user chooses to	Stopped ⁵ Started only if user chooses to

	All other editions on all supported operating systems: Domain user ^{1,3}		autostart.	autostart.
SQL Server Active Directory Helper	Network service	Local System, Network Service	Disabled	Stopped
SQL Writer	Local system	Local System	Manual	Stopped

¹ **Important** Microsoft recommends that you not use the Network Service account for the SQL Server or the SQL Server Agent services. Local User or Domain User accounts are more appropriate for these SQL Server services.

²Set as manual in failover cluster configurations.

³For unattended installations, this property is required. If it is not specified, Setup will fail. To specify local system, use SQLAccount=LocalSystem or ASAccount=LocalSystem. For more information and sample syntax for remote and unattended installations, see [How to: Install SQL Server 2005 from the Command Prompt](#).

⁴ SQL Server Setup can install but will not configure Notification Services. For more information about enabling Notification Services after Setup, see the "Configuring Notification Services Windows Services" topic in SQL Server 2005 Books Online.

⁵For failover cluster installations, the SQL Server Browser is set to start automatically, and is started by default following Setup.

⁶For more information about supported Windows accounts for SQL Server Agent, see [The supported Windows account types that you can use to run the SQL Server Agent service in SQL Server 2005](#).

 **Important:**

For failover cluster installations, local system and local service accounts are not allowed for clustered services like SQL Server, SQL Server Agent, and

SSAS. For more information, see [Before Installing Failover Clustering](#).

For SQL Server 2005 installations in side-by-side configurations with earlier versions of SQL Server, SQL Server 2005 services must use accounts found only in the global domains group. Additionally, accounts used by SQL Server 2005 services must not appear in the local Administrators group. Failure to comply with this guideline will cause unexpected security behavior.

Using a Domain User Account

A domain user account may be preferred when the service must interact with network services. Many server-to-server activities can be performed only with a domain user account, for example:

- Remote procedure calls.
- Replication.
- Backing up to network drives.
- Heterogeneous joins that involve remote data sources.
- SQL Server Agent mail features and SQL Mail. This restriction applies if using Microsoft Exchange. Most other mail systems also require clients, like the SQL Server and SQL Server Agent services, to be run on accounts with network access.

Using the Local Service Account

The Local Service account is a special, built-in account that is similar to an authenticated user account. The Local Service account has the same level of access to resources and objects as members of the Users group. This limited access helps safeguard your system if individual services or processes are compromised. Services that run as the Local Service account access network resources as a null session with no credentials.

Using the Network Service Account

The Network Service account is a special, built-in account that is similar to an authenticated user account. The Network Service account has the same level of access to resources and objects as members of the Users group. Services that run

as the Network Service account access network resources using the credentials of the computer account.

 **Important:**

Microsoft recommends that you do not use the Network Service account for the SQL Server or SQL Server Agent services. Local User or Domain User accounts are more appropriate for these SQL services.

Using the Local System Account

The Local System account is a highly privileged account; use caution when assigning Local System permissions to SQL Server service accounts.

 **Security Note:**

To increase the security of your SQL Server installation, run SQL Server services under a local Windows account with the lowest possible privileges.

Changing User Accounts

To change the password or other properties of any SQL Server–related service, use SQL Server Configuration Manager. If your Windows password changes, be sure also to update the SQL Server services settings in Windows. If you have Kerberos enabled, be sure to update the Service Principal Name (SPN) directory property for Active Directory.

For more information, see [Changing Passwords and User Accounts](#). For information about using the Services add-in in Microsoft Windows to change SQL Server service accounts, see [How to change the SQL Server or SQL Server Agent service account without using SQL Enterprise Manager in SQL Server 2000 or SQL Server Configuration Manager in SQL Server 2005](#).

▣ **Identifying Instance-Aware and Instance-Unaware Services**

Some SQL Server services are instance-aware while others are instance-unaware. Each instance-aware service is associated with a specific SQL Server instance, and has its own registry hive. You can install multiple copies of instance-aware services by running SQL Server Setup for installation of each component or service. Instance-unaware services are shared among all installed SQL Server instances; they are not associated with a specific instance, are installed only once, and cannot be installed side-by-side.

Instance-aware services in Microsoft SQL Server 2005 include:

- SQL Server
- SQL Server Agent
- Analysis Services
- Reporting Services
- Full-Text Search

Instance-unaware services in SQL Server 2005 include:

- Notification Services
- Integration Services
- SQL Server Browser
- SQL Server Active Directory Helper
- SQL Writer

▣ Reviewing Windows NT Rights and Privileges Granted for SQL Server Service Accounts

SQL Server Setup creates user groups for different SQL Server services and adds the service accounts to them as appropriate. These groups simplify granting the permissions required to run SQL Server services and other executables, and help to secure SQL Server files. Note that unique group names are required when installing SQL Server 2005 on a domain controller.

User groups created by SQL Server Setup are granted the following Windows NT rights and privileges.

SQL Server service	User group
SQL Server	Default instance: <i>SQLServer2005MSSQLUser\$ComputerName\$MS</i> Named instance: <i>SQLServer2005MSSQLUser\$ComputerName\$Inst</i>

SQL Server Agent	<p>Default instance: SQLServer2005SQLAgentUser\$<i>ComputerName</i>\$M</p> <p>Named instance: SQLServer2005SQLAgentUser\$<i>ComputerName</i>\$I</p>
Analysis Services	<p>Default instance: SQLServer2005MSOLAPUser\$<i>ComputerName</i>\$M</p> <p>Named instance: SQLServer2005MSOLAPUser\$<i>ComputerName</i>\$In</p>
Reporting Services ¹	<p>Default instance: SQLServer2005ReportServerUser\$<i>ComputerName</i>\$MSSQLSERVE SQLServer2005ReportingServicesWebServiceUser\$<i>ComputerName</i></p> <p>Named instance: SQLServer2005ReportServerUser\$<i>ComputerName</i> and SQLServer2005ReportingServicesWebServiceUser\$<i>ComputerName</i></p>
Notification Services ²	<p>Default or named instance: SQLServer2005NotificationServicesUser</p>

Integration Services	Default or named instance: SQLServer2005DTSUser\$ <i>ComputerName</i>
Full-Text Search	Default instance: SQLServer2005MSFTEUser\$ <i>ComputerName</i> \$MS Named instance: SQLServer2005MSFTEUser\$ <i>ComputerName</i> \$Inst
SQL Server Browser	Default or named instance: SQLServer2005SQLBrowserUser\$Comp
SQL Server Active Directory Helper	Default or named instance: SQLServer2005MSSQLServerADHelperUser\$ <i>ComputerName</i>
SQL Writer	N/A

¹For Reporting Services, SQL Server Setup must also create SQLServer2005ReportingServicesWebServiceUser\$*InstanceName* and SQLServer2005ASP.NETUser user groups and grant access control lists (ACLs) to them. For more information, see the section below on access control lists.

² SQL Server Setup can install but will not configure Notification Services. For more information on enabling Notification Services after Setup, see the "Configuring Notification Services Windows Services" topic in SQL Server 2005 Books Online.

³ SQL Server Setup does not check or grant permissions for this service.

▣ Reviewing Access Control Lists Created for SQL Server Service Accounts

SQL Server 2005 service accounts must have access to resources. Access control lists (ACLs) are set at the user group level. The following table lists ACLs set by SQL Server Setup.

 **Important:**

For failover cluster installations, resources on shared disks cannot be set for any ACL in a local user group. Instead, those resources will be set to an ACL for a local account.

Service account for	Files and
MSSQLServer	Instid\MSSQL\backup
	Instid\MSSQL\binn
	Instid\MSSQL\data
	Instid\MSSQL\FTData
	Instid\MSSQL\Install
	Instid\MSSQL\Log
	Instid\MSSQL\Repldata
	90\shared
	90\shared\Error dumps
	90\com
	Instid\MSSQL\Template Data

SQLServerAgent	Instid\MSSQL\binn
	Instid\MSSQL\Log
	Instid\MSSQL\jobs
	90\com
	90\shared
	90\shared\Errordumps
FTS	Instid\MSSQL\FTData
	Instid\MSSQL\FTRef
	90\shared
	90\shared\Errordumps
	Instid\MSSQL\Install
MSSQLServerOLAPservice	90\shared
	90\shared\msmdlocal.ini
	Instid\OLAP
	Instid\Olap\Data
	Instid\Olap\Log
	90\shared\Errordumps
SQLServer2005ReportServerUser	Instid\Reporting Services\Log

	Instid\Reporting Services\Rep
	Instid\Reportingservices\Rep
	Instid\Reportingservices\Rep
	Instid\Reporting Services\rep
	Instid\Reporting Services\RS
	90\shared
	90\shared\Errordumps
SQLServer2005ReportingServicesWebServiceUser	Instid\Reporting Services\Log
	Instid\Reporting Services\Rep
	Instid\Reportingservices\Rep
	InstID\Reporting Services\rep
	Instid\Reportingservices\Rep
	Instid\Reporting Services\rep
	Instid\Reporting Services\RS
	Instid\Reporting Services\rep
	Instid\Reporting Services\rep
	Instid\Reporting Services\reportManager\web
	90\shared
	90\shared\Errordumps

Notification services	90\Notification services
	90\shared
	90\shared\Errordumps
MSDTSServer	90\dts\binn\MsDtsSrvr.ini.xml
	90\dts\binn
	90\shared
	90\shared\Errordumps
SQL Server Browser	90\shared\msmdlocal.ini
	90\shared
	90\shared\Errordumps
MSADHekper	N/A (Runs as built in account)
SQLWriter	N/A (Runs as local system)
User	Instid\MSSQL\binn
	Instid\Reporting Services\Rep
	Instid\Reportingservices\Rep
	InstID\Reporting Services\rep
	90\dts
	90\tools

	80\tools
	90\sdk
	Microsoft SQL Server\90\Set

In addition to SQL Server service startup accounts, access control permissions may have to be granted to built-in accounts or other SQL Server service accounts. The following table lists additional ACLs set by SQL Server Setup.

Requesting component	Account	Resource	
MSSQLServer	Performance Log Users	Instid\MSSQL\binn	L
	Performance Monitor Users	Instid\MSSQL\binn	L
	Performance Log Users	Instid\MSSQL\binn\sqlctr90.dll	R
	Performance Monitor Users	Instid\MSSQL\binn\sqlctr90.dll	R
	Administrator only	\\.\root\Microsoft\SqlServer\ServerEvents\ <sql_instance_name> ¹	F
	Administrators	\tools\binn\schemas\sqlserver\2003\03\showplan	F
	System	\tools\binn\schemas\sqlserver\2003\03\showplan	F
	Users	\tools\binn\schemas\sqlserver\2003\03\showplan	R
			D
			R
			S
			F
			F
			F
Reporting services	<Report Server Web Service Account>	<install>\Reporting Services\LogFiles	F
			F

			F
			F
			F
			F
			F
	Report Manager Application pool identity	<install>\Reporting Services\ReportManager	R
	ASP.NET account	<install>\Reporting Services\ReportManager	R
	Everyone	<install>\Reporting Services\ReportManager	R
	Report Manager Application pool identity	<install>\Reporting Services\ReportManager\Pages*.*	R
	ASP.NET account	<install>\Reporting Services\ReportManager\Pages*.*	R
	Everyone	<install>\Reporting Services\ReportManager\Pages*.*	R
	Report Manager Application pool identity	<install>\Reporting Services\ReportManager\Styles*.*	R
	ASP.NET account	<install>\Reporting Services\ReportManager\Styles*.*	R
	Everyone	<install>\Reporting Services\ReportManager\Styles*.*	R
	Report Manager Application pool identity	<install>\Reporting Services\ReportManager\webctrl_client\1_0*.*	R
	ASP.NET account	<install>\Reporting Services\ReportManager\webctrl_client\1_0*.*	R

	Everyone	<install>\Reporting Services\ReportManager\webctrl_client\1_0*.*	R
	<Report Server Web Service Account>	<install>\Reporting Services\ReportServer	R
	<Report Server Web Service Account>	<install>\Reporting Services\ReportServer\global.asax	F
	Everyone	<install>\Reporting Services\ReportServer\global.asax	R F F F
	Network service	<intsall>\Reporting Services\ReportServer\ReportService.asmx	F
	Everyone	<intsall>\Reporting Services\ReportServer\ReportService.asmx	R S F F F F F F D R S F

	ReportServer Windows Services Account	<i><install></i> \Reporting Services\ReportServer\RSReportServer.config	F F F F F F F F
	Everyone	Report Server keys (Instid hive)	Q E N R
	Terminal Services User	Report Server keys (Instid hive)	Q S C E N D R
			Q S C

	Power Users	Report Server keys (Instid hive)	E N D R
--	-------------	----------------------------------	------------------

¹This is the WMI provider namespace.

▣ Reviewing Windows Permissions for SQL Server Services

The following table shows service names, the term used to refer to the default and named instances of SQL Server services, a description of the service function, and required minimum permissions.

Display name	Service name	Description
SQL Server (<i>InstanceName</i>)	Default instance: MSSQLSERVER Named instance: MSSQL\$ <i>InstanceName</i>	SQL Server Database Engine. The path to the executable file is MSSQL\Binn\sqlservr.exe.

SQL Server Agent (<i>InstanceName</i>)	Default instance: SQLServerAgent Named instance: SQLAgent\$ <i>InstanceName</i>	Executes jobs, monitors SQL Server, allows automation of some administrative tasks. The path to the executable file is MSSQL\Binn\sqlagent90.exe.

Analysis Services Services (<i>InstanceName</i>)	Default instance: MSSQLServerOLAPService Named instance: MSOLAP\$ <i>InstanceName</i>	The service that provides online analytical processing (OLAP) and data mining functional intelligence applications. The path to the executable file is \SQLServer\OLAP\Bin\msmdsrv.exe.
Report Server	Default instance: ReportServer Named instance: ReportServer\$ <i>InstanceName</i>	Manages, executes, renders, schedules, and distributes reports. The path to the executable is \SQLServer\ReportServer\Bin\ReportServer.exe.
Notification Services		Notification Services is a platform for deploying applications that generate notifications. SQL Server Setup can configure Notification Services. For more information, see "Configuring Notification Services" topic in SQL Server 2005 Books Online.
Integration Services	Default or named instance: MSDTSServer	Provides management support for Integration Services package storage and execution. The path to the executable file is \SQLServer\MSDTS\Binn\msdtssrvr.exe.
SQL Server Browser	Default or named instance: SQLBrowser	The name resolution service that provides connection information for client connections. This service is shared across multiple SQL Server instances. The path to the executable file is \SQLServer\90\shared\sqlbrowser.exe.

Microsoft FullText Search (MSFTESQL)	Default instance: MSFTESQL Named instance: MSFTESQL\$ <i>InstanceName</i>	Quickly creates full-text indexes on properties of structured and semi-structured data to enable fast linguistic searches on this data. The path to the executable file is \MSSQL\Binn\msftesql.exe.
SQL Server Active Directory Helper	Default or named instance: MSSQLServerADHelper.	Publishes and manages SQL Server Active Directory. The path to the executable is \90\Sh
SQL Writer	SQLWriter	Allows backup and restore applications to use the Volume Shadow Copy Service (VSS) to create a backup of a single instance of the SQL Writer Server instances on the server. The path to the executable file is \90

¹For more information on how to verify that each of the required Windows permissions is set, see [How to: Verify Permissions for SQL Server Services](#).

▣ Reviewing Additional Considerations

The following table shows permissions required for SQL Server services in order for them to provide additional functionality.

Service/Application	Functionality	Required permission
SQL Server (MSSQLSERVER)	Write to a mail slot using xp_sendmail .	Network write privileges.
SQL Server (MSSQLSERVER)	Run xp_cmdshell for a user other than a SQL Server administrator.	Act as part of operating system and replace a process-level token.
SQL Server Agent (MSSQLSERVER)	Use the autorestart feature.	Must be a member of the Administrators local group.
Database Engine Tuning Advisor	Tunes databases for optimal query performance.	On first use, a user with system administrator privileges must initialize the application. After initialization, users who own tables (dbo users) can use the Database Engine Tuning Advisor to tune only those tables that they own. For more information, see "Initializing Database Engine Tuning Advisor on First Use" in SQL Server 2005 Books Online.

Important:

Before upgrading to SQL Server 2005, enable Windows Authentication for SQL Server Agent and verify that the SQL Server Agent service account is a member of the SQL Server **sysadmin** group.

▣ Localized Service Names

The following table shows service names used by localized version of Microsoft Windows.

Language	Name for Local Service	Name for Network Service	Name
English			
Simplified Chinese			
Traditional Chinese	NT AUTHORITY\LOCAL SERVICE	NT AUTHORITY\NETWORK SERVICE	NT AUT
Korean			
Japanese			
German	NT- AUTORITÄT\LOKALER DIENST	NT- AUTORITÄT\NETZWERKDIENTST	NT- AUT
French	AUTORITE NT\SERVICE LOCAL	AUTORITE NT\SERVICE RÉSEAU	AUT NT\S
Italian	NT AUTHORITY\SERVIZIO LOCALE	NT AUTHORITY\SERVIZIO DI RETE	NT AUT
Spanish	NT AUTHORITY\SERVICIO LOC	NT AUTHORITY\SERVICIO DE RED	NT AUT
Russian	NT AUTHORITY\LOCAL SERVICE	NT AUTHORITY\NETWORK SERVICE	NT AUT

▣ Change History

Release	History
12 December 2006	Changed content: <ul style="list-style-type: none">• Added security guidelines for service accounts in side-by-side configurations.
17 July 2006	Changed content: <ul style="list-style-type: none">• Revised content flow and presentation to make the topic more readable.• Added a link to the Knowledge Base article for using the Services add-in to change SQL Server and SQL Server Agent service accounts.• Added Russian localized service names.
5 December 2005	Changed content: <ul style="list-style-type: none">• Removed LocalService from the list of accounts SQL Server Agent can use.• Updated user groups created by SQL Server Setup.

▣ **See Also**

Reference

[Service Account](#)

[Service Accounts \(Clusters\)](#)

Concepts

[Security Considerations for a SQL Server Installation](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Features Supported by the Editions of SQL Server 2005



[See Also](#)



[Preparing to Install SQL Server 2005](#) >

Updated: **17 July 2006**

Use the following tables to determine which features are supported by the various editions of Microsoft SQL Server 2005.

▣ SQL Server 2005 Database Engine Features

The following abbreviations represent SQL Server 2005 editions:

- EE = SQL Server 2005 Enterprise Edition
- DE = SQL Server 2005 Developer Edition
- SE = SQL Server 2005 Standard Edition
- WG = SQL Server 2005 Workgroup Edition
- SSE = SQL Server 2005 Express Edition
- SSEA = SQL Server 2005 Express Edition with Advanced Services

Features/Database Engine Enhancements	EE (32-bit) (64-bit)	EE (32-bit) DE (32-bit) DE (64-bit)	SE (32-bit) SE (64-bit)	WG (32-bit)	SSE (32-bit)	SSEA (32-bit)
Programmability Features						
Microsoft .NET Framework	Yes		Yes	No	No	No
Common language runtime integration	Yes		Yes	Yes	Yes	Yes
User-defined Aggregates	Yes		Yes	Yes	Yes	Yes
User-defined Data Types	Yes		Yes	Yes	Yes	Yes
User-defined Functions	Yes		Yes	Yes	Yes	Yes
SQL Server Managed Provider (managed data access)	Yes		Yes	Yes	Yes	Yes
Extended Triggers	Yes		Yes	Yes	Yes	Yes
Eventing	Yes		Yes	Yes	Yes	Yes
Features/Database Engine Enhancements	EE (32-bit)		SE (32-bit)			
	EE (64-bit)		SE (64-bit)	WG (32-bit)	SSE (32-bit)	SSEA (32-bit)
	DE (32-bit)					
	DE (64-bit)					
Data Types						
CLR User-defined Data Types	Yes		Yes	Yes	Yes	Yes

New XML Data Type	Yes	Yes	Yes	Yes	Yes
Features/Database Engine Enhancements	EE (32-bit) EE (64-bit) DE (32-bit) DE (64-bit)	SE (32-bit) SE (64-bit)	WG (32-bit)	SSE (32-bit)	SSEA (32-bit)
Transact-SQL Enhancements					
Exception Handling	Yes	Yes	Yes	Yes	Yes
Extensions for Querying Support	Yes	Yes	Yes	Yes	Yes
FOR XML Enhancements	Yes	Yes	Yes	Yes	Yes
Intrinsics (String Function)	Yes	Yes	Yes	Yes	Yes
LOB Programming Model	Yes	Yes	Yes	Yes	Yes
New Analytic Functions (Ranking)	Yes	Yes	Yes	Yes	Yes
New Apply Operator (Relational Operator)	Yes	Yes	Yes	Yes	Yes
New Pivot and Unpivot Operators (Relational Operators)	Yes	Yes	Yes	Yes	Yes
Notification Support	Yes	Yes	Yes	Yes	Yes
Recursive Queries	Yes	Yes	Yes	Yes	Yes
Top Enhancements	Yes	Yes	Yes	Yes	Yes
Distributed Query	Yes	Yes	Yes	Yes	Yes
Features/Database Engine Enhancements	EE (32-bit) EE (64-bit) DE (32-bit) DE (64-bit)	SE (32-bit) SE (64-bit)	WG (32-bit)	SSE (32-bit)	SSEA (32-bit)
Full-Text Search Enhancements					
Full-Text Linked Server					

Queries	Yes	Yes	Yes	No	Yes
Full-Text Search Across Multiple Columns	Yes	Yes	Yes	No	Yes
Specification of Language in Query	Yes	Yes	Yes	No	Yes
New Message Service Broker	Yes	Yes	Yes	Client only	Client only
HTTP Support (Native HTTP)	Yes	Yes	Yes	No	No
Features/Database Engine Enhancements	EE (32-bit) EE (64-bit) DE (32-bit) DE (64-bit)	SE (32-bit) SE (64-bit)	WG (32-bit)	SSE (32-bit)	SSEA (32-bit)
Manageability Features					
Backup and Restore of Full-text Catalogs	Yes	Yes	Yes	No	Yes
Full-text catalogs included in database attach/detach operations	Yes	Yes	Yes	No	Yes
Full-Text Search Upgrade	Yes	Yes	Yes	No	Yes
Query Against Views (Formerly Indexing of Views)	Yes	Yes	Yes	No	No
Indexing of XML Datatype	Yes	Yes	Yes	No	No
Index Performance Improvements (formerly Inline Gathering , MSSearch 3.0)	Yes	Yes	Yes	No	No
Parallel Service Security (formerly Security Context Change)	Yes	Yes	Yes	No	No
Rich Status Reporting	Yes	Yes	Yes	No	No

Features/Database Engine Enhancements	EE (32-bit) EE (64-bit) DE (32-bit) DE (64-bit)	SE (32-bit) SE (64-bit)	WG (32-bit)	SSE (32-bit)	SSEA (32-bit)
Security Features					
All Permissions Grantable	Yes	Yes	Yes	Yes	Yes
Fine-Grained Administration Rights	Yes	Yes	Yes	Yes	Yes
Integrated Security for SQL Authorization	Yes	Yes	Yes	Yes	Yes
Separation of Users and Schema	Yes	Yes	Yes	Yes	Yes
Features/Database Engine Enhancements	EE (32-bit) EE (64-bit) DE (32-bit) DE (64-bit)	SE (32-bit) SE (64-bit)	WG (32-bit)	SSE (32-bit)	SSEA (32-bit)
Availability Features (Database Failure and Redundancy)					
Failover Clustering	Yes	2-node	No	No	No
Multi-Instance Support	50	16	16	16	16
Database Mirroring	Yes	Safety FULL only	No	No; can be used as a witness	No; can be used as a witness
Log Shipping	Yes	Yes	Yes	No	No
Database Snapshot	Yes	No	No	No	No
Features/Database Engine Enhancements	EE (32-bit) EE (64-bit) DE (32-bit)	SE (32-bit) SE (64-bit)	WG (32-bit)	SSE (32-bit)	SSEA (32-bit)

	DE (64-bit)				
Database Maintenance Features					
Data and Backup Checksum Integrity Checks	Yes	Yes	Yes	Yes	Yes
Dedicated Administrator Connection	Yes	Yes	Yes	Yes ¹	Yes ¹
Dynamic AWE	Yes	Yes	No	No	No
Database Available during Recovery Undo	Yes	No	No	No	No
Highly-available Upgrade	Yes	No	No	No	No
Hot Add Memory ²	Yes	No	No	No	No
Mirrored Backup Media	Yes	No	No	No	No
Online Index Operations	Yes	No	No	No	No
Page Restore	Yes	Yes	Yes	Yes	Yes
Online Page and File Restore	Yes	No	No	No	No
Parallel Index Operations	Yes	No	No	No	No
Features/Database Engine Enhancements	EE (32-bit) EE (64-bit) DE (32-bit) DE (64-bit)	SE (32-bit) SE (64-bit)	WG (32-bit)	SSE (32-bit)	SSEA (32-bit)
XML Features					
XML Datatype	Yes	Yes	Yes	Yes	Yes
Xquery Support	Yes	Yes	Yes	Yes	Yes
XML Data Manipulation Language (XML DML)	Yes	Yes	Yes	Yes	Yes
XML Views	Yes	Yes	Yes	Yes	Yes
FOR XML Enhancements	Yes	Yes	Yes	Yes	Yes
XML Type System Support (XML Schema Support)	Yes	Yes	Yes	Yes	Yes
Features/Database Engine Enhancements	EE (32-bit) EE (64-bit)	SE (32-bit)	WG (32-bit)	SSE (32-bit)	SSEA (32-bit)

	DE (32-bit)	SE (64-bit)	WG (32-bit)	SSE (32-bit)	SSEA (32-bit)
	DE (64-bit)				
Scalability and Performance Features					
Database Engine Tuning Advisor (Replaces Index Tuning Wizard)	Yes	Yes	Yes	No	No
Updateable Distributed Partitioned Views	Yes	No	No	No	No
Enhanced Read-ahead and Scan (Super Scan)	Yes	No	No	No	No
Indexed Views (Materialized Views) ³	Yes	Yes	Yes	Yes	Yes
Multiple Active Result Sets (MARS)	Yes	Yes	Yes	Yes	Yes
Persisted Computed Columns	Yes	Yes	Yes	Yes	Yes
Queuing Support	Yes	Yes	Yes	Yes	Yes
Snapshot Isolation Level (Row-level Versioning)	Yes	Yes	Yes	Yes	Yes
Table and Index Partitioning	Yes	No	No	No	No
VIA Support	Yes	No	No	No	No
Parallel DBCC	Yes	No	No	No	No
Features/Database Engine Enhancements	EE (32-bit)	SE (32-bit)	WG (32-bit)	SSE (32-bit)	SSEA (32-bit)
	EE (64-bit)	SE (64-bit)			
	DE (32-bit)				
	DE (64-bit)				
International Features (Collation, Character Support, and Global Deployment)					
Complex Script/Bi-directional Support	Yes	Yes	Yes	Yes	Yes
GB18030 Support	Yes	Yes	Yes	Yes	Yes

Surrogate Support	Yes	Yes	Yes	Yes	Yes
Language Pack	Yes	Yes	Yes	Yes	Yes

¹The Dedicated Administrator Connection feature for SQL Server Express Edition is supported with the trace flag 7806. For more information on trace flags, see the "Trace Flags (Transact-SQL)" topic in SQL Server 2005 Books Online.

²This feature requires SQL Server 2005 Enterprise Edition and is only available for 64-bit SQL Server, and for 32-bit SQL Server when AWE is enabled.

³This feature is supported in all editions of SQL Server 2005, except that automatic matching by Query Optimizer is supported only in Enterprise Edition and Developer Edition.

▣ SQL Server 2005 Management Tools Features

The following abbreviations represent SQL Server 2005 editions:

- EE = SQL Server 2005 Enterprise Edition
- DE = SQL Server 2005 Developer Edition
- SE = SQL Server 2005 Standard Edition
- WG = SQL Server 2005 Workgroup Edition
- SSE = SQL Server 2005 Express Edition
- SSEA = SQL Server 2005 Express Edition with Advanced Services

Features/Management Tools Enhancements	EE (32-bit) DE (32-bit) EE (64-bit) DE (64-bit)	SE (32-bit)	WG (32-bit)	SE (64-bit)	SSE (32-bit)	SSEA (32-bit)
Programmability Features						
Design Tools Enhancements						
Query Designer	Yes	Yes	Yes	No	No	No
Query Editor	Yes	Yes	Yes	No	No	No
MDX Query Editor	Yes	Yes	Yes	No	No	No
MDX IntelliSense	Yes	Yes	Yes	No	No	No
Transact-SQL IntelliSense	Yes	Yes	Yes	No	No	No
Version Control Support	Yes	Yes	Yes	No	No	No
XML/A	Yes	Yes	Yes	No	No	No
Features/Management Tools Enhancements	EE (32-bit) DE (32-bit) EE (64-bit) DE (64-bit)	SE (32-bit)	WG (32-bit)	SE (64-bit)	SSE (32-bit)	SSEA (32-bit)
Management Programming						

SQL Server Management Objects (SMO)	Yes	Yes	Yes	No	Yes	Yes
WMI Configuration Provider	Yes	Yes	Yes	No	No	No
Manageability Features						
Configuration						
SQL Server Configuration Manager, MMC Snap-in	Yes	Yes	Yes	Yes	Yes	Yes
Features/Management Tools Enhancements	EE (32-bit) DE (32-bit) EE (64-bit) DE (64-bit)	SE (32-bit)	WGSE (32-bit)	(64-bit)	SSE (32-bit)	SSEA (32-bit)
Management						
Microsoft Visual Studio Integration	Yes	Yes	Yes	Yes	Yes	Yes
New Dialog Style (Fully Scriptable and Schedulable User Interface)	Yes	Yes	Yes	No	No	No
SQL Server Management Studio	Yes	Yes	Yes	Yes	No	SQL Server Management Studio Express (a subset of SQL Server Management Studio)
Master Server	Yes (automatic)	Yes (manual)	No	No	No	No
Target Server	Yes	Yes	Yes	No	No	No
Features/Management	EE (32-bit) DE (32-bit)	SE (32-	WGSE	SSE		

Tools Enhancements	EE (64-bit)	bit)	(32-bit)	(64-bit)	(32-bit)	SSEA (32-bit)
	DE (64-bit)					
Operations						
Microsoft Operations Manager pack for SQL Server 2005	Yes	Yes	Yes	Yes	No	No
SQLCMD Command Prompt Tools (formerly OLESQL)	Yes	Yes	Yes	Yes	Yes	Yes
SQL Server Agent Job Reuse	Yes	Yes	Yes	Yes	No	No
SQL Server Agent Performance Counters	Yes	Yes	Yes	Yes	No	No
SQL Server Agent Scheduling	Yes	Yes	Yes	Yes	No	No
SQL Server Agent Security	Yes	Yes	Yes	Yes	No	No
SQL Server Agent Subsystems for OLAP and SSIS	Yes	Yes	Yes	Yes	No	No
SQL Server Agent Web Assistant	Yes	Yes	Yes	Yes	No	No
SQL Mail	Yes on 32-bit No on 64-bit	Yes	Yes	No	No	No
Database Mail	Yes on 32-bit No on 64-bit	Yes	Yes	No	No	No
Features/Management Tools Enhancements	EE (32-bit) DE (32-bit) EE (64-bit) DE (64-bit)	SE (32-bit)	WG (32-bit)	SE (64-bit)	SSE (32-bit)	SSEA (32-bit)
Scalability and						

Performance						
Database Engine Tuning Advisor	Yes	Yes	Yes	Yes	No	No
Performance Counter Correlation	Yes	Yes	Yes	Yes	No	No
SQL Server Profiler Enhancements	Yes	Yes	Yes	Yes	No	No
Profiling Analysis Services	Yes	Yes	Yes	Yes	No	No
Exportable Showplan and Deadlock Traces	Yes	Yes	Yes	Yes	No	No
Features/Management Tools Enhancements	EE (32-bit) DE (32-bit) EE (64-bit) DE (64-bit)	SE (32-bit)	WG (32-bit)	SE (64-bit)	SSE (32-bit)	SSEA (32-bit)
International Features (Collation, Character Support, and Global Deployment)						
Complex Script/ Bi-directional Support	Yes	Yes	Yes	Yes	Yes	Yes
GB18030 Support	Yes	Yes	Yes	Yes	Yes	Yes
Surrogate Support	Yes	Yes	Yes	Yes	Yes	Yes

▣ SQL Server 2005 Analysis Services Features

The following abbreviations represent SQL Server 2005 editions:

- EE = SQL Server 2005 Enterprise Edition
- DE = SQL Server 2005 Developer Edition
- SE = SQL Server 2005 Standard Edition
- WG = SQL Server 2005 Workgroup Edition
- SSE = SQL Server 2005 Express Edition
- SSEA = SQL Server 2005 Express Edition with Advanced Services

Features/Analysis Services Enhancements	EE (32-bit) DE (32-bit) EE (64-bit) DE (64-bit)	SE (32-bit)	WG (32-bit)	SSE (32-bit) SSEA (32-bit)	SE (64-bit)
Programmability Features					
.NET Framework	Yes	Yes	No	No	Yes
.NET Stored Procedures	Yes	Yes	Yes	Yes	Yes
Cross Platform, Zero Footprint Client Access	Yes	Yes	No	No	Yes
Native XML Web services	Yes	Yes	No	No	Yes
HTTP access	Yes	Yes	No	No	Yes
Binary and compressed XML transport	Yes	No	No	No	No
Manageability Features					
Auto Referential Integrity Handling	Yes	Yes	No	No	Yes
Capture and Replay	Yes	Yes	No	No	Yes
Fine-Grain Administration Rights	Yes	Yes	No	No	Yes
Profiler	Yes	Yes	No	No	Yes
Trace	Yes	Yes	No	No	Yes
Flight Recorder	Yes	Yes	No	No	Yes
Features/Analysis Services	EE (32-bit) DE (32-bit)	SE (32-bit)	WG (32-bit)	SSE (32-bit) SSEA (32-bit)	SE (64-bit)

Enhancements	EE (64-bit) DE (64-bit)	bit)	bit)	SSEA (32-bit)	bit)
Management Programming Features					
Analysis Management Objects (AMO and DMO)	Yes	Yes	No	No	Yes
Availability Features					
Enhanced Backup and Restore	Yes	Yes	No	No	Yes
Failover Clustering	Yes	Yes	No	No	Yes
Multi-Instances	50	16	No	No	16
Server Synchs	Yes	Yes	No	No	Yes
Features/Analysis Services Enhancements	EE (32-bit) DE (32-bit) EE (64-bit) DE (64-bit)	SE (32-bit)	WG (32-bit)	SSE (32-bit) SSEA (32-bit)	SE (64-bit)
Business Intelligence Features					
Auto Packaging and Deployment	Yes	Yes	No	No	Yes
Business Intelligence Wizard	Yes	Yes	No	No	Yes
Data Source Views Management	Yes	Yes	No	No	Yes
Integrated Business Intelligence Development Studio	Yes	Yes	No	No	Yes
Autobuild Builder Technology	Yes	Yes	No	No	Yes
MDX Debugger	Yes	Yes	No	No	Yes
Multi-Partition Editor	Yes	Yes	No	No	Yes
Source Control and Versioning	Yes	Yes	No	No	Yes
Unified Calculation Editor	Yes	Yes	No	No	Yes
Features/Analysis Services Enhancements	EE (32-bit) DE (32-bit) EE (64-bit)	SE (32-bit)	WG (32-bit)	SSE (32-bit) SSEA (32-bit)	SE (64-bit)

	DE (64-bit)				
Data Mining Features					
Enhanced Integrated OLAP and Data Mining Functionality (MDX Prediction function, DM dimensions)	Yes	Yes	No	No	Yes
Parallelism for model processing	Yes	No	No	No	No
Parallelism for model prediction	Yes	No	No	No	No
Text-Mining Term Extraction Transformation (SSIS)	Yes	No	No	No	No
Text-Mining Term Lookup Transform (SSIS)	Yes	No	No	No	No
Data Mining Query Transformation (SSIS)	Yes	No	No	No	No
Data Mining Processing Destination (SSIS)	Yes	No	No	No	No
Algorithm Plug-in API	Yes	No	No	No	No
Algorithm Viewers	Yes	Yes	No	No	Yes
Data Mining Tools (including Wizards, Editors, Query Builders, Lift Chart)	Yes	Yes	No	No	Yes
Standard Data Mining Algorithms	Yes	Yes	No	No	Yes
Advanced configuration and tuning options for Data Mining algorithms	Yes	No	No	No	No
Reporting Integration with DM Prediction Queries	Yes	Yes	No	No	Yes
Features/Analysis Services Enhancements	EE (32-bit)			SSE (32-bit)	
	DE (32-bit)	SE (32-bit)	WG (32-bit)		SE (64-bit)
	EE (64-bit)			SSEA (32-bit)	
	DE (64-bit)				
Advanced Analytic Features					
Account Intelligence	Yes	No	No	No	No
Cross-database/cross-server linked measures and dimensions	Yes	No	No	No	No
Metadata translations	Yes	No	No	No	No

Perspectives	Yes	No	No	No	No
Semi-additive Measures	Yes	No	No	No	No
Writeback Dimensions	Yes	No	No	No	No
Features/Analysis Services Enhancements	EE (32-bit) DE (32-bit) EE (64-bit) DE (64-bit)	SE (32-bit)	WG (32-bit)	SSE (32-bit) SSEA (32-bit)	SE (64-bit)
Data Warehouse Building Features					
Create cubes without database	Yes	Yes	No	No	No
Auto-generate Staging and Data Warehouse Schema	Yes	Yes	No	No	No
Auto-generate DTS Packages for Updating Data Warehouse Data	Yes	Yes	No	No	No
Extended Dimensional Modeling Features					
Measure expressions	Yes	Yes	No	No	Yes
Scalability and Performance Features					
Proactive Caching	Yes	No	No	No	No
Auto Parallel Partition Processing	Yes	No	No	No	No
Partitioned Cubes	Yes	No	No	No	No
Distributed Partitioned Cubes	Yes	No	No	No	No

▣ SQL Server 2005 Reporting Services Features

The following abbreviations represent SQL Server 2005 editions:

- EE = SQL Server 2005 Enterprise Edition
- DE = SQL Server 2005 Developer Edition
- SE = SQL Server 2005 Standard Edition
- WG = SQL Server 2005 Workgroup Edition
- SSE = SQL Server 2005 Express Edition
- SSEA = SQL Server 2005 Express Edition with Advanced Services

Features/Reporting Services Enhancements	EE (32-bit) DE (32-bit) EE (64-bit) DE (64-bit)	SE (32-bit) SE (64-bit)	WG (32-bit)	SSE (32-bit)	SSEA (32-bit)
Support for remote and nonrelational data sources	Yes	Yes	No	No	No
DHTML, Excel, PDF, and Image rendering extensions	Yes	Yes	Yes	No	Yes
MHTML, CSV, XML, and Null rendering extensions	Yes	Yes	No	No	No
E-mail and file share delivery extensions	Yes	Yes	No	No	No
Custom data processing, delivery, and rendering extensions	Yes	Yes	No	No	No
Custom authentication extensions	Yes	Yes	Yes	No	No
Report caching	Yes	Yes	No	No	No
Report history	Yes	Yes	No	No	No
Scheduling	Yes	Yes	No	No	No
Subscriptions	Yes	Yes	No	No	No
Data-driven subscriptions	Yes	No	No	No	No
User-defined role definitions	Yes	Yes	No	No	No
Report model data sources	Yes	Yes	Yes	No	No
Report model item security	Yes	Yes	No	No	No

Support for infinite clickthrough in ad hoc reports	Yes	No	No	No	No
Report Builder	Yes	Yes	Yes	No	No
Report Manager	Yes	Yes	Yes	No	Yes
SQL Server Management Studio	Yes	Yes	Yes	No	No
Report Designer in Business Intelligence Development Studio	Yes	Yes	Yes	No	Yes ¹
Report server scale-out deployment	Yes	No	No	No	No

For information on additional Reporting Services features available in SQL Server 2005, see "Reporting Services Features" in SQL Server 2005 Books Online.

¹ Business Intelligence Development Studio is installed by using SQL Server 2005 Express Edition Toolkit. For more information, see **SQL Server Express Toolkit** in SQL Server 2005 Express Edition Books Online.

▣ SQL Server 2005 Notification Services Features

The following abbreviations represent SQL Server 2005 editions:

- EE = SQL Server 2005 Enterprise Edition
- DE = SQL Server 2005 Developer Edition
- SE = SQL Server 2005 Standard Edition
- WG = SQL Server 2005 Workgroup Edition
- SSE = SQL Server 2005 Express Edition
- SSEA = SQL Server 2005 Express Edition with Advanced Services

Features/Notification Services Enhancements	EE (32-bit) DE (32-bit) EE (64-bit) DE (64-bit)	SE (32-bit)	WG (32-bit)	SSE (32-bit) SSEA (32-bit)	SE (64-bit)
Notification Services SDK (API, NSControl, ICF/ADF)	Yes	Yes	No	No	Yes
Notification Services Engine and Client components	Yes	Yes	No	No	Yes
Scalability (Parallelism, Multicast, Distributed deployment)	Yes	No	No	No	No

▣ SQL Server 2005 Integration Services Features

The following abbreviations represent SQL Server 2005 editions:

- EE = SQL Server 2005 Enterprise Edition
- DE = SQL Server 2005 Developer Edition
- SE = SQL Server 2005 Standard Edition
- WG = SQL Server 2005 Workgroup Edition
- SSE = SQL Server 2005 Express Edition
- SSEA = SQL Server 2005 Express Edition with Advanced Services

Features/Integration Services Enhancements	EE (32-bit) (32-bit) EE (64-bit) DE (64-bit)	DE (64-bit)	SE (32-bit)	WG (32-bit)	SSE (32-bit) SSEA (32-bit)	SE (64-bit)
SQL Server Import and Export Wizard and supporting connections, source and destination adapters, and tasks	Yes		Yes	Yes	No	Yes
Execute SQL Task	Yes		Yes	Yes	No	Yes
OLE DB Source and Destination Adapters	Yes		Yes	Yes	No	Yes
SSIS command prompt tools	Yes		Yes	Yes	No	Yes
SSIS Package Designer	Yes		Yes	Yes	No	Yes
Legacy support for DTS packages	Yes		Yes	Yes	Yes	Yes
SSIS Service	Yes		Yes	No	No	Yes
All other source and destination adapters, tasks, and transformations, except for those listed below	Yes		Yes	No	No	Yes
Data Mining Query Transformation	Yes		No	No	No	No
Data Mining Model Training Destination Adapter	Yes		No	No	No	No
Fuzzy Grouping Transformation	Yes		No	No	No	No
Fuzzy Lookup Transformation	Yes		No	No	No	No
Term Extraction Transformation	Yes		No	No	No	No
Term Lookup Transformation	Yes		No	No	No	No
Slowly Changing Dimension						

Transformation and Wizard	Yes	Yes	No	No	No
Dimension Processing Destination Adapter	Yes	No	No	No	No
Partition Processing Destination Adapter	Yes	No	No	No	No

▣ SQL Server 2005 Replication Features

The following abbreviations represent SQL Server 2005 editions:

- EE = SQL Server 2005 Enterprise Edition
- DE = SQL Server 2005 Developer Edition
- SE = SQL Server 2005 Standard Edition
- WG = SQL Server 2005 Workgroup Edition
- SSE = SQL Server 2005 Express Edition
- SSEA = SQL Server 2005 Express Edition with Advanced Services

Features/Replication Enhancements	EE (32-bit)	DE (32-bit)	SE	WG	SSE (32-bit)	SE
	EE (64-bit)	DE (64-bit)	(32-bit)	(32-bit)	SSEA (32-bit)	(64-bit)
Merge replication	Yes	4	Yes	Yes ¹	Subscriber only ²	Yes
Transactional replication	Yes		Yes	Yes ¹	Subscriber only ³	Yes
Snapshot replication	Yes		Yes	Yes	Subscriber only	Yes
Non-SQL Server Subscribers	Yes		Yes	No	No	Yes
Oracle publishing	Yes		No	No	No	No
Peer-to-peer transactional replication	Yes		No	No	No	No

¹If an instance of WG is used as a Publisher, it supports a maximum of 25 subscriptions to all merge publications and 5 subscriptions to all transactional publications. It supports an unlimited number of subscriptions to snapshot publications.

²You cannot configure SSE as a Publisher or Distributor for any type of replication. However, SSE can be a Subscriber, and merge replication allows changes to be replicated in both directions between a Publisher and Subscriber.

³SSE does not support queued updating using pull subscriptions. It does support queued updating using push subscriptions and immediate updating using push or pull subscriptions (immediate updating requires that Microsoft Distributed Transaction Coordinator (MS DTC) be running on the same computer as the SSE instance).

⁴Replication supports 32-bit and 64-bit versions of SQL Server in the same topology. All replication features are supported on the 64-bit version with the following considerations:

- Push subscriptions from a 64-bit Distributor require the appropriate 64-bit OLEDB provider or ODBC driver at the Distributor; pull subscriptions from a 64-bit Subscriber require the appropriate 64-bit provider or driver at the Subscriber. For more information about push and pull subscriptions, see **Subscribing to Publications**.
- To use Windows Synchronization Manager on a 64-bit Subscriber that is running in Windows on Windows mode (WOW), you must launch the executable mobsync.exe directly from the \Windows\Syswow64 directory, rather than from the **Start** menu. Launching the executable directly ensures the correct version is used. For more information about WOW, see [Hardware and Software Requirements for Installing SQL Server 2005](#). For more information about Windows Synchronization Manager, see **Windows Synchronization Manager**.
- Transformable push subscriptions from a 64-bit Distributor are not supported, because the necessary 64-bit Data Transformation Services (DTS) components are not available. Pull subscriptions from a 32-bit installation of SQL Server 2005 should work with a 64-bit Publisher or Distributor. For more information about transformable subscriptions, see **Transforming Published Data**.

▣ Change History

Release	History
12 December 2006	Changed content: <ul style="list-style-type: none">• Removed footnote stating that the Microsoft OLE DB Provider for DB2 is not supported on Itanium architecture.
17 July 2006	Changed content: <ul style="list-style-type: none">• Documented support for Microsoft .NET stored procedures on all editions of SQL Server 2005
5 December 2005	Changed content: <ul style="list-style-type: none">• Documented support for the Dedicated Administrator Connection feature for SQL Server 2005 Express Edition.• Added support information for SQL Server 2005 Express Edition with Advanced Services.

▣ **See Also**

Other Resources

[Product Specifications for SQL Server 2005](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Configuring High Availability



[See Also](#)



[Preparing to Install SQL Server 2005](#) >

Updated: **5 December 2005**

This section introduces several Microsoft SQL Server 2005 high-availability solutions that improve the availability of servers or databases. A high-availability solution masks the effects of a hardware or software failure and maintains the availability of applications so that the perceived downtime for users is minimized.

SQL Server 2005 provides several options for creating high availability for a server or database. High-availability options include the following:

- **Failover clustering**
Failover clustering provides high-availability support for an entire instance of SQL Server. A failover cluster is a combination of one or more nodes, or servers, with two or more shared disks. Applications such as SQL Server and Notification Services are each installed into a Microsoft Cluster Service (MSCS) cluster group, known as a resource group. At any time, each resource group is owned by only one node in the cluster. The application service has a virtual name that is independent of the node names, and is referred to as the failover cluster instance name. An application can connect to the failover cluster instance by referencing the failover cluster instance name. The application does not have to know which node hosts the failover cluster instance.
A SQL Server failover cluster instance appears on the network as a single computer, but has functionality that provides failover from one node to another if the current node becomes unavailable. For example, during a non-disk hardware failure, operating system failure, or planned operating system upgrade, you can configure an instance of SQL Server on one node of a failover cluster to fail over to any other node in the disk group. A failover cluster does not protect against disk failure. You can use failover clustering to reduce system downtime and provide higher

application availability. Failover clustering is supported in SQL Server 2005 Enterprise Edition, Developer Edition and, with some restrictions, Standard Edition. For more information about failover clustering, see [Failover Clustering](#) and [Installing a Failover Cluster](#).

- Database mirroring

Database mirroring is primarily a software solution to increase database availability by supporting almost instantaneous failover. Database mirroring can be used to maintain a single standby database, or *mirror database*, for a corresponding production database that is referred to as the *principal database*.

The mirror database is created by restoring a database backup of the principal database with no recovery. This makes the mirror database is inaccessible to clients. However, it is possible to use it indirectly for reporting purposes by creating a database snapshot on the mirror database. The database snapshot provides clients with read-only access to the data in the database as it existed when the snapshot was created. Each database mirroring configuration involves a *principal server* that contains the principal database, and a mirror server that contains the mirror database. The mirror server continuously brings the mirror database up to date with the principal database.

Database mirroring runs with either synchronous operation in high-safety mode, or asynchronous operation in high-performance mode. In high-performance mode, the transactions commit without waiting for the mirror server to write the log to disk, which maximizes performance. In high-safety mode, a committed transaction is committed on both partners, but at the risk of increased transaction latency.

In its simplest configuration, database mirroring involves only the principal and mirror servers. In this configuration, if the principal server is lost, the mirror server can be used as a warm standby server, with possible data loss. High-safety mode supports an alternative configuration, high-safety mode with automatic failover. This configuration involves a third server instance, known as a *witness*, which enables the mirror server to act as a hot standby server. Failover from the principal database to the mirror database typically takes a few seconds. Database mirroring is fully supported in SQL Server Standard Edition and Enterprise Edition, but the failover partners must use the same edition. Server instances that are running on SQL Server Workgroup Edition or Express Edition support the witness role only. For more information about database mirroring, see **Database Mirroring**.

- Log shipping

Like database mirroring, log shipping operates at the database level. Log shipping can be used to maintain one or more warm standby databases, referred to as *secondary databases*, for a corresponding production database that is referred to as the *primary database*. Each secondary database is created by restoring a database backup of the primary database with no recovery, or with standby. Restoring with standby permits the resulting secondary database to be used for limited reporting purposes.

A log shipping configuration includes a single primary server that contains the primary database, one or more secondary servers that each have a secondary database, and a monitor server. Each secondary server updates its secondary database at regular intervals from log backups of the primary database. Log shipping involves a user-modifiable delay between when the primary server creates a log backup of the primary database and when the secondary server restores the log backup. Before a failover can occur, a secondary database must be brought fully up-to-date by manually applying any unrestored log backups.

Log shipping provides the flexibility of supporting multiple standby databases. If you require multiple standby databases, you can use log shipping alone or as a supplement to database mirroring. When these solutions are used together, the current principal database of the database mirroring configuration is also the current primary database of the log shipping configuration.

Log shipping is supported in SQL Server 2005 Enterprise Edition, Standard Edition, and Workgroup Edition. For more information about log shipping, see **Log Shipping**.

- Replication

Replication uses a publish-subscribe model, allowing a primary server, referred to as the Publisher, to distribute data to one or more secondary servers, or Subscribers. Replication allows real-time availability and scalability across these servers. It supports filtering to provide a subset of data at Subscribers, and also allows partitioned updates. Subscribers are online and available for reporting or other functions, without query recovery. SQL Server offers three types of replication: snapshot, transactional, and merge. Transactional replication provides the lowest latency and is most commonly used for high availability. For more information, see **Improving Scalability and Availability**.

Replication is supported in all editions of SQL Server 2005. Replication

publishing is not available with SQL Server 2005 Express Edition or SQL Server 2005 Compact Edition. For a complete list of replication features that are supported by each edition, see [Features Supported by the Editions of SQL Server 2005](#).

 **Important:**

A well designed and implemented backup and restore strategy is essential to any high-availability solution. For more information, see **Backing Up and Restoring Databases**, and **Backing Up and Restoring Replicated Databases**.

Selecting a High-Availability Solution

The following list presents considerations for selecting a high-availability solution:

- Failover clustering and database mirroring both provide the following:
 - Automatic detection and failover
 - Manual failover
 - Transparent client redirect

Failover clustering has the following constraints:

- Operates at the server instance scope
- Requires certified hardware
- Has no reporting on standby
- Utilizes a single copy of the database
- Does not protect against disk failure

Database mirroring offers the following benefits:

- Operates at the database scope.
- Uses a single, duplicate copy of the database

 **Note:**

If you require additional copies, you can use log shipping on the database in addition to database mirroring.

- Uses standard servers
- Provides limited reporting on the mirror server by using database

snapshots.

- When operating synchronously, provides for zero work loss through delayed commit on the principal database.

Database mirroring offers a substantive increase in availability over the level previously possible with SQL Server and offers an easy-to-manage alternative to failover clustering.

 **Note:**

For information about using database mirroring with a failover cluster, see **Database Mirroring and Failover Clustering**. For information about using Notification Services with failover clustering, see **Using Failover Clustering With Notification Services**. For information about using Notification Services with database mirroring, see **Using Log Shipping or Database Mirroring with Notification Services**.

- **Log shipping**

Log shipping can be a supplement or an alternative to database mirroring. Although similar in concept, asynchronous database mirroring and log shipping have key differences. Log shipping offers the following distinct capabilities:

- Supports multiple secondary databases on multiple server instances for a single primary database.
- Allows a user-specified delay between when the primary server backs up the log of the primary database and when the secondary servers must restore the log backup. A longer delay can be useful, for example, if data is accidentally changed on the primary database. If the accidental change is noticed quickly, a delay can let you retrieve still unchanged data from a secondary database before the change is reflected there.

Compared to the shortest time that is required by log shipping to reflect a change to a secondary database, asynchronous database mirroring has the potential advantage of a shorter time between when a given change is made in the primary database and when that change is reflected to the mirror database.

An advantage of database mirroring over log shipping is that high-safety mode is a no data loss configuration that is supported

as a simple failover strategy.

 **Note:**

For information about using log shipping with database mirroring, see **Log shipping and Database Mirroring Interaction**. For information about using log shipping with Notification Services, see **Using Log Shipping or Database Mirroring with Notification Services**.

- Replication

Replication offers the following benefits:

- Allows filtering within the database to provide a subset of data at the secondary databases because it operates at the database scope
- Allows more than one redundant copy of the database
- Allows real-time availability and scalability across multiple databases, supporting partitioned updates
- Allows complete availability of the secondary databases for reporting or other functions, without query recovery.

 **Note:**

For information about using database mirroring with replication, see **Replication and Database Mirroring**.

▣ In This Section

Topic	Description
Failover Clustering	Contains information about sharing a combination of one or more nodes (servers) with two or more hard disks.
Database Mirroring	Contains information about how database mirroring works and how to configure and manage a database mirroring session.
Log Shipping	Contains information about how log shipping works and how to configure and manage a log shipping configuration.

▣ **See Also**

Reference

[Hardware and Software Requirements for Installing SQL Server 2005](#)
[Features Supported by the Editions of SQL Server 2005](#)

Other Resources

Replication

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Failover Clustering



[See Also](#)



[Preparing to Install SQL Server 2005](#) > [Configuring High Availability](#) >

Failover clustering in Microsoft SQL Server 2005 provides high-availability support for an entire SQL Server instance. For example, you can configure a SQL Server instance on one node of a failover cluster to fail over to any other node in the cluster during a hardware failure, operating system failure, or a planned upgrade.

A failover cluster is a combination of one or more nodes (servers) with two or more shared disks, known as a resource group. The combination of a resource group, along with its network name, and an internet protocol (IP) address that makes up the clustered application or server, is referred to as a *failover cluster* or a *failover cluster instance*. A SQL Server failover cluster appears on the network as if it were a single computer, but has functionality that provides failover from one node to another if the current node becomes unavailable. A failover cluster appears on the network as a normal application or single computer, but it has additional functionality that increases its availability.

▣ **When to Use Failover Clustering**

Use failover clustering to:

- Administer a failover cluster from any node in the clustered SQL Server configuration. For more information, see [Installing a Failover Cluster](#).
- Allow one failover cluster node to fail over to any other node in the failover cluster configuration. For more information, see [Installing a Failover Cluster](#).
- Configure Analysis Services for failover clustering. For more information, see [How to: Install Analysis Services on a Failover Cluster](#).
- Execute full-text queries by using Microsoft Search service with failover clustering. For more information, see [Using SQL Server Tools with Failover Clustering](#).

▣ Installing Failover Clustering

To use failover clustering, you must follow specific installation steps.

To install, configure, and maintain a SQL Server failover cluster, use SQL Server Setup. You can also use Setup to upgrade an existing 32-bit instance of SQL Server 7.0 or Microsoft SQL Server 2000 to a SQL Server 2005 failover cluster. For more information, see [Version and Edition Upgrades](#).

Use SQL Server Setup in a failover cluster to:

- Install SQL Server on multiple nodes in a failover cluster. You are limited only by the number of nodes supported by the operating system. Before installing failover clustering, you must ensure that your system meets minimum requirements, and configure the Microsoft Cluster Service (MSCS). For more information on specific requirements for a failover cluster, see [Hardware and Software Requirements for Installing SQL Server 2005](#) and [Before Installing Failover Clustering](#).
All nodes in a failover cluster must be of the same platform, either 32-bit or 64-bit. Furthermore, 64-bit SQL Server editions must be installed on 64-bit hardware running the 64-bit versions of Microsoft Windows operating systems. For more information, see [Hardware and Software Requirements for Installing SQL Server 2005](#).
- To install a Microsoft SQL Server 2005 failover cluster, you must create and configure a failover cluster by running SQL Server Setup. For more information, see [Installing a Failover Cluster](#).
- Add or remove nodes from a failover cluster configuration without affecting the other cluster nodes. For more information, see [Maintaining a Failover Cluster](#).
- Specify multiple IP addresses for each failover cluster instance. With SQL Server 2005, you can specify multiple IP addresses for each server, allowing you to use all available network IP subnets. SQL Server 2005 limits you to one IP address per network. This provides alternative network connections if one subnet fails.

▣ **Failover Clustering Support**

In Microsoft SQL Server 2005, the number of nodes supported for failover clustering depends on the operating system. Some operating systems may require fiber optic channels for clustering. For more information, see [Hardware and Software Requirements for Installing SQL Server 2005](#).

For a list of tools, features, and services supported with SQL Server 2005 failover clustering, see [Features Supported by the Editions of SQL Server 2005](#).

▣ **See Also**

Tasks

[How to: Create a New SQL Server 2005 Failover Cluster \(Setup\)](#)

Concepts

[Before Installing Failover Clustering](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Before Installing Failover Clustering



[See Also](#)



[Preparing to Install SQL Server 2005](#) > [Configuring High Availability](#) > [Failover Clustering](#) >

Updated: **12 December 2006**

Before you install a Microsoft SQL Server 2005 failover cluster, you must select the hardware and the operating system on which SQL Server 2005 will run. You must also configure Microsoft Cluster Service (MSCS), and review network, security, and considerations for other software that will run on your failover cluster.

▣ Preinstallation Checklist

Before you begin the failover cluster installation process, review the items below.

Verify Your Hardware Solution

- Your hardware must be listed on the [Microsoft Windows Catalog and Hardware Compatibility List](#). The hardware system must appear under the category of a cluster solution.
 - ▣ **Important:**
Individual cluster components added together do not constitute an approved system for failover clustering. Only systems purchased as a cluster solution and listed in the cluster group are approved. When checking the Microsoft Windows Catalog and Windows Hardware Compatibility List, specify "cluster" as the category. All other categories are for OEM use. For more information, see [The Microsoft support policy for server clusters, the Hardware Compatibility List, and the Windows Server Catalog](#).
- Special hardware compatibility testing is necessary when implementing a failover server cluster on a Storage Area Network (SAN). The entire hardware solution must be in the Cluster/Multi-cluster Device category of the Microsoft Windows Catalog and Hardware Compatibility List. For more information, see the Microsoft Knowledge Base article, [Support for multiple clusters attached to the same SAN device](#).
- If the cluster solution includes geographically dispersed cluster nodes, additional items like network latency and shared disk support must be verified. The entire solution must be on the Geographic Cluster Hardware Compatibility List. For more information, see the Microsoft Knowledge Base article, [Windows clustering and geographically separate sites](#).
- SAN configurations are also supported on Microsoft Windows 2000 Advanced Server and Datacenter Editions. The Microsoft Windows Catalog and Hardware Compatibility List category "Cluster/Multi-cluster Device" lists the set of SAN-capable storage devices that have been tested and are supported as SAN storage units with multiple MSCS clusters attached. By

matching the devices on this list with the complete cluster configurations defined in the Microsoft Windows Catalog and Hardware Compatibility List cluster category, it is possible to deploy a set of Windows servers and clusters on a SAN fabric with shared storage devices in a way that is supported by Microsoft. For more information, see the Microsoft Knowledge base article, [The Datacenter Program and Windows 2000 Datacenter Server Product](#).

- If you deploy a SQL Server 2005 failover cluster on Internet small computer system interface (iSCSI) technology components, we recommend that you use appropriate caution. For more information, see the Microsoft knowledge base article, [Support for SQL Server 2000 on](#).
- For support information, see the Microsoft Knowledge base article, [Microsoft support policy for a SQL Server failover cluster](#).
- Consider quorum disk resource sharing. In a server cluster, the quorum disk contains a master copy of the server cluster configuration, and is also used as a tie-breaker if all network communication fails between cluster nodes. Depending on the type of server cluster you implement, the quorum disk may or may not be a physical disk on the shared cluster disk array. Although it is best to reserve an entire cluster disk for use as the quorum disk, resources other than the quorum resource may be permitted to access the quorum disk.

However, making the quorum resource share the same disk with other resources forces you to choose between two undesirable alternatives. Either you must configure the resource so that its failure does not affect the group, or you must allow the group to be affected by the other resource's failures. In the first case, you lose failover support for the resource; in the second, the quorum resource fails over along with the rest of the group that contains both the quorum resource and the failed resource. As a result, the entire cluster is offline for as long as it takes the group to fail over.

For more information about proper quorum drive configuration, see the Knowledge Base article, [Quorum Drive Configuration Information](#).

- To install a SQL Server 2005 failover cluster when the source installation files and the cluster exist on different domains, copy the installation files to the primary node of the cluster, then start the installation from the primary node.

Verify Your Operating System Settings

- Make sure that your operating system is installed properly and designed to support failover clustering. For more information on supported operating systems, see [Hardware and Software Requirements for Installing SQL Server 2005](#).
- Enable Windows Cryptographic Service Provider (CSP) on Windows Server 2003. If the CSP service is stopped or disabled on any cluster node, SQL Server Setup fails with a Windows Logo Requirement dialog.
- Enable the Task Scheduler service on all operating systems for remote and cluster installation. If the Task Scheduler is disabled, SQL Server Setup fails with Error 1058. For more information, see [How to: Enable Windows Task Scheduler Service](#).
- SQL Server 2005 supports mount points; the clustered installations of SQL Server are limited to the number of available drive letters. Assuming that you use only one drive letter for the operating system, and all other drive letters are available as normal cluster drives or cluster drives hosting mount points, you are limited to a maximum of 25 instances of SQL Server per failover cluster.

A mounted volume, or mount point, allows you to use a single drive letter to refer to many disks or volumes. If you have a drive letter D: that refers to a regular disk or volume, you can connect or "mount" additional disks or volumes as directories under drive letter D: without the additional disks or volumes requiring drive letters of their own.

Special mount point considerations for SQL Server 2005 failover clustering:

- SQL Server Setup requires that the base drive of a mounted drive has an associated drive letter. For failover cluster installations, this base drive must be a clustered drive.
- The base drive, the one with the drive letter, cannot be shared among failover cluster instances. This is a normal restriction for failover clusters, but is not a restriction on stand-alone, multi-instance servers.
- Take extra care when setting up your failover cluster to ensure that both the base drive and the mounted disks or volumes are all listed as resources in the resource group. SQL Server Setup does not take care of this automatically, nor does SQL Server check for this during CREATE/ALTER DATABASE.
- Ensure that the mounted disks or volumes are mounted under the correct lettered base drive, and make sure that the mounted drive is set to be dependent on the parent drive.

Configure Microsoft Cluster Server

- Microsoft Cluster Server (MSCS) must be configured on at least one node of your server cluster. MSCS is only supported if it is installed on a hardware configuration that has been tested for compatibility with the MSCS software. You must also run SQL Server 2005 Enterprise Edition or Standard Edition in conjunction with MSCS. SQL Server 2005 Enterprise Edition supports failover clusters with up to 8 nodes. SQL Server 2005 Standard Edition supports 2-node failover clusters.
For more information about installing and configuring MSCS on Windows Server 2003, see [Server clusters](#).
For more information on installing and configuring MSCS on Windows 2000 operating systems, see [Step by Step Guide to Installing Cluster Service](#).
- MSCS must be able to verify that the failover cluster instance is running by using the **IsAlive** thread. This requires connecting to the server using a trusted connection. By default, the account that runs the cluster service is configured as an administrator on all nodes in the cluster, and the BUILTIN\Administrators group has permission to log into SQL Server. These settings change only if you change permissions on the cluster nodes. If the BUILTIN\Administrators account is removed, ensure that the account that the Cluster Service is running under can log into SQL Server for the **IsAlive** check. If it cannot, the **IsAlive** check will fail. At a minimum, the MSCS Cluster Service account must have **public** rights to SQL Server so that it can run **@@servername** on a regular basis.
- When you install MSCS, it is very important to use separate service accounts to log on to MSCS and SQL Server. Otherwise, the cluster service password cannot be changed using the cluster command.
- When using MSCS, one node must be in control of the shared SCSI bus prior to the other node coming online. Failure to do this can cause application failover to go into an online pending state and either prevent failover to the other node, or totally fail. If your cluster system has a proprietary install process, the proprietary process should be used.

Install Microsoft Distributed Transaction Coordinator

- Before installing SQL Server 2005 on a failover cluster, determine whether the Microsoft Distributed Transaction Coordinator (MSDTC) cluster

resource must be created. If you are installing only the Database Engine, the MSDTC cluster resource is not required. If you are installing the Database Engine and SSIS, Notification Services, or Workstation Components, you must install MSDTC. This requirement applies to both Windows 2000 and Windows Server 2003 operating systems.

The MSDTC transaction manager, MSDTC proxy, and Component Services administrative tools are installed on each node of the Windows-based server cluster. The cluster uses Microsoft Cluster Services (MSCS) as part of the setup of the Windows-based server cluster.

To manually configure MSDTC on a Windows Server 2003 operating system (running MSCS), see the Knowledge Base articles [How to configure Microsoft Distributed Transaction Coordinator on a Windows 2003 cluster](#) and [How to enable network DTC access in Windows Server 2003](#).

To help ensure availability between multiple clustered applications, Microsoft highly recommends that the MSDTC have its own resource group and resources. If MSDTC cannot be configured to have its own resource group, the recommended alternate choice is to use the Cluster group and Quorum drive.

Configure Microsoft Distributed Transaction Coordinator

- After you install the operating system and configure your cluster, you must configure MSDTC to work in a cluster by using the Cluster Administrator. Failure to cluster MSDTC will not block SQL Server Setup, but SQL Server application functionality may be affected if MSDTC is not properly configured.

Any process running on any node in the cluster can use MSDTC. These processes simply call the MSDTC proxy, and the MSDTC proxy automatically forwards MSDTC calls to the MSDTC transaction manager, which controls the entire cluster.

If the node running the MSDTC transaction manager fails, the transaction manager is automatically restarted on another node in the cluster. The newly restarted transaction manager reads the MSDTC log file on the shared cluster disk to determine the outcome of pending and recently completed transactions.

Resource managers reconnect to the transaction manager and perform recovery to determine the outcome of pending transactions. Applications reconnect to MSDTC so they can initiate new transactions.

For example, suppose that the MSDTC transaction manager is active on system B. The application program and resource manager on system A call the MSDTC proxy. The MSDTC proxy on system A forwards all MSDTC calls to the MSDTC transaction manager on system B.

If system B fails, the MSDTC transaction manager on system A takes over. It reads the entire MSDTC log file on the shared cluster disk, performs recovery, and then serves as the transaction manager for the entire cluster.

Other Software Considerations

- Ensure that all cluster nodes are configured identically, including COM+, disk drive letters, and users in the administrators group.
- Verify that the cluster interconnect (heartbeat) is properly configured. For more information, see the Knowledge Base article, [Recommended private "Heartbeat" configuration on a cluster server](#).
- Verify that you have cleared the system logs in all nodes and viewed the system logs again. Ensure that the logs are free of any error messages before continuing.
- For SQL Server 2005 installations in side-by-side configurations with previous versions of SQL Server, SQL Server 2005 services must use accounts found only in the global domains group. Additionally, accounts utilized by SQL Server 2005 services must not appear in the local Administrators group. Failure to comply with this guideline will result in unexpected security behavior.
- If you install SQL Server 2005 into a Windows 2000 cluster group with multiple disk drives and choose to place your data on one of the drives, the SQL Server resource is set to be dependent only on that drive. To put data or logs on another or additional disk resources, you must first add a dependency to the SQL Server resource for the additional disk. For more information, see [How to: Add Dependencies to a SQL Server 2005 Resource](#).
- If you are deploying Windows 2000 or Windows Server 2003 cluster nodes in an environment where there are no pre-existing Microsoft Windows 2000, or Windows Server 2003 domain controllers, see [Windows 2000 and Windows Server 2003 cluster nodes as domain controllers](#).
- To use encryption, install the server certificate with the fully qualified DNS name of the MSCS cluster on all nodes in the SQL Server failover cluster. For example, if you have a two-node cluster, with nodes named

"Test1.DomainName.com" and "Test2.DomainName.com" and a SQL Server failover cluster instance named "Virtsql", you must get a certificate for "Virtsql.DomainName.com" and install the certificate on the **test1** and **test2** nodes. Then you can select the **Force protocol encryption** check box on the SQL Server Configuration Manager to configure your failover cluster for encryption.

 **Important:**

Do not select the **Force protocol encryption** check box until you have certificates installed on all participating nodes in your failover cluster instance.

- Verify that anti-virus software is not installed on your MSCS cluster. For more information, see the Knowledge Base article, [Antivirus software may cause problems with cluster services](#).
- SQL Server 2005 is not supported on Windows Server 2003 Terminal Server. For more information, see the Knowledge Base article, [SQL Server 2000 is not supported on Windows Server 2003 Terminal Server application server](#).
- Verify that the disk where SQL Server will be installed is uncompressed. If you attempt to install SQL Server to a compressed drive, SQL Server Setup fails.
- When naming a cluster group for your failover cluster installation, you must not use any of the following characters in the cluster group name:
 - Less than operator (<)
 - Greater than operator (>)
 - Double quote (")
 - Single quote (')

Network Considerations

- Verify that you have disabled NetBIOS for all private network cards before beginning SQL Server Setup.
- The network name and IP address of your SQL Server should not be used for any other purpose, such as file sharing. If you want to create a file share resource, use a different, unique network name and IP address for the resource.

 **Important:**

Microsoft recommends that you do not use file shares on data drives, as they can affect SQL Server behavior and performance.

- Even though SQL Server 2005 supports both named Pipes and TCP/IP Sockets over TCP/IP within a cluster, Microsoft recommends that you use TCP/IP Sockets in a clustered configuration.

Other Considerations

- To create a failover cluster, you must be a local administrator with permissions to logon as a service, and to act as part of the operating system on all nodes of the failover cluster instance.
- Before you install or update a SQL Server failover cluster, disable all applications and services that might use SQL Server components during installation, but leave the disk resources online.
- Create domain groups for the clustered services that will be installed as part of your SQL Server 2005 failover cluster. SQL Server service, SQL Server Agent service, Analysis Services service, and Full-Text Search service must run as domain accounts that are members of the global or local domain group. If necessary, ask your domain administrator for the names of existing domain groups, or to create domain groups for your failover cluster. For more information, see [Domain Groups for Clustered Services](#).
- SQL Server failover clustering is not supported where cluster nodes are domain controllers.
- Configure Domain Name Service (DNS) or Windows Internet Name Service (WINS). A DNS server or WINS server must be running in the environment where your SQL Server failover cluster will be installed. SQL Server Setup requires dynamic domain name service (DDNS) registration of the SQL Server IP interface virtual reference. If the dynamic registration cannot be completed, Setup fails and the installation is rolled back. If no dynamic registration is available, you must have pre-registered your server in DNS.
- Review content in [Security Considerations for a SQL Server Installation](#).
- Review content in [Check Parameters for the System Configuration Checker](#).
- Consider whether the SQL Server tools, features, and components you want to use are supported with failover clustering. For more information, see [Failover Clustering](#).

- Consider how you will monitor and maintain your failover cluster to achieve your high availability goals. For more information, see [Maintaining a Failover Cluster](#) and [Using SQL Server Tools with Failover Clustering](#).
- To reduce the time required to install a SQL Server 2005 failover cluster, you can pre-install Microsoft .NET Framework version 2.0 on all failover cluster nodes before running SQL Server Setup.

▣ Change History

Release	History
12 December 2006	<p>Changed content:</p> <ul style="list-style-type: none">• Added recommendation to take SQL Server offline before installing or updating a failover cluster instance.• Added security guidelines for service accounts in side-by-side configurations.• Updated domain groups section to specify that domain accounts must be members of the global or local domain group.• Updated recommendations for installing MSDTC.

▣ **See Also**

Reference

[Hardware and Software Requirements for Installing SQL Server 2005](#)
[Check Parameters for the System Configuration Checker](#)

Concepts

[Installing a Failover Cluster](#)
[Security Considerations for a SQL Server Installation](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Installing a Failover Cluster



[See Also](#)



[Preparing to Install SQL Server 2005](#) > [Configuring High Availability](#) > [Failover Clustering](#) >

To install a Microsoft SQL Server 2005 failover cluster, you must create and configure a failover cluster instance by running SQL Server Setup. This topic explains the elements of a failover cluster, as well as important information about naming a failover cluster instance. It also describes basic steps for installing a failover cluster.

▣ **Elements of a Failover Cluster Instance**

A failover cluster instance can run on one or more computers that are participating nodes of a failover cluster. The number of participating nodes is limited only by the operating system.

A failover cluster instance contains:

- A combination of one or more disks in a Microsoft Cluster Service (MSCS) cluster group, also known as a resource group. Each resource group can contain at most one instance of SQL Server.
- A network name for the failover cluster instance.
- One or more IP addresses assigned to the failover cluster instance.
- One instance of SQL Server 2005 that includes SQL Server, SQL Server Agent, and the Full-text Search (FTS) service.

▣ **Naming a Failover Cluster Instance**

A SQL Server failover cluster instance always appears on the network as if it were a single computer. You must use the SQL Server failover cluster instance name to connect to the SQL Server failover cluster, not the machine name of the node it happens to be running on. Doing so ensures that you are always able to connect to the failover cluster instance using the same name, regardless of which node is running SQL Server.

The name of your failover cluster instance must be unique to your domain. SQL Server does not listen on the IP address of the local servers. Instead, SQL Server listens only on the virtual IP address created during installation of the SQL Server failover cluster instance.

SQL Server depends on distinct registry keys and service names within the failover cluster to ensure that SQL Server functionality continues after a failover. Therefore, the name you provide for the instance of SQL Server, including the default instance, must be unique across all nodes in the failover cluster. Using unique instance names ensures that instances of SQL Server that are configured to fail over to a single server have distinct registry keys and service names.

▣ **Considerations for SQL Server Consolidation with a Failover Cluster**

- When planning to consolidate multiple stand-alone servers to SQL Server failover cluster instances, we recommend, as part of the planning process, that you verify that the cluster node hardware configuration is sufficient to support the number of instances of SQL Server to be hosted.

Presented below are two common scenarios and the basis for the recommended solutions that can be applied to your site configuration.

Scenario 1

You are a hosting site and want to offer up to 23 instances of SQL Server with the following resource requirements:

- 2 processors for 23 instances of SQL Server as a single cluster node would require 46 CPUs.
- 2 GB of memory for 23 instances of SQL Server as a single cluster node would require 48 GB of RAM (2 GB of additional memory for the operating system).
- 4 disks for 23 instances of SQL Server as a cluster disk array would require 92 disks.

If you limit the failover members to only half of the available nodes, the hardware requirements would be reduced to:

- 23 processors, 24 GB of memory per node, and availability of 46 disks.

With support for mount points, the total number of disks could be further reduced. SQL Server would require 23 disks each hosting 3-mount points or more, for increased disk space availability.

A potential limitation is if the processor and memory requirements rise, the existing hardware might not be capable of supporting those requirements without loss of performance to the existing instances of SQL Server.

Scenario 2

Migration of existing instances of SQL Server to a failover cluster

First, you must collect current baselines for the existing servers, noting any existing bottlenecks. Assuming baseline performance statistics provide the following needs, and that a single drive with mount points would cover drive requirements, that would leave you with the following configuration:

- SQL Server 1 - 8 processors, 16 GB of RAM
- SQL Server 2 - 4 processors, 8 GB of RAM (needs RAM, plan for 12

GB)

- SQL Server 3 – 2 processors, 16GB of RAM (baseline shows need for additional processors)
- SQL Server 4 – 4 processors, 8 GB of RAM (needs additional network bandwidth, add network adapters for 4 dedicated connections)

To replace these four servers, the cluster node would require the following minimum configuration:

- 18 processors, 54 GB of RAM, and 6 network adapters

The process for calculating the minimum configuration for a single node in a failover cluster considers:

- Total CPU requirements
- Total memory requirements, and
- Required disks
- Add 2 GB of RAM for the operating system

Note that the disk constraint encountered in previous SQL Server versions does not affect SQL Server 2005. With support for mounted drives in SQL Server 2005, each instance of SQL Server requires just one cluster disk for data files.

▣ Installing a Failover Cluster

To install a failover cluster, you must be a local administrator with permission to log on as a service and to act as part of the operating system on all nodes in the failover cluster.

To install a failover cluster by using the SQL Server Setup program, follow these steps:

1. Identify the information you need to create your failover cluster instance (for example, cluster disk resource, IP addresses, and network name) and the nodes available for failover. For more information:

- [Before Installing Failover Clustering](#)
- [Security Considerations for a SQL Server Installation](#)

These configuration steps must take place before you run the SQL Server Setup program; use the Windows Cluster Administrator to carry them out. You must have one MSCS group for each failover cluster instance you want to configure.

2. Start the SQL Server Setup program to begin your failover cluster installation. After you enter all necessary information, the Setup program installs a new instance of SQL Server binaries on the local disk of each computer in the failover cluster and installs the system databases on the specified cluster disk. The binaries are installed in exactly the same path on each cluster node, so you must ensure that each node has a local drive letter in common with all the other nodes in the cluster. A new SQL Server failover cluster instance with all failover cluster resources is created when SQL Server Setup completes the installation.

After installation completes, if any resource (including SQL Server fails for any reason, resources are either restarted on the same node, or the group moves to an available failover cluster node, depending on the failover cluster threshold configuration. The available failover cluster node is one of the nodes defined as available nodes for the failover cluster instance during SQL Server installation.

 **Note:**

You can also create a failover cluster instance using the .ini file provided on the SQL Server Setup CD. Installing a failover cluster from the command prompt is for advanced users only. For more information about

installing using the .ini file, see [Running Setup from the Command Prompt](#).

▣ **See Also**

Reference

[Hardware and Software Requirements for Installing SQL Server 2005](#)

Concepts

[Before Installing Failover Clustering](#)

Other Resources

[Upgrading to SQL Server 2005](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Failover Clustering Example



[See Also](#)



[Configuring High Availability](#) > [Failover Clustering](#) > [Installing a Failover Cluster](#) >

The following example illustrates how to configure Microsoft SQL Server 2005 failover clustering.

▣ **Prerequisites**

Before continuing, read [Before Installing Failover Clustering](#).

▣ **Example**

CLUSTERNODEA and CLUSTERNODEB are two computers in a failover cluster.

Run SQL Server Setup on the cluster node in control of the cluster disk where SQL Server data files will be located, and create a default instance of SQL Server on a failover cluster instance named **SQLCLUSTA**. This failover cluster instance can run on either CLUSTERNODEA or CLUSTERNODEB.

From this point forward, connect to the server by specifying **SQLCLUSTA** as the server name in the connection string.

Run Setup again on the cluster node in control of the cluster disk where SQL Server data files will be located.

▣ **Important:**

The cluster disk used by SQL Server must appear as a separate disk. You must not use the logical partition of a disk used for a previous failover cluster installation.

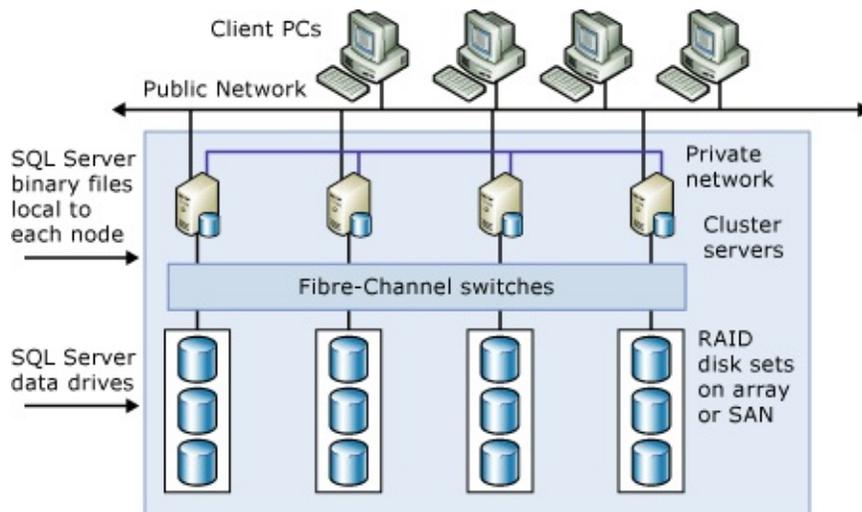
Create a named instance, **Inst1**, on a new MSCS server named **SQLCLUSTB**. This must be installed in a different Microsoft Cluster Service (MSCS) cluster resource group. This failover cluster instance can run on either CLUSTERNODEA or CLUSTERNODEB, as defined by the nodes you added to the cluster definition during SQL Server Setup.

From this point forward, connect to the server by specifying **SQLCLUSTB\Inst1** as the connection string.

The two failover cluster instances are running in the MSCS cluster consisting of CLUSTERNODEA and CLUSTERNODEB. Other than that, they are completely separate from each other. Each failover cluster instance resides in a different MSCS resource group, and each has a unique set of IP addresses, a distinct network name, and data files that reside on a separate set of shared cluster disks.

When a failover occurs for any resource in an MSCS resource group, all resources that are members of that group also fail over. For SQLCLUSTA, any failure (from the disk resources, IP address, the network name, or the installations of SQL Server within the failover cluster instance) causes all members of the cluster group to fail over when the failover threshold is reached.

The following illustration depicts a two-node cluster with binaries and data. Each failover cluster in this illustration must have exclusive ownership of the disk on which the data and log files are located.



▣ **See Also**

Concepts

[Before Installing Failover Clustering](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Upgrading to a SQL Server 2005 Failover Cluster



[See Also](#)



[Configuring High Availability](#) > [Failover Clustering](#) > [Installing a Failover Cluster](#) >

SQL Server 2005 supports upgrade of the Database Engine and Analysis Services. For more information on upgrade support, see [Upgrading to SQL Server 2005](#) and [Version and Edition Upgrades](#).

▣ **See Also**

Tasks

[How to: Upgrade to a SQL Server 2005 Failover Cluster \(Setup\)](#)

Reference

[Version and Edition Upgrades](#)

Concepts

[Preparing to Install SQL Server 2005](#)

Other Resources

[Upgrading to SQL Server 2005](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Maintaining a Failover Cluster



[See Also](#)



[Preparing to Install SQL Server 2005](#) > [Configuring High Availability](#) > [Failover Clustering](#) >

After you have installed a Microsoft SQL Server 2005 failover cluster, you can change or repair it using the SQL Server Setup program. For example, you can add additional nodes to a failover cluster instance, run a clustered instance as a stand-alone instance, or remove a node from a clustered instance configuration.

Note:

Cluster maintenance in this release of SQL Server is supported only through **Add or Remove Programs** in Control Panel.

▣ **Adding a Node to an Existing Failover Cluster Instance**

SQL Server Setup gives you the option of maintaining an existing failover cluster instance. If you choose this option, you can add other nodes to your failover cluster configuration. You can add as many nodes as the Microsoft Windows operating system will support.

For more information, see [How to: Create a New SQL Server 2005 Failover Cluster \(Setup\)](#).

▣ **Removing a Node from an Existing Failover Cluster Instance**

You can remove a node from a failover cluster configuration. Each node in a SQL Server failover cluster configuration is considered a peer without dependencies on other nodes on the cluster, and you can remove any node. A damaged node does not have to be available to be removed, and the removal process does not uninstall the SQL Server binaries from the unavailable node. A removed node can be added back to a failover cluster at any time. For more information, see [How to: Create a New SQL Server 2005 Failover Cluster \(Setup\)](#).

▣ **Running a Clustered Instance of SQL Server as a Stand-Alone Instance**

Usually, a clustered instance of SQL Server runs under the control of Microsoft Cluster Service (MSCS). However, it may be necessary to run a clustered instance of SQL Server as a stand-alone instance. For example, to run a SQL Server instance in single-user mode, it may be necessary to run a clustered instance of SQL Server as a stand-alone instance from a command prompt instead of starting SQL Server through MSCS. To connect to a clustered instance of SQL Server in stand-alone mode using sockets, both the IP address and network name resources must be online for the failover cluster on which the instance was installed.

If these resources cannot be online, connect using Named Pipes. You must create an alias on the client side to communicate with the pipe name on which the instance of SQL Server is listening. Use SQL Server SQL Server Configuration Manager to find out the pipe name.

▣ **Changing Service Accounts**

You should not change passwords for any of the SQL Server service accounts when a failover cluster node is down or offline. If you must do this, you must reset the password again using SQL Server Configuration Manager when all nodes are back online.

If the service account for SQL Server is not an administrator in your cluster, the administrative shares cannot be deleted on any nodes of the cluster. The administrative shares must be available in a cluster for SQL Server to function.

▣ **Important:**

Do not use the same account for the SQL Server service account and the MSCS account. If the password changes for the MSCS account, your SQL Server installation will fail.

▣ **See Also**

Tasks

[How to: Recover from Failover Cluster Failure in Scenario 1](#)

[How to: Recover from Failover Cluster Failure in Scenario 2](#)

[How to: Remove a SQL Server 2005 Failover Clustered Instance \(Setup\)](#)

Concepts

[Before Installing Failover Clustering](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Optimizing Failover Cluster Performance



[See Also](#)



[Configuring High Availability](#) > [Failover Clustering](#) > [Maintaining a Failover Cluster](#) >

To optimize performance when using failover clustering, consider the following issues.

▣ **Write Caching**

If your disk controller is not external to your clustered computer, you must turn off write caching within the controller to prevent data loss during a failover.

Write-back caching cannot be used on host controllers in a cluster without hindering performance. However, if you use external controllers, you continue to provide performance benefits. External disk arrays are not affected by failover clustering and can synchronize the cache correctly, even across a SCSI bus.

▣ **File Shares**

Do not use the same drive for file shares and clustered SQL Server databases. Using this drive impacts recovery time and can cause a failover of the cluster group in cases of a resource failure.

▣ **Maximum Memory Size**

Failover clustering performance benefits from generous memory allocation. The theoretical maximum memory is the total memory available, less memory for the operating system and the other cluster and local-to-node installed resources, divided by the instances of Microsoft SQL Server installed. Determine and set your maximum memory size using this formula. We recommend that you set your working maximum memory to 80 to 90 percent of this amount, and then gather performance baseline statistics to further optimize the setting.

▣ **Minimum Memory Size**

Minimum memory should be set where multiple SQL Server instances exist and maximum memory settings are in place. Setting the minimum memory size reduces failover time.

▣ **Cluster CPU Affinity**

CPU affinity default settings are preferred in most situations. However, if your installation is exceeding the maximum recommended performance thresholds, set cluster CPU affinity using an N-1 scenario.

▣ **Fixed IP Ports and Fixed Memory Settings**

Use fixed IP ports and fixed memory settings.

▣ **See Also**

Other Resources

**sp_configure (Transact-SQL)
Setting Server Configuration Options**

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Using SQL Server Tools with Failover Clustering



[See Also](#)



[Configuring High Availability](#) > [Failover Clustering](#) > [Maintaining a Failover Cluster](#) >

Updated: **5 December 2005**

You can use most Microsoft SQL Server tools and services with failover clustering. Before doing so, however, you should review the following considerations.

▣ Full-Text Queries

To use the Microsoft Search service to perform full-text queries with failover clustering, observe the following requirements:

- In order for full-text queries to function properly on failover clusters, an instance of Microsoft SQL Server 2005 must run on the same system account on all failover cluster nodes.

▣ **Note:**

This is a general requirement of failover clustering, and is not specific to Microsoft Search service.

- To change the startup account for SQL Server 2005 in the failover cluster, you must use SQL Server Configuration Manager. If you use Control Panel or the Services Application in Microsoft Windows to change the startup account, you will break the full-text configuration for SQL Server.

▣ **SQL Server Configuration Manager**

To use SQL Server Configuration Manager with failover clustering, observe the following requirements:

- To change the startup account for SQL Server 2005 in the failover cluster, you must use SQL Server Configuration Manager. If you use Control Panel or the Services Application in Windows, you might break your server configuration.
- When creating or altering databases, you can only view the cluster disks for the local failover cluster instance.
- If you use SQL Server Configuration Manager to reset the properties of the SQL Server service account, you are prompted to restart SQL Server. In a failover cluster configuration, this takes the full-text and SQL Server Agent resources offline; however, it does not bring the full-text or SQL Server Agent resources back online when SQL Server is restarted. You must restart those resources manually using the Windows Cluster Administrator utility.

▣ **SQL Server Management Studio**

- If you are browsing a table using SQL Server Management Studio and lose the connection to SQL Server during a failover, you see the error message, "Communication Link Failure." You must press the ESC key to undo the changes and exit out of the SQL Server Management Studio window. You cannot run a query, save any changes, or edit the grid.

▣ **Service Control Manager**

Use Service Control Manager to start or stop a clustered instance of SQL Server.
You cannot pause a clustered instance of SQL Server.

▣ **SQL Server Profiler**

You can use SQL Server Profiler with failover clustering. However, if a failover occurs when you are running a SQL Server Profiler trace, you must restart the trace after the server comes back online.

▣ **SQL Mail**

SQL Mail in SQL Server 2005 can be configured remotely by Configuration Manager, if your client remotely connects using Microsoft SQL Server 2000 Service Pack 3.0 or later.

▣ **See Also**

Reference

[Features Supported by the Editions of SQL Server 2005](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Failover Cluster Troubleshooting



[See Also](#)



[Configuring High Availability](#) > [Failover Clustering](#) > [Maintaining a Failover Cluster](#) >

Updated: **17 July 2006**

This topic provides information about the following issues:

- Basic troubleshooting steps.
- Recovering from a failover cluster failure.
- Resolving the most common failover clustering problems.
- Using extended stored procedures and COM objects.

▣ **Basic Troubleshooting Steps**

When working with SQL Server 2005 failover clustering, remember that the server cluster consists of a failover cluster instance that runs under Microsoft Cluster Services (MSCS). The SQL Server instance may be hosted by Microsoft MSCS-based nodes that provide the Microsoft Server Cluster.

If problems exist on the nodes that host the server cluster, those problems may manifest themselves as issues with your failover cluster instance. To investigate and resolve these issues, troubleshoot a SQL Server failover cluster in the following order:

1. **Hardware:** Review Microsoft Windows system event logs.
2. **Operating system:** Review Windows system and application event logs.
3. **Network:** Review Windows system and application event logs. Verify the current configuration against the Knowledge Base article, [Recommend](#).
4. **Security:** Review Windows application and security event logs.
5. **MSCS:** Review Windows system, application event, and cluster logs.
6. **SQL Server:** Troubleshoot as normal after the hardware, operating system, network, security, and MSCS foundations are verified to be problem-free.

▣ **Recovering from Failover Cluster Failure**

Usually, failover cluster failure is to the result of one of two causes:

- Hardware failure in one node of a two-node cluster. This hardware failure could be caused by a failure in the SCSI card or in the operating system. To recover from this failure, remove the failed node from the failover cluster using the SQL Server Setup program, address the hardware failure with the computer offline, bring the machine back up, and then add the repaired node back to the failover cluster instance.
For more information, see [How to: Create a New SQL Server 2005 Failover Cluster \(Setup\)](#) and [How to: Recover from Failover Cluster Failure in Scenario 1](#).
- Operating system failure. In this case, the node is offline, but is not irretrievably broken.
To recover from an operating system failure, recover the node and test failover. If the SQL Server instance does not fail over properly, you must use the SQL Server Setup program to remove SQL Server from the failover cluster, make necessary repairs, bring the computer back up, and then add the repaired node back to the failover cluster instance.
Recovering from operating system failure this way can take time. If the operating system failure can be recovered easily, avoid using this technique.
For more information, see [How to: Create a New SQL Server 2005 Failover Cluster \(Setup\)](#) and [How to: Recover from Failover Cluster Failure in Scenario 2](#).

Also consider the following changes to failover clustering in SQL Server 2005:

- In SQL Server 2005, log files are located in a different directory than in previous SQL Server releases. For more information, see [How to: View SQL Server 2005 Setup Log Files](#).
- In SQL Server 2005, instance-unaware components are installed on the primary node only; in SQL Server 2000, instance-unaware components were installed on all nodes. For more information, see [Setting Up Windows Service Accounts](#).
- In SQL Server 2005, directory and registry structures are different than in previous releases. For more information, see [Setting Up Windows Service Accounts](#) and [File Locations for Default and Named Instances of SQL](#)

[Server 2005.](#)

▣ Resolving Common Problems

The following list describes common usage issues and explains how to resolve them.

Problem: Incorrect use of command-prompt syntax to install SQL Server 2005

Issue 1: It is difficult to diagnose Setup issues when using the **/qn** switch from the command prompt, as the **/qn** switch suppresses all Setup dialog boxes and error messages. If the **/qn** switch is specified, all Setup messages, including error messages, are written to Setup log files. For more information about log files, see [How to: View SQL Server 2005 Setup Log Files](#).

Resolution 1: Use the **/qb** switch instead of the **/qn** switch. If you use the **/qb** switch, the basic UI in each step will be displayed, including error messages.

Issue 2: Not following the format used in the template.ini file, located in the root directory of SQL Server 2005 installation media.

Resolution 2: Though unexpected characters may be ignored by the Setup program, include all required variables in your installation command.

There should be no space between a variable and its value in a command line. For example, there is no space in "ADDLOCAL=ALL". If "ADDLOCAL = ALL" is used, Setup fails. Another example is the IP and its value. "IP=www.xxx.yyy.zzz,Local Area Connection" is correct. If a space is inserted after ",", Setup fails.

Problem: SQL Server 2005 cannot log on to the network after it migrates to another node

Issue 1: SQL Server service accounts are unable to contact a domain controller.

Resolution 1: Check your event logs for signs of networking issues such as adapter failures or DNS problems. Verify that you can ping your domain controller.

Issue 2: SQL Server service account passwords are not identical on all cluster nodes, or the node does not restart a SQL Server service that has migrated from a failed node.

Resolution 2: Change the SQL Server service account passwords using SQL Server Configuration Manager. If you do not, and you change the SQL Server service account passwords on one node, you must also change the passwords on all other nodes. SQL Server Configuration Manager does this automatically.

Problem: SQL Server cannot access the cluster disks

Issue 1: Firmware or drivers are not updated on all nodes.

Resolution 1: Verify that all nodes are using correct firmware versions and same driver versions.

Issue 2: A node cannot recover cluster disks that have migrated from a failed node on a shared cluster disk with a different drive letter.

Resolution 2: Disk drive letters for the cluster disks must be the same on both servers. If they are not, review your original installation of the operating system and Microsoft Cluster Service (MSCS).

Problem: Failure of a SQL Server service causes failover

Resolution: To prevent the failure of specific services from causing the SQL Server group to fail over, configure those services using Cluster Administrator in Windows, as follows:

- Clear the **Affect the Group** check box on the **Advanced** tab of the **Full Text Properties** dialog box. However, if SQL Server causes a failover, the full-text search service restarts.

Problem: SQL Server does not start automatically

Resolution: Use Cluster Administrator in MSCS to automatically start a failover cluster. The SQL Server service should be set to start manually; the Cluster Administrator should be configured in MSCS to start the SQL Server service.

Problem: The Network Name is offline and you cannot connect to SQL Server using TCP/IP

Issue 1: DNS is failing with cluster resource set to require DNS.

Resolution 1: Correct the DNS problems.

Issue 2: A duplicate name is on the network.

Resolution 2: Use NBTSTAT to find the duplicate name and then correct the issue.

Issue 3: SQL Server is not connecting using Named Pipes.

Resolution 3: To connect using Named Pipes, create an alias using the SQL Server Configuration Manager to connect to the appropriate computer. For example, if you have a cluster with two nodes (**Node A** and **Node B**), and a failover cluster instance (**Virtsql**) with a default instance, you can connect to the server that has the Network Name resource offline using the following steps:

1. Determine on which node the group containing the instance of SQL Server is running by using the Cluster Administrator. For this example, it is **Node A**.
2. Start the SQL Server service on that computer using **net start**. For more information about using **net start**, see **Starting SQL Server Manually**.
3. Start the SQL Server SQL Server Configuration Manager on **Node A**. View the pipe name on which the server is listening. It should be similar to `\\.\$\VIRTSQL\pipe\sql\query`.
4. On the client computer, start the SQL Server Configuration Manager.
5. Create an alias SQLTEST1 to connect through Named Pipes to this pipe name. To do this, enter **Node A** as the server name and edit the pipe name to be `\\.\pipe\$\VIRTSQL\sql\query`.
6. Connect to this instance using the alias SQLTEST1 as the server name.

Problem: SQL Server Setup fails on a cluster with error 1058

Issue: Disabling Task Scheduler Service on cluster nodes causes Setup to fail with Error 1058. The following entry is in core.log:

 Copy C

```
Error: SetTargetComputer on \\machinename failed with
Unable to start service (1058)
Error: RunRemoteProcess Received return code 1058 from STPCOMPAQ3790
```

This is the last line of the log:

 Copy Code

```
<EndFunc Name='DwLaunchMsiExec' Return='1058' GetLastError='183'>
```

Resolution: Use cluster manager to enable Task Scheduler Service on all cluster nodes. For more information, see [How to: Enable Windows Task Scheduler Service](#).

Problem: SQL Server Setup fails on a cluster with error 11001

Issue: An orphan registry key in [HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.X\Cluster]

Resolution: Make sure the MSSQL.X registry hive is not currently in use, and then delete the cluster key.

Problem: Cluster Setup Error: "The installer has insufficient privileges to access this directory: <drive>\Microsoft SQL Server. The installation cannot continue. Log on as an administrator or contact your system administrator"

Issue: This error is caused by a SCSI shared drive that is not partitioned properly.

Resolution: Re-create a single partition on the shared disk using the following steps:

1. Delete the disk resource from the cluster.
2. Delete all partitions on the disk.
3. Verify in the disk properties that the disk is a basic disk.
4. Create one partition on the shared disk, format the disk, and assign a drive letter to the disk.

5. Add the disk to the cluster using Cluster Administrator (cluadmin).
6. Run SQL Server Setup.

Problem: Applications fail to enlist SQL Server 2005 resources in a distributed transaction

Issue: Because the Microsoft Distributed Transaction Coordinator (MS DTC) is not completely configured in Windows, applications may fail to enlist SQL Server 2005 resources in a distributed transaction. This problem can affect linked servers, distributed queries, and remote stored procedures that use distributed transactions.

Resolution: To prevent such problems, you must fully enable MS DTC services on the server where SQL Server 2005 is installed.

To fully enable MS DTC, use the following steps:

1. In Control Panel, open **Administrative Tools**, and then open **Computer Management**.
2. In the left pane of Computer Management, expand **Services and Applications**, and then click **Services**.
3. In the right pane of Computer Management, right-click **Distributed Transaction Coordinator**, and select **Properties**.
4. In the **Distributed Transaction Coordinator** window, click the **General** tab, and then click **Stop** to stop the service.
5. In the **Distributed Transaction Coordinator** window, click the **Logon** tab, and set the logon account NT AUTHORITY\NetworkService.
6. Click **Apply** and **OK** to close the **Distributed Transaction Coordinator** window. Close the **Computer Management** window. Close the **Administrative Tools** window.

Note:

For installations of SQL Server 2005 on computers participating in a failover cluster, MS DTC must be fully enabled and clustered before you run Setup. If MS DTC is not clustered, Setup fails. Before running Setup, use the Microsoft Cluster Administrator to ensure that MS DTC has been clustered.

Problem: SQL Server 2005 failover cluster Setup might fail when installing from CD

Issue: When selecting a failover cluster component and a client tools component during SQL Server 2005 Setup, the installation might fail with a "Remote Setup Failed" error after you insert CD 2.

Resolution: Copy installation files from both CDs to the same directory on the active cluster node or network share. For example,

 [Copy Code](#)

```
c :  
cd\  
md SQLENT
```

Copy both CDs to the SQLENT directory:

```
\SQLENT\Servers
```

```
\SQLENT\Tools
```

 **Note:**

This is not an issue with DVD installations.

▣ **Using Extended Stored Procedures and COM Objects**

When you use extended stored procedures with a failover clustering configuration, all extended stored procedures must be installed on a SQL Server-dependent cluster disk. Doing so ensures that when a node fails over, the extended stored procedures can still be used.

If the extended stored procedures use COM components, the administrator must register the COM components on each node of the cluster. The information for loading and executing COM components must be in the registry of the active node in order for the components to be created. Otherwise, the information remains in the registry of the computer on which the COM components were first registered.

▣ Change History

Release	History
17 July 2006	Changed content: <ul style="list-style-type: none"><li data-bbox="532 499 1341 541">• Added section about failed Setup from CD media.

▣ **See Also**

Concepts

[How to: View SQL Server 2005 Setup Log Files](#)

Other Resources

How Extended Stored Procedures Work

Execution Characteristics of Extended Stored Procedures

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Configuring Availability How-to Topics



[See Also](#)



[Preparing to Install SQL Server 2005](#) > [Configuring High Availability](#) >

This section contains topics that describe how to set up and use failover clustering features in SQL Server 2005:

- [Failover Clustering How-to Topics](#)
- **Database Mirroring How-to Topics**
- **Log Shipping How-To Topics**

▣ **See Also**

Concepts

[Failover Clustering](#)

Other Resources

Database Mirroring

Log Shipping

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Failover Clustering How-to Topics



[See Also](#)



[Preparing to Install SQL Server 2005](#) > [Configuring High Availability](#) > [Configuring Availability How-to Topics](#) >

The how-to topics in this section contain information about creating or maintaining a failover cluster. Before you create a Microsoft SQL Server 2005 failover cluster, you must configure Microsoft Cluster Services (MSCS) by using the Cluster Administrator.

▣ In This Section

- [How to: Create a New SQL Server 2005 Failover Cluster \(Setup\)](#)
- [How to: Add Dependencies to a SQL Server 2005 Resource](#)
- [How to: Remove a SQL Server 2005 Failover Clustered Instance \(Setup\)](#)
- [How to: Recover from Failover Cluster Failure in Scenario 1](#)
- [How to: Recover from Failover Cluster Failure in Scenario 2](#)
- [How to: Change the IP Address of a SQL Server 2005 Failover Cluster](#)
- [How to: Enable Windows Task Scheduler Service](#)
- [How to: Install Microsoft Distributed Transaction Coordinator](#)

▣ **Related Topics**

- [Failover Clustering](#)
- [Before Installing Failover Clustering](#)

▣ **See Also**

Other Resources

[Configuring Availability How-to Topics](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Create a New SQL Server 2005 Failover Cluster (Setup)



[See Also](#)



[Configuring High Availability](#) > [Configuring Availability How-to Topics](#) > [Failover Clustering How-to Topics](#) >

Updated: **12 December 2006**

You can install Microsoft SQL Server 2005 or upgrade to SQL Server 2005 from a previous SQL Server version using the SQL Server 2005 Installation Wizard. The SQL Server 2005 Setup program is Microsoft Windows Installer-based, and provides a single feature tree for installation of all SQL Server 2005 components; You do not have to install each component individually.

For upgrade information, see [Version and Edition Upgrades](#) and [Upgrading to SQL Server 2005](#).

To install a stand-alone instance of SQL Server 2005, see [How to: Install SQL Server 2005 \(Setup\)](#). To upgrade a stand-alone or failover cluster instance, see [How to: Upgrade to a SQL Server 2005 Failover Cluster \(Setup\)](#).

▣ Prerequisites

Before you begin, review the following SQL Server Books Online topics:

- [Preparing to Install SQL Server 2005](#)
- [Before Installing Failover Clustering](#)
- It is very important that you also read and explicitly follow instructions in the [Recommended](#) Knowledge Base article.
- Before you create a Microsoft SQL Server failover cluster, you must configure Microsoft Cluster Service (MSCS) and use Cluster Administrator in Microsoft Windows 2000, or Windows Server 2003, Enterprise Server to create at least one cluster disk resource on a shared drive.
- For local installations, you must run SQL Server Setup as an administrator. If you install SQL Server from a remote share, you must use a domain account that has read and execute permissions on the remote share.
- We recommend that you run Setup from the active node.

☒ **Note:**

Take note of the location of the shared drive in the Cluster Administrator before you run SQL Server Setup because you need this information to create a new failover cluster.

▣ To create a new SQL Server 2005 failover cluster

1. Insert the Microsoft SQL Server 2005 DVD into your DVD drive. If the SQL Server Installation Wizard does not launch automatically, double-click **Splash.hta** in the root directory of the DVD.

If you are installing from a network location, share the installation files and connect to the location from the computer where you want to install SQL Server.

2. To begin the installation process, click **Install SQL Server**.
3. On the **End User License Agreement** page, read the license agreement,

and then select the check box to accept the licensing terms and conditions. Accepting the license agreement activates the **Next** button. To continue, click **Next**. To end Setup, click **Cancel**.

4. SQL Server Component Update runs, installing software required prior to installing SQL Server. For more information about component requirements, click the **Help** button at the bottom of the page. To continue after the component update has run, click **Finish**.
5. On the **Welcome** page of the SQL Server Installation Wizard, click **Next** to continue.
6. On the **System Configuration Check (SCC)** page, the installation computer is scanned for conditions that may block Setup. For more information about configuration check items, see [System Configuration Check \(SCC\)](#).

To interrupt the scan, click **Stop**. To proceed with Setup after the scan completes, click **Continue**.

 **Note:**

If the scan reports any errors, identify the error and click **Exit**. You are not able to complete the installation until blocking errors are removed. For more information, see [Check Parameters for the System Configuration Checker](#).

7. On the **Microsoft SQL Server Installation** page, the SQL Server Installation Wizard prepares to install the software. The Installation Wizard proceeds when the preparation is complete.
8. On the **Registration Information** page, enter information in the **Name** and **Company** text boxes. To continue, click **Next**.
9. On the **Components to Install** page, select the components for your installation.

A description for each component group appears in the **Components to be Installed** pane when you select it. You can select any combination of check boxes. When you select SQL Server or Analysis Services, if

Setup detects that you are installing to a failover cluster, the **Create a SQL Server failover cluster** or **Create an Analysis Server failover cluster** check boxes are enabled. You must select this option to install a failover cluster. If you already have the administration tools installed on another server and you wish to shorten the installation time, you can choose not to install administration tools on the initial installation, allowing you to configure the new server remotely while running SQL Server Setup a second time to install the administration tools.

To install individual components or to specify where files are installed, click **Advanced**. Otherwise, make your selections and then click **Next**.

10. If you clicked **Advanced** on the previous page, the **Feature Selection** page is displayed. On the **Feature Selection** page, select the program features to install using the drop-down boxes. To install components to a custom directory, select the feature and then click **Browse**. For more information about the functionality of this page, see [Feature Selection](#). To continue, click **Next** when your feature selections are complete.
11. On the **Instance Name** page, select a default or named instance for your installation. If a default or named instance is already installed, and you select the existing instance for your installation, Setup upgrades it and provides you the option to install additional components.

To install a new default instance, there must not be a default instance on the computer.

To install a new named instance, click the **Named Instance** and type a unique instance name in the space provided. For more information about instance naming rules, see [Instance Name](#).

12. On the **Virtual SQL Server Name** page, type a unique instance name for your installation. This is the name used to identify the failover cluster on your network. The virtual SQL Server name must be unique on your network, and must have a name that is different than the host cluster and cluster nodes. To proceed, click **Next**.

 **Important:**

You cannot name an instance DEFAULT or MSSQLSERVER. Names

must follow rules for SQL Server identifiers. For more information about instance naming rules, see [Instance Name](#).

 **Note:**

The **Virtual SQL Server Name** page only appears if Setup detects that you are running MSCS. If the **Virtual SQL Server Name** page does not appear, cancel Setup and configure MSCS. For more help on this page, see [Virtual Server Name](#).

13. On the **Virtual SQL Server Configuration** page, enter one IP address for each network configured for client access. That is, enter one IP address for each network on which the failover cluster instance is available to clients on a public (or mixed) network. Only one IP address is allowed on each subnet.

To avoid IP address conflicts, make sure that the IP addresses assigned to the heartbeat and to SQL Server 2005 are not shared by any other application.

Select the network for which you want to enter an IP address, and then enter the IP address. Click **Add**.

The IP address and the subnet are displayed. The subnet is supplied by MSCS. Continue to enter IP addresses for each installed network until you have populated all desired networks with an IP address. To proceed, click **Next**. For more information, see [Virtual Server Configuration](#).

14. On the **Cluster Group Selection** page, select the group that contains the drive you want to use for SQL Server from the **Available Cluster Groups**. The selected cluster group is where the SQL Server virtual SQL Server resources are placed. If you select the group containing the cluster quorum resource, a warning is displayed recommending you do not install to the cluster quorum resource. You can also specify a custom path for data files in the **Data Files** field. To proceed, click **Next**. For more information, see [Cluster Group Selection](#).
15. On the **Cluster Node Configuration** page, specify the nodes to include in the failover cluster instance. To add a node to the failover cluster

instance, select the computer from the **Available Nodes** list, and click **Add**. To remove a node from the failover cluster instance, select the computer from the **Selected Nodes** list, and click **Remove**. Any nodes that are unavailable are displayed in the **Unavailable Nodes** list, as well as the reason the node is unavailable.

To create a one-node failover cluster, only the node that will be installed with the failover cluster instance should be listed in the **Required Node** list.

To proceed, click **Next**. For more information, see [Cluster Node Configuration](#).

16. On the **Remote Account Information** page, specify the remote login information by typing a username and password that has administrator rights **on all nodes** of the cluster. This dialog only appears if you are installing to a remote node. The credentials you provide are used only during SQL Server Setup to configure SQL Server on each node in the failover cluster instance. This account must be a valid administrator account on all nodes of the cluster. To proceed, click **Next**. For more information, see [Remote Account Information](#).
17. On the **Service Accounts (Clusters)** page, specify the username, password and domain name for SQL Server service accounts. You can use one account for all of the services. All service accounts for a failover cluster instance must be domain accounts. Check the **Customize for each service account** box to specify the **Username**, **Password**, and **Domain** for specific services, selecting the service from the **Service** list. The account for the SQL Server Agent service must be an administrator for the cluster node. Do not use the same account that was assigned to the Cluster service.

Optionally, you can specify an individual account for each service. To specify an Individual account for each service, select **Customize for each service account**, select a service name from the drop-down box, and then provide login credentials for the service. To proceed, click **Next**. For more information, see [Service Accounts \(Clusters\)](#).

 **Note:**

The domain name cannot be a full DNS name. For example, if your DNS name is *my-domain-name.com*, use *my-domain-name* in the **Domain** field. SQL Server does not accept *my-domain-name.com* in the **Domain** field.

18. On the **Domain Groups for Clustered Services** page, provide a DomainName\GroupName for each clustered service being installed: the SQL Server service, SQL Server Agent service, Analysis Services service, and Full-Text Search service must run as domain accounts that are members of the global or local domain group. To proceed, click **Next**. For more information, see [Domain Groups for Clustered Services](#).
19. On the **Authentication Mode** page, choose the authentication mode to use for your SQL Server installation. You must also enter and confirm a strong password for the System Administrator (**sa**) login. To proceed, click **Next**.

 **Security Note:**

When possible, use Windows Authentication.

 **Security Note:**

For this release, SQL Server Setup may not enforce the strong password requirement on some default configurations of Windows Server 2003 where the machine is not a member of a domain. Setting strong passwords is essential to the security of your system. Always use strong passwords. For strong password guidelines, see [Authentication Mode](#) in SQL Server 2005 Books Online.

20. On the **Collation Settings** page, specify the sorting behavior for your server. You can use one collation for SQL Server and Analysis Services, or you can specify collations for individual components.

To set separate collation settings for SQL Server and for Analysis Services, select the check box to **Customize for each service account**. After you select the check box, a drop-down selection box appears. Select a service from the drop-down selection box and then assign its collation. Repeat for each service. To proceed, click **Next**. For more

information, see [Collation Settings](#).

21. On the **Error and Usage Report Settings** page, optionally clear the checkboxes to disable error reporting or feature usage reporting. To proceed, click **Next**. For more information, see [Error and Usage Report Settings](#).
22. On the **Ready to Install** page, review the summary of features and components for your SQL Server installation. To proceed, click **Install**.
23. On the **Installation Progress** page, you can monitor installation progress on individual cluster nodes by selecting a node from the Node drop-down selection box. Setup progress is displayed for the selected node. To view the log file for a component during installation, click the product or status name on the **Installation Progress** page.
24. On the **Completing the Microsoft SQL Server Installation Wizard** page, you can view the Setup summary log by clicking the link provided on this page. To exit the SQL Server Installation Wizard, click **Finish**.
25. If you are instructed to restart the computer, do so now. It is important to read the message from the Setup program when you are done with installation. Failure to restart any of the specified nodes may cause failures when you run the Setup program in the future.

▣ **Next Steps**

- **Configure your new SQL Server installation** - To reduce the attackable surface area of a system, SQL Server 2005 selectively installs and activates key services and features. For more information on how to activate SQL Server 2005 features, see [SQL Server Surface Area Configuration](#).

For information about log file locations, see [How to: View SQL Server 2005 Setup Log Files](#).

For information about verifying your new SQL Server installation, see [How to: Verify a Successful Installation of SQL Server 2005 Services](#).

▣ **See Also**

Tasks

[How to: Install SQL Server 2005 from the Command Prompt](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Add or Remove Nodes in a SQL Server 2005 Failover Cluster (Setup)

 [See Also](#)



[Configuring High Availability](#) > [Configuring Availability How-to Topics](#) > [Failover Clustering How-to Topics](#) >

Updated: **12 December 2006**

Use this procedure to manage nodes in a Microsoft SQL Server 2005 failover clustered instance.

Important:

To update or remove a SQL Server failover cluster, you must be a local administrator with permission to log on as a service on all nodes of the failover cluster.

To add or remove nodes in a SQL Server 2005 failover cluster

1. Double-click **Add or Remove Programs** in Control Panel.
2. Select the SQL Server 2005 component to update, and then click **Change**. The SQL Server 2005 Installation Wizard starts.
3. On the **Component Selection** page, select the SQL Server 2005 instance name to change. Click **Next**.
4. On the **Feature Maintenance** page, select a component to change from within the selected SQL Server 2005 instance. Click **Next**. The SQL Server 2005 System Configuration Checker starts.

On the **Welcome** page, click **Next**. The System Configuration Checker

scans your computer's existing configuration.

5. After the System Configuration Checker finishes its scan, click **Next**.
6. On the **Change or Remove Instance** page, click **Maintain the Virtual Server**.
7. On the **Cluster Node Configuration** page, specify the nodes to include in the failover cluster instance. To add a node to the failover cluster, select the computer from the **Available Nodes** list and click **Add**. To remove a node from the failover cluster, select the computer from the **Selected Nodes** list and click **Remove**. Unavailable nodes are displayed in the **Unavailable Nodes** list, along with the reason for unavailability.

To create a one-node failover cluster, only the node that will be installed with the failover cluster should be listed in the **Required Node** list.

Click **Next**. For more information, see [Cluster Node Configuration](#).

8. On the **Ready to Update** page, review the summary for your SQL Server installation. When you are satisfied with your choices, click **Update**.
9. On the **Installation Progress** page, you can monitor installation progress for individual cluster nodes by selecting a node from the **Node** list. Setup progress is displayed for the selected node. To view the log file for a component during installation, click the product or status name on the **Installation Progress** page.
10. On the **Completing the Microsoft SQL Server Installation Wizard** page, you can view the Setup summary log by clicking the link provided on this page. To exit the SQL Server Installation Wizard, click **Finish**.

If you are instructed to restart the computer, do so now. It is important to follow instructions from the Setup program after installation is complete. Failure to restart any of the specified nodes may cause failures when you run the Setup program again.

11. All nodes of a failover cluster instance must be at the same version level. After completing SQL Server Setup, you must download and

apply the latest SQL Server 2005 service pack and/or patches to ensure that all failover cluster nodes are at the same version level.

▣ **See Also**

Tasks

[How to: Read a SQL Server 2005 Setup Log File](#)

[How to: Manually Uninstall a SQL Server 2005 Failover Cluster](#)

Concepts

[How to: View SQL Server 2005 Setup Log Files](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Change the IP Address of a SQL Server 2005 Failover Cluster



[See Also](#)



[Configuring High Availability](#) > [Configuring Availability How-to Topics](#) > [Failover Clustering How-to Topics](#) >

To maintain or update a Microsoft SQL Server 2005 failover cluster, use **Add or Remove Programs** in Control Panel. For information about command prompt syntax, see [How to: Install SQL Server 2005 from the Command Prompt](#).

Before you begin, review the following Books Online topic: [Before Installing Failover Clustering](#).

For additional information about changing the IP address of network adapters in the nodes of a cluster, see [this Microsoft Knowledge Base article](#).

Important:

To maintain or update a SQL Server failover cluster, you must be a local administrator with permission to logon as a service **on all nodes** of the failover cluster.

To change the IP address of an existing SQL Server 2005 failover cluster

1. From the Start menu, click **Run**, and then type **cluadmin .** to launch the Cluster Administrator and connect to the local instance.
2. In the left-hand pane, expand the Group node and select the Group to be updated.
3. In the right-hand pane, right-click the SQL IP Address1(failover cluster

instance name) and select **Properties**.

4. Click the **Parameters** tab, and then update the IP address for SQL IP Address1(SQL Server Name) in the **Address** field.
5. Click **Apply** to implement the change. You may see a warning dialog that says:

"Cluster Administrator Standard Extension: An error occurred attempting to set properties: The properties were stored but not all changes will take effect until the next time the resource is brought online. Error ID: 5024 (000013a0)."

6. Click **OK** to clear the **Cluster Administrator Standard Extension** dialog, and you will be returned to the Cluster Administrator. To take resources offline and then put them back online, use the following steps.
7. In the right-hand pane, right-click the SQL IP Address1(failover cluster instance name) and select **Take Offline**. You will see the SQL IP Address1(failover cluster instance name), SQL Network Name(failover cluster instance name), and SQL Server status change from Online to Offline Pending, and then to Offline.

 **Note:**

It may take 30 minutes or more for IP changes to propagate through a corporate network. You should try to bring the SQL IP Address 1(failover cluster instance name) and the SQL Network Name(failover cluster instance name) online, then try to ping the failover cluster instance name and wait for it to be resolved by the DNS to the new IP address.

8. In the right-hand pane, right-click SQL Server, and then select **Bring Online**. You will see the SQL IP Address1(failover cluster instance name), SQL Network Name(failover cluster instance name), and SQL Server status change from Offline to Online Pending, and then to Online.
9. Close Cluster Administrator.

▣ **See Also**

Concepts

[How to: View SQL Server 2005 Setup Log Files](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Install Microsoft Distributed Transaction Coordinator



[See Also](#)



[Configuring High Availability](#) > [Configuring Availability How-to Topics](#) > [Failover Clustering How-to Topics](#) >

After you install the operating system and configure your cluster, you must configure Microsoft Distributed Transaction Coordinator (MS DTC) on Microsoft Windows 2000 and Microsoft Windows Server 2003 operating systems to work in a cluster by using the Cluster Administrator. Failure to cluster MS DTC will not block SQL Server 2005 Setup, but SQL Server 2005 application functionality may be affected if MS DTC is not properly configured. To help ensure availability between multiple clustered applications, it is highly recommended that the Configure Microsoft Distributed Transaction Coordinator have its own resource group and resources.

To manually configure MS DTC on a Windows Server 2003 operating system (running MSCS)

1. See [How to Configure Microsoft Distributed Transaction Coordinator on a Windows Server](#).

▣ **See Also**

Tasks

[How to: Create a New SQL Server 2005 Failover Cluster \(Setup\)](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Enable Windows Task Scheduler Service



[Configuring High Availability](#) > [Configuring Availability How-to Topics](#) > [Failover Clustering How-to Topics](#) >

The procedure in this topic is resolution for a situation where SQL Server Setup fails on a cluster with Error 1058, due to a disabled Windows Task Scheduler Service on cluster nodes. If the Windows Task Scheduler Service is disabled on a cluster node, the error in the in_core.log will be similar to:

```
Connection to remote computer's scheduler service.
```

```
Error: SetTargetComputer on \\machinename failed with  
Unable to start service (1058)
```

The error text displayed in the Windows user interface will be

The service cannot be started, either because it is disabled or because it has no enabled devices associated with it.

Resolution to this error is to use cluster manager to enable Task Scheduler Service on all cluster nodes before running Microsoft SQL Server 2005 Setup.

To enable the Task Scheduler Service

1. From Administrative Tools in Windows, double-click **Services**.
2. In the **Name** column, right-click **Task Scheduler**, and then click **Start**.
3. Run SQL Server 2005 Setup to install SQL Server.

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Enable Kerberos Authentication on a SQL Server Failover Cluster



[See Also](#)



[Configuring High Availability](#) > [Configuring Availability How-to Topics](#) > [Failover Clustering How-to Topics](#) >

Kerberos is a network authentication protocol designed to provide strong authentication for client/server applications. Kerberos provides a foundation for interoperability while enhancing the security of enterprise-wide network authentication.

You can use Kerberos authentication with Microsoft SQL Server 2005 stand-alone instances or with failover cluster instances running on Microsoft Windows 2000 Service Pack 3 (SP3). SQL Server 2005 supports this functionality as part of a typical Microsoft Windows 2000 Active Directory domain installation.

When the Network Name resource that SQL Server is dependant on is in a Windows 2000-based cluster, you can use Kerberos authentication on the resource after you upgrade the computer to Windows 2000 SP3. For additional information about enabling Kerberos on server clusters, see the Knowledge Base article, "[Kerberos Support on Windows 2.](#)"

The following section describes how to connect to a server that is running Microsoft Internet Information Services (IIS) to make a Kerberos connection to a server that is running SQL Server.

▣ Prerequisites

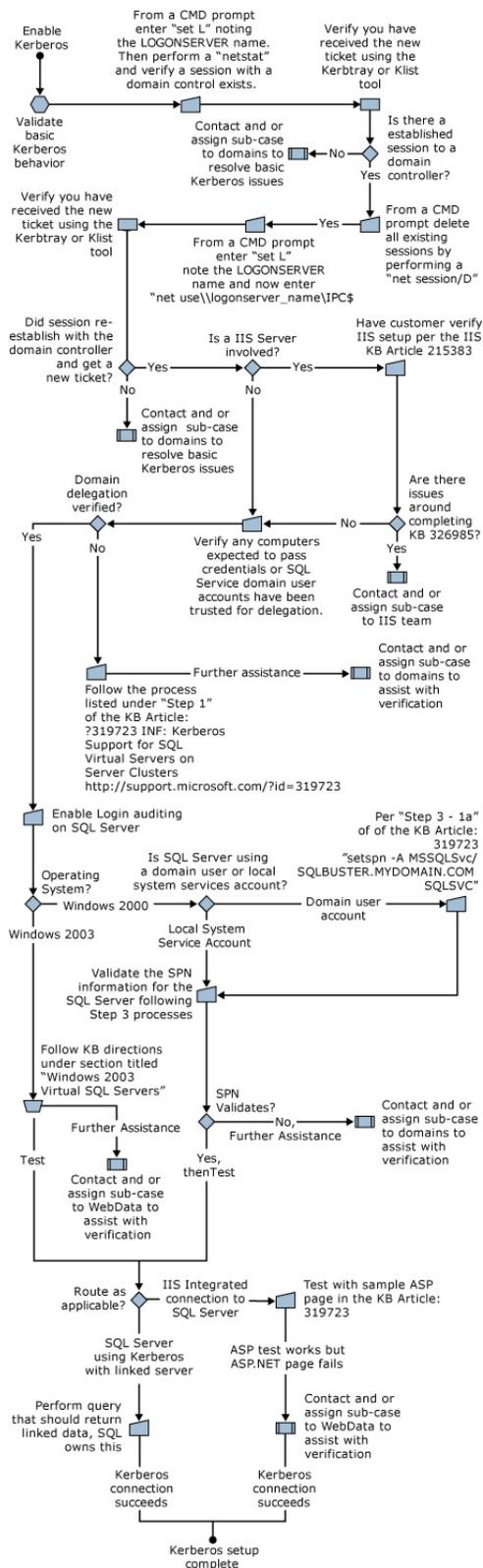
You can only use this functionality if you are running Windows 2000 SP3.

Before you perform the Setup procedure, download the Kerbtray and the SetSPN utilities.

- To download the Kerbtray utility, visit the following [Micro](#). With Kerbtray.exe, you can easily verify or remove (or both) Kerberos tickets from any of the associated computers that are being used.
- To download the SetSPN utility, visit the following [Microsoft Web site](#).

SQL Server only uses Kerberos if the client uses the TCP/IP protocol to connect to SQL Server. For example, if a client uses the Named Pipes protocol, Kerberos is not used. If you have multiple instances of SQL Server on a computer, you must configure a Server Principal Name (SPN) for each instance of SQL Server because each instance of SQL Server uses a unique TCP-IP port.

Flow chart showing at which points during the execution of the KB Article, 319723 INF: Kerberos Support for SQL Virtual Servers on Server Clusters other technology teams should be engaged for assistance.



□ **Connect to a server that is running Microsoft Internet Information Services and make a Kerberos connection to SQL Server 2005**

1. Step 1: Configure the domain controller

On a domain controller, in Active Directory Users and Computers:

1. Right-click the computer that you want to set up for delegation (IIS Services server), and then click to select **Trust this computer for delegation**. If the computer that is running SQL Server is what appears to be the last computer contacted but that computer has a linked server, it must also be granted delegation permissions. If it is not the last computer in the chain, all the computers that are intermediaries must be trusted for delegation.
2. Grant delegation permission to the SQL Server service account domain user account. You must have a domain user account for clustered SQL Server installations (this step is not required for computers that are running SQL Server that are using a local system account):
 1. In the Users folder, right-click the user account, and then click **Properties**.
 2. In the user account properties dialog box, click the **Account** tab.
 3. Under **Account Options**, click to select the **Account is Trusted for Delegation** check box. Make sure that the **Account is sensitive and cannot be delegated** check box is cleared for this account.
3. Use the Kerbtray.exe utility to verify that Kerberos tickets were received from the domain controller and host:
 1. Right-click the Kerbtray icon in the notification area, and then click **purge tickets**.
 2. Wait for the green Kerbtray icon to change from green to yellow. As soon as this occurs, open a command prompt window and run this command:

```
net session* /d
```

This will drop the existing sessions, and force a new session to be established and a Kerberos ticket received.

Step 2: Configure the IIS services server

1. Replace the default Web site **Wwwroot** files with the sample .asp files. To create the sample .asp files, use the code that is provided in the "**ASP test script for SQL Server data retrieval**" section.
2. Add the file to the **Wwwroot** folder. To do so, use the sample code in the "**ASP Test Script for SQL Server Data Retrieval**" section. Save the file as **Default.asp**.
3. Reconfigure the Web server to use Integrated Windows Authentication only:
 1. Right-click the default Web server, and then click the **Security** folder.
 2. In the **Security** folder, make the correct changes, and then click to clear **anonymous access**.
 3. From a command prompt, run this command:

```
cscript C:\Inetpub\Adminscripts\adsutil.vbs get w3svc/NTAuthenticationProviders
```

If **Negotiate** is enabled, the following is returned:

```
NTAuthenticationProviders : (STRING) Negotiate,NTLM
```

For additional information on configuring IIS to support both Kerberos and NTLM authentication, see the Knowledge Base article, "[How to Configure IIS to Support Both Kerberos and NTLM Authentication](#)."

 **Note:**

You must install Microsoft Data Access (MDAC) 2.8 SP1, or later, on the IIS Services server. To do so (and to make the tools available for testing), install the Microsoft SQL Server 2000 client tools to the Web server. To install only MDAC 2.8 SP1 or later without installing the client tools, visit the following [Microsoft Web site](#).

1. Verify that the HKLM\SW\MS\MSSQLSERVER\Client\DSQUERY value is present in the registry. If the value is not displayed, add it as DSQUERY:Reg_SZ:DBNETLIB.
2. Use the Kerbtray.exe utility to verify that Kerberos tickets were received from the domain controller and host:
 1. Right-click the Kerbtray icon in the notification area, and then click **purge tickets**.
 2. Wait for the green Kerbtray icon to change from green to yellow. As soon as this occurs, open a command prompt window and run this command:

```
net session * /d
```

This drops the existing sessions, and forces a new session to be established and a Kerberos ticket received.

Step 3: Create an SPN for SQL Server

Caution:

SQL Server only uses Kerberos if the client uses the TCP/IP protocol to connect to SQL Server. For example, if a client uses the Named Pipes protocol, Kerberos is not used. If you have multiple instances of SQL Server on a computer, you must configure a Server Principal Name (SPN) for each instance of SQL Server because each instance of SQL Server uses a unique TCP-IP port.

Important:

If the SQL Server service is running under the **LocalSystem** account, you do not have to manually configure an SPN for SQL Server. The SPN is created automatically when the SQL Server service starts. If the SQL Server service runs under a domain user account, you must manually configure an SPN. To do so, follow these steps.

To configure an SPN for SQL Server, use the SETSPN utility in the Microsoft Windows Resource Kit. To download the SETSPN utility, visit the following [Mic](#).

Before you run SETSPN, consider the following information:

- You must run **setspn.exe** under a logon account with permissions to register the SPN.
- Note the domain user account that the instance of SQL Server is running under. In the following examples, this account is named <SQL_Service_Account>.

Important If the instance of SQL Server is running under the **LocalSystem** account, you do not have to run the SETSPN utility.

- You must have the fully qualified domain name (FQDN) of the computer that is running SQL Server. To determine the FQDN of the computer that is running SQL Server, use the ping utility. To do so, follow these steps:

1. Ping the computer that is running SQL Server to determine its IP address:

```
C:\>ping MySQLServer
```

```
Pinging MySQLServer.MyDomain.com [10.10.10.10] with 32 bytes of data:
```

```
Reply from 10.10.10.10: bytes=32 time=1ms TTL=128
```

2. Use **ping -a** to perform a reverse lookup of the IP address to make sure that the FQDN is correctly returned by the Domain Name System (DNS) protocol:

```
C:\>ping -a 10.10.10.10
```

```
Pinging MySQLServer.MyDomain.com [10.10.10.10] with 32 bytes of data:
```

```
Reply from 10.10.10.10: bytes=32 time<1ms TTL=128
```

3. Ping the failover cluster instance name to obtain the IP address, and then run `ping -a` to make sure that the FQDN is correctly returned by the DNS.

 **Note:**

If you are using SQL Server failover clustering, you use the FQDN for the failover cluster instance name. Note the exact TCP/IP port that the instance of SQL Server uses. To determine this information, open SQL Server Configuration Manager on the computer that is running SQL Server, click the instance of SQL Server, and then view the properties for the TCP/IP protocol (default port).

After you determine the domain user account that the SQL Server service is running under, the FQDN of the computer that is running SQL Server, and the TCP/IP port that the instance of SQL Server is using, follow these steps to create the SPN for SQL Server.

 **Note:**

You must be a member of the Domain Administrators group to run the **SETSPN** command.

1. If you are using SQL Server failover clustering, run the following SETSPN command:

```
setspn -A MSSQLSvc/<FQDN> <SQL_Service_Account>
```

For example, if MySQLServer.MyDomain.com is running under the domain user account SQLSVC, where MySQLServer.MyDomain.com is the name of the instance of SQL Server 2005 that is clustered, run the

following command:

```
setspn -A MSSQLSvc/MySQLServer.MyDomain.com SQLSVC
```

2. For both clustered and non-clustered computers that are running SQL Server, run the following SETSPN command to register an SPN for the port that the computer that is running SQL Server is using:

```
setspn -A MSSQLSvc/<FQDN>:<Port> <SQL_Service_Account>
```

For example, if MySQLServer.MyDomain.com is running under the domain user account SQLSVC on port 1433, run the following SETSPN command:

```
setspn -A MSSQLSvc/MySQLServer.MyDomain.com:1433 SQLSVC
```

3. After the SPN is registered, verify that it is correctly registered by using the LIST feature (-L switch) of the SETSPN utility. Run SETSPN -L <SQL_Service_Account> to list all the SPNs that are registered to the domain user account that the instance of SQL Server is running under:

```
setspn -L <SQL_Service_Account>
```

For example, if MySQLServer.MyDomain.com is running under the domain user account SQLSVC on port 1433, run the following command:

```
setspn -A SQLSVC
```

The SPN that you created in step 1 (if SQL Server is clustered) and step 2 (if SQL Server is not clustered) is shown in the following output:

```
C:\>setspn -l SQLSVC
```

```
Registered ServicePrincipalNames for
```

```
CN=SQLSVC,CN=Users,DC=MyDomain,DC=com:
```

```
    MSSQLSvc/MySQLServer.MyDomain.com
```

```
    MSSQLSvc/MySQLServer.MyDomain.com:1433
```

 **Note:**

If you are using SQL Server failover clustering, you must register a SPN without the port number and another SPN with the port number. With a typical, non-clustered computer that is running SQL Server, you only have to register the SPN with the port number. However, if you have an additional SPN without the port number, it will not cause any problems with non-clustered computers.

 **Note:**

You can also use the **Ldifde.exe** utility on the domain controller to verify both SPN registrations. This is described in the "How to gather a list of Active Directory server principal name information" section in this topic.

Step 4: Configure the client computers

1. For each client that will connect, verify that Microsoft Internet Explorer is configured to use Windows Authentication:
2. In Internet Explorer, on the **Tools** menu, click **Internet Options**.
3. Click the **Advanced** tab.
Under **Security**, click to select **Enable Integrated Windows Authentication (requires restart)**, and then click **OK**.

Step 5: Test the configuration

For each computer that is involved:

1. Log on to the computer, and then use Kerbtray.exe to verify that the computer can obtain a valid Kerberos ticket from the domain controller.
2. Use Kerbtray.exe to remove all tickets on the computer.
3. Create and connect to the Web page that returns the SQL Server data.

Note:

Replace SQLSERVERNAME with the name of the computer that is running SQL Server:

- If data is returned, this page displays the authentication type **Negotiate**, and the SQL Server data for the result of the **sp_helpdb** stored procedure that should return a list of the databases on the server that is being connecting to through the .ASP page.
- If you have auditing turned on in SQL Server, in the Application log you will see that the connection is "trusted".

ASP test script for SQL Server data retrieval

The following is an ASP test script for SQL Server data. If you use this code sample, make sure that you replace SQLSERVERNAME with the name of the computer that is running SQL Server:

```
<%@ Language=VBScript %>
```

```
<HTML>
```

```
<HEAD>

<META NAME="GENERATOR" Content="Microsoft Visual Studio 6.0">

</HEAD>

<BODY>

<%= "'auth_user' is" & request.servervariables("auth_user")%>

<P>

<%= "'auth_type' is" & request.servervariables("auth_type")%>

<P>

Connections string is <B>" Provider=SQLOLEDB.1;Integrated
Security=SSPI;Persist Security Info=False;Initial Catalog=pubs;Data
Source=SQLSERVERNAME </B>

<P>

<%

set rs = Server.CreateObject("ADODB.Recordset")

set cn = Server.CreateObject("ADODB.Connection")

cn.Open "Provider=SQLOLEDB.1;Integrated Security=SSPI;Persist
Security Info=False;Initial Catalog=pubs;Data Source=SQLSERVERNAME"

rs.open "MASTER..sp_helpdb",cn

Response.Write cstr(rs.Fields.Count) + "<BR>"

while not rs.EOF

Response.Write cstr(rs(0)) + "<BR>"

rs.MoveNext

wend

rs.Close

cn.Close
```

```
set rs = nothing ' Frees memory reserved by the recordset.  
set cn = nothing ' Frees memory reserved by the connection.  
%>  
</BODY>  
</HTML>
```

How to gather a list of Active Directory server principal name information

To gather a list of Active Directory server principal name (SPN) information, type the following command on one of your domain controllers, where betaland is the NetBIOS domain name and NewoutputUsers.txt is the name of the output file that you will use to port the results. If you do not use a full path, the file is placed in the current folder where you run the command line. This sample command queries the whole domain:

```
ldifde -d "CN=Users,DC=betaland" -l servicePrincipalName -F Newoutput
```

This syntax creates a file named NewoutputUsers.txt that contains information that is similar to the output in the "Domain level output of NewouputUsers.txt" section of this topic.

This output may be overwhelming when you gather it for a whole domain. Therefore, to limit the gathered information to a specific user name, use the following syntax, where User Name is the user name and betaland is the domain that you are querying:

```
ldifde -d "CN=User Name,DC=betaland" -l servicePrincipalName -F Newo
```

Gathering the information for a specific user greatly reduces the data that you must search through. If you gather the information for a whole domain, search for the specific user name of the server in question. In the output sample, you see:

- Entries for servers that no longer exist, but that were not completely removed from Active Directory.

- The user "User Name" has valid SPN information on about ten different servers.

Additionally, you can use the Active Directory Service Interfaces (ADSI) tool to correct Active Directory entries that are not valid.

Caution If you use the ADSI Edit snap-in, the LDP utility, or any other LDAP version 3 client, and you incorrectly modify the attributes of Active Directory objects, you can cause serious problems. These problems may require you to reinstall Microsoft Windows 2000 Server, Microsoft Exchange 2000 Server, or both. Microsoft cannot guarantee that problems that occur if you incorrectly modify Active Directory object attributes can be solved. Modify these attributes at your own risk.

Domain level output of NewouputUsers.txt

dn: CN=User Name,CN=Users,DC=betaland

changetype: add

servicePrincipalName: MSSQLSvc/CLUSTERDEFAULT.betaland:1257

servicePrincipalName: MSSQLSvc/INST3.betaland:3616

servicePrincipalName: MSSQLSvc/INST2.betaland:3490

servicePrincipalName: MSSQLSvc/SQLMAN.betaland:1433

servicePrincipalName: MSSQLSvc/VSS1.betaland:1433

servicePrincipalName: MSSQLSvc/INST1.betaland:2536

servicePrincipalName: MSSQLSvc/INST4.betaland:3967

servicePrincipalName: MSSQLSvc/SQLVIRTUAL1.betaland:1434

servicePrincipalName: MSSQLSvc/SQLVIRTUAL.betaland:1433

servicePrincipalName: MSSQLSvc/SQLBUSTER.betaland:1315

▣ **See Also**

Concepts

[Installing SQL Server 2005](#)

Other Resources

[Configuring Availability How-to Topics](#)

[Upgrading to SQL Server 2005](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Upgrade to a SQL Server 2005 Failover Cluster (Setup)



[See Also](#)



[Configuring High Availability](#) > [Configuring Availability How-to Topics](#) > [Failover Clustering How-to Topics](#) >

Updated: **5 December 2005**

You can install Microsoft SQL Server 2005 or upgrade to SQL Server 2005 from a previous SQL Server version using the SQL Server 2005 Installation Wizard.

For upgrade information, see [Version and Edition Upgrades](#) and [Upgrading to SQL Server 2005](#).

To install a stand-alone instance of SQL Server 2005, see [How to: Install SQL Server 2005 \(Setup\)](#).

▣ Prerequisites

Before you begin, review the following important information:

- [Preparing to Install SQL Server 2005](#)
- [Before Installing Failover Clustering](#)
- Review [Using Upgrade Advisor to Prepare for Upgrades](#).
- [Upgrading the Database Engine](#)
- It is very important that you read and explicitly follow instructions in the article on this [Microsoft](#).
- For local installations, you must run SQL Server Setup as an administrator. If you install SQL Server from a remote share, you must use a domain account that has read and execute permissions on the remote share.
- To upgrade a SQL Server instance to a SQL Server 2005 failover cluster, the instance being upgraded must be a failover cluster. To upgrade a stand-alone instance of SQL Server to a SQL Server 2005 failover cluster, install a new SQL Server 2005 failover cluster and then migrate user databases from the stand-alone instance using the Copy Database Wizard. For more information on database migration, see **Using the Copy Database Wizard**.

▣ To upgrade to a SQL Server 2005 failover cluster

1. Insert the SQL Server 2005 DVD into your DVD drive. If the SQL Server Installation Wizard does not launch automatically, double-click **Splash.hta** in the root directory of the DVD.

If you are installing from a network location, share the installation files and connect to the location from the computer where you want to install SQL Server.

2. To begin the installation process, click **Install SQL Server**.
3. On the **End User License Agreement** page, read the license agreement, and then select the check box to accept the licensing terms and conditions. Accepting the license agreement activates the **Next** button.

To continue, click **Next**. To end Setup, click **Cancel**.

4. SQL Server Component Update runs, installing software required prior to installing SQL Server. For more information about component requirements, click the **Help** button at the bottom of the page. To continue after the component update has run, click **Next**.
5. On the **Welcome** page of the SQL Server Installation Wizard, click **Next** to continue.
6. On the **System Configuration Check (SCC)** page, the installation computer is scanned for conditions that may block Setup. For more information about configuration check items, see [System Configuration Check \(SCC\)](#).

To interrupt the scan, click **Stop**. To proceed with Setup after the scan completes, click **Next**.

 **Note:**

If the scan reports any errors, identify the error and click **Exit**. You are not able to complete the installation until blocking errors are removed. For more information, see [Check Parameters for the System Configuration Checker](#).

1. On the **Microsoft SQL Server Installation** page, the SQL Server Installation Wizard prepares to install the software. The Installation Wizard proceeds when the preparation is complete.
2. On the **Registration Information** page, enter information in the **Name** and **Company** text boxes. To continue, click **Next**.
3. On the **Components to Install** page, select the components for your installation.

A description for each component group appears in the **Components to be Installed** pane when you select it. You can select any combination of check boxes. When you select SQL Server or Analysis Services, if Setup detects that you are installing to a SQL Server failover cluster, the **Create a SQL Server failover cluster** or **Create an Analysis Server failover cluster** check boxes are enabled. You must select this option to install a failover cluster. If you already have the administration tools installed on another server and you wish to shorten the installation time, you can choose not to

install administration tools on the initial installation, allowing you to configure the new server remotely while running SQL Server Setup a second time to install the administration tools.

To install individual components or to specify where files are installed, click **Advanced**. Otherwise, make your selections and then click **Next**.

4. If you clicked **Advanced** on the previous page, the **Feature Selection** page is displayed. On the **Feature Selection** page, select the program features to install using the drop-down boxes. To install components to a custom directory, select the feature and then click **Browse**. For more information about the functionality of this page, see [Feature Selection](#). To continue, click **Next** when your feature selections are complete.
5. On the **Instance Name** page, select a default or named instance for your installation. If a default or named instance is already installed, and you select the existing instance for your installation, Setup upgrades it and provides you the option to install additional components.
To install a new default instance, there must not be a default instance on the computer.
To install a new named instance, click the **Named Instance** and type a unique instance name in the space provided. For more information about instance naming rules, see [Instance Name](#).
6. On the **Virtual SQL Server Name** page, type a unique instance name for your installation. This is the name used to identify the failover cluster on your network. The SQL Server failover cluster instance name must be unique on your network, and must have a name that is different than the host cluster and cluster nodes. To proceed, click **Next**.

 **Important:**

You cannot name an instance DEFAULT or MSSQLSERVER. Names must follow rules for SQL Server identifiers. For more information about instance naming rules, see [Instance Name](#).

 **Note:**

The **Virtual SQL Server Name** page only appears if Setup detects that you are running MSCS. If the **Virtual SQL Server Name** page does not appear, cancel Setup and configure MSCS. For more help on this page, see [Virtual Server Name](#).

1. On the **Existing Components** page, select the component you would like to upgrade, and then click **Details** to view the report that verifies that upgrade is an available option for that component. If the Installation Options report shows that the component is available to upgrade, click **Close** to return to the **Existing Components** page, and then click **Next** to proceed with Setup.
2. On the **Service Accounts (Clusters)** page, specify the username, password and domain name for SQL Server service accounts. You can use one account for all of the services. All service accounts for a SQL Server failover cluster instance must be domain accounts. Check the **Customize for each service account** box to specify the **Username, Password, and Domain** for specific services, selecting the service from the **Service** list. The account for the SQL Server Agent service must be an administrator for the cluster node. Do not use the same account that was assigned to the Cluster service.

Optionally, you can specify an individual account for each service. To specify an Individual account for each service, select **Customize for each service account**, select a service name from the drop-down box, and then provide login credentials for the service. To proceed, click **Next**. For more information, see [Service Accounts \(Clusters\)](#).

 **Note:**

The domain name cannot be a full DNS name. For example, if your DNS name is *my-domain-name.com*, use *my-domain-name* in the **Domain** field. SQL Server does not accept *my-domain-name.com* in the **Domain** field.

1. On the **Upgrade Logon Information** page, provide logon credentials for the SQL Server instance to be upgraded, as the SQL Server 2005 Installation Wizard must connect to the existing SQL Server instance during the upgrade. To proceed, click **Next**.
2. On the **Error and Usage Report Settings** page, optionally clear the checkbox to disable error reporting or feature usage reporting. To proceed, click **Next**. For more information, see [Error and Usage Report Settings](#).
3. On the **Ready to Install** page, review the summary of features and components for your SQL Server installation. To proceed, click **Install**.
4. On the **Completing the Microsoft SQL Server Installation Wizard** page, you can view the Setup summary log by clicking the link provided on this page. To exit the SQL Server Installation Wizard, click **Finish**.
5. If you are instructed to restart the computer, do so now. It is important to

read the message from the Setup program when you are done with installation. Failure to restart any of the specified nodes may cause failures when you run the Setup program in the future.

▣ Next Steps

- **Configure your new SQL Server installation.** To reduce the attackable surface area of a system, SQL Server 2005 selectively installs and activates key services and features. For more information about how to activate SQL Server 2005 features, see [SQL Server Surface Area Configuration](#).

For information about log file locations, see [How to: View SQL Server 2005 Setup Log Files](#).

For information about verifying your new SQL Server installation, see [How to: Verify a Successful Installation of SQL Server 2005 Services](#).

▣ **See Also**

Tasks

[How to: Install SQL Server 2005 from the Command Prompt](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Rename a SQL Server 2005 Virtual Server



[See Also](#)



[Configuring High Availability](#) > [Configuring Availability How-to Topics](#) > [Failover Clustering How-to Topics](#) >

When a SQL Server instance is part of a failover cluster, the process of renaming the virtual server differs from that of renaming a stand-alone instance. For information on renaming a stand-alone instance of SQL Server 2005, see [How to: Rename a Computer that Hosts a Stand-Alone Instance of SQL Server 2005](#).

The name of the virtual server is always the same as the name of the SQL Network Name (the SQL Virtual Server Network Name). Although you can change the name of the virtual server, you cannot change the instance name. For example, you can change a virtual server named VS1\instance1 to some other name, such as SQL35\instance1, but the instance portion of the name, instance1, will remain unchanged.

Before you begin

Before you begin the renaming process, review the items below.

- SQL Server does not support renaming servers involved in replication, except in the case of using log shipping with replication. The secondary server in log shipping can be renamed if the primary server is permanently lost. For more information, see **Replication and Log Shipping**.
- When renaming a virtual server that is configured to use database mirroring, you must turn off database mirroring before the renaming operation, and then re-establish database mirroring with the new virtual server name. Metadata for database mirroring will not be updated automatically to reflect the new virtual server name.

To rename a virtual server

1. Using Cluster Administrator, change the SQL Network Name to the new name.
2. Take the network name resource offline. This takes the SQL Server resource and other dependent resources offline as well.
3. Bring the SQL Server resource back online.

▣ **Verify the Renaming Operation**

After a virtual server has been renamed, any connections that used the old name must now connect using the new name.

To verify that the renaming operation has completed, select information from either `@@servername` or `sys.servers`. The `@@servername` function will return the new virtual server name, and the `sys.servers` table will show the new virtual server name. To verify that the failover process is working correctly with the new name, the user should also try to fail the SQL Server resource over to the other nodes.

For connections from any node in the cluster, the new name can be used almost immediately. However, for connections using the new name from a client computer, the new name cannot be used to connect to the server until the new name is visible to that client computer. The length of time required for the new name to be propagated across a network can be a few seconds, or as long as 3 to 5 minutes, depending on the network configuration; additional time may be required before the old virtual server name is no longer visible on the network.

To minimize network propagation delay of a virtual server renaming operation, use the following steps:

▣ **To minimize network propagation delay**

1. Issue the following commands from a command prompt on the server node:

```
ipconfig /flushdns  
ipconfig /registerdns  
nbtstat -RR
```

 [Copy Code](#)

▣ **See Also**

Tasks

[How to: Rename a Computer that Hosts a Stand-Alone Instance of SQL Server 2005](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Add Dependencies to a SQL Server 2005 Resource



[See Also](#)



[Configuring High Availability](#) > [Configuring Availability How-to Topics](#) > [Failover Clustering How-to Topics](#) >

If you install Microsoft SQL Server 2005 into a Windows 2000 cluster group with multiple disk drives and choose to place your data on one of the drives, the SQL Server resource will be set to be dependent only on that drive. To put data or logs on another disk, you must first add a dependency to the SQL Server resource for the additional disk.

It is important to not that if you add any other resources to the SQL Server group, those resources must always have their own unique SQL network name resources and their own SQL IP address resources.

Do not use the existing SQL network name resources and SQL IP address resources for anything other than SQL Server. If SQL Server resources are shared with other resources, the following problems may occur:

- Outages that are not expected may occur.
- Service pack installations may not be successful.
- The SQL Server Setup program may not be successful. If this problem occurs, you cannot install additional instances of SQL Server or perform routine maintenance.

Consider these additional issues:

- FTP with SQL Server replication: For instances of SQL Server that use FTP with SQL Server replication, your FTP service must use one of the same physical disks as the installation of SQL Server that is set up to use the FTP service.
- SQL Server resource dependencies: If you add a resource to a SQL Server group and you have a dependency on the SQL Server resource to make sure that SQL Server is available, Microsoft recommends that you add a dependency on the SQL Server Agent resource. Do not add a dependency on the SQL Server resource. To make sure that the computer

that is running SQL Server remains highly available, configure the SQL Server Agent resource so that it does not affect the SQL Server group if the SQL Server Agent resource fails.

- File shares and printer resources: When you install File Share resources or Printer cluster resources, they should not be put on the same physical disk resources as the computer that is running SQL Server. If they are put on the same physical disk resources, you may experience performance degradation and loss of service to the computer that is running SQL Server.
- MS DTC considerations: After you install the operating system and configure your cluster, you must configure MS DTC to work in a cluster by using the Cluster Administrator. Failure to cluster MS DTC will not block SQL Server 2005 Setup, but SQL Server 2005 application functionality may be affected if MS DTC is not properly configured. If you install Microsoft Distributed Transaction Coordinator (MS DTC) in your SQL Server group and you have other resources that are dependent on MS DTC, MS DTC will not be available if this group is offline or during a failover. By default, SQL Server requires that the MS DTC resource be put in a SQL Server group when you are running the SQL Server Failover Wizard on new installations. If MS DTC has already been clustered, SQL Server 7.0 uses the existing location and resources but SQL Server 2000 does not. Microsoft recommends that you put MS DTC in its own group with its own physical disk resource, if it is possible.

▣ **To add a dependency to a SQL Server 2005 resource**

1. Open the Cluster Administrator in Windows Administrative Tools.
2. Locate the group that contains the applicable SQL Server resource that you would like to make dependent.
3. If the resource for the disk is already in this group, go to step 4. Otherwise, locate the group that contains the disk. If that group and the group that contains SQL Server are not owned by the same node, move the group containing the resource for the disk to the node that owns the SQL Server group.

4. Bring the the SQL Server resource offline.
5. Select the SQL Server resource, open the **Properties** dialog box, and use the **Dependencies** tab to add the disk to the set of SQL Server dependencies.

▣ **See Also**

Tasks

[How to: Read a SQL Server 2005 Setup Log File](#)

Concepts

[Before Installing Failover Clustering](#)

[How to: View SQL Server 2005 Setup Log Files](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Remove a SQL Server 2005 Failover Clustered Instance (Setup)



[See Also](#)



[Configuring High Availability](#) > [Configuring Availability How-to Topics](#) > [Failover Clustering How-to Topics](#) >

Updated: 5 December 2005

Use this procedure to uninstall a Microsoft SQL Server 2005 failover clustered instance.

Important:

To update or remove a SQL Server failover cluster, you must be a local administrator with permission to logon as a service on all nodes of the failover cluster.

Before you begin

Consider the following important points before you uninstall a SQL Server 2005 failover cluster:

- On a computer running SQL Server 2005 along with previous SQL Server versions, Enterprise Manager and other programs that depend on SQL-DMO may be disabled. This may occur in the following situations:
 - Side-by-side installations of SQL Server 2005, SQL Server 2000, and/or SQL Server 7.0, where any of the versions is uninstalled.
 - Side-by-side installations of SQL Server 2000 with SQL Server 2005, where SQL Server 2000 is installed after SQL Server 2005.

This issue is due to removal of the registration for the SQL Server 2005 SQL-DMO COM library. To re-enable SQL Server 2000 Enterprise Manager and other programs with SQL-DMO dependencies, register SQL-DMO by running 'regsvr32.exe sqldmo.dll' from a command prompt.

- If SQL Server 2000 and SQL Server 2005 are installed side-by-side in a clustered environment, SQL Native Client should not be uninstalled if you uninstall SQL Server 2005. The SQL Server Resource DLL from SQL Server 2005 always uses SQL Native Client to connect to SQL Server. When SQL Server 2005 is installed, the SQL Server Cluster Resource DLL is upgraded. When SQL Server 2005 is uninstalled, the SQL Server Resource DLL is not replaced with the version required by SQL Server 2000. Leaving SQL Native Client on cluster nodes allows the SQL Server Resource DLL to continue to connect to instances of SQL Server 2000.
- If SQL Native Client is uninstalled by accident, SQL Server resources will fail to start. To reinstall SQL Native Client, run the SQL Server 2005 Setup program to install SQL Server 2005 prerequisites.
- If you uninstall a failover cluster that has more than one SQL IP cluster resource, you must remove the additional SQL IP resources using cluster administrator.

For information about command prompt syntax, see [How to: Install SQL Server 2005 from the Command Prompt](#).

▣ **To remove a SQL Server 2005 failover cluster**

1. To begin the uninstallation process, launch **Add or Remove Programs** in Control Panel.
2. Select the SQL Server 2005 component to uninstall, and then click **Remove**. This will launch the SQL Server 2005 Installation Wizard.
3. On the **Component Selection** page, select **Remove SQL Server 2005 instance components** and then click the instance name and common components to uninstall. To continue, click **Next**.
4. On the **Remote Login Information** page, provide the password for the administrator account on all nodes of the clustered instance. To continue, click **Next**.
5. On the **Confirmation** page, verify the selections for SQL Server components that will be removed. To continue with the uninstallation

process, click **Finish**.

6. The SQL Server Installation Wizard will remove SQL Server 2005 components from your computer. If Setup requires access to the SQL Server 2005 installation media, you will be prompted to insert the CD or DVD into the CD or DVD drive. The CD or DVD used to uninstall SQL Server must have the same build number as the SQL Server instance you are trying to uninstall. For network installations, Setup must have access to the network build folder.
7. The SQL Server System Configuration Checker will scan your computer's existing configuration. On the **Welcome** page, click **Next**. After the System Configuration Checker finishes its scan, click **Next** to continue.
8. The **Setup Progress** page will display status as it removes SQL Server 2005 from your computer. On the **Completing the Microsoft SQL Server Installation Wizard** page, you can view the Setup summary log by clicking the link provided on this page. To exit the SQL Server Installation Wizard, click **Finish**.

 **Note:**

Do not attempt to uninstall MSXML or SQLXML; they will be automatically removed when all dependent components are uninstalled.

▣ **See Also**

Tasks

[How to: Read a SQL Server 2005 Setup Log File](#)

[How to: Manually Uninstall a SQL Server 2005 Failover Cluster](#)

Concepts

[How to: View SQL Server 2005 Setup Log Files](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Manually Uninstall a SQL Server 2005 Failover Cluster



[See Also](#)



[Configuring High Availability](#) > [Configuring Availability How-to Topics](#) > [Failover Clustering How-to Topics](#) >

Consider the following important points before you uninstall a Microsoft SQL Server 2005 failover cluster:

- On a computer running SQL Server 2005 along with previous SQL Server versions, Enterprise Manager and other programs that depend on SQL-DMO may be disabled. This may occur in the following situations:
 - Side-by-side installations of SQL Server 2005, SQL Server 2000, and SQL Server 7.0, where any of the versions is uninstalled.
 - Side-by-side installations of SQL Server 2000 with SQL Server 2005, where SQL Server 2000 is installed after SQL Server 2005.

This issue is due to removal of the registration for the SQL Server 2005 SQL-DMO COM library. To re-enable Enterprise Manager and other programs with SQL-DMO dependencies, register SQL-DMO by running **regsvr32.exe sqlldmo.dll** from a command prompt. For more information, see [Troubleshooting an Installation of the SQL Server Database Engine](#).

- If SQL Native Client is uninstalled by accident, SQL Server resources will fail to start. To reinstall SQL Native Client, run the SQL Server 2005 Setup program to install SQL Server 2005 prerequisites.
- If you uninstall a failover cluster that has more than one SQL IP cluster resource, you must remove the additional SQL IP resources using cluster administrator.

To uninstall a SQL Server 2005 failover cluster after a failed uninstallation using the SQL Server Setup program

1. Backup the existing registry on each node before removing SQL Server 2005.

2. Using the Registry Editor utility (regedit.exe) locate the registry key HKLM\Software\Microsoft\Microsoft SQL Server*<instid>*\Setup, where *<instid>* represents the specific instance of SQL Server 2005 being uninstalled (for example, MSSQL.1 for the first instance, MSSQL.2 for the second instance, etc.). In this key set **SqlCluster=0**. This will uncluster the clustered instance of SQL Server 2005.

Caution Incorrectly editing the registry can cause serious problems that may require you to reinstall your operating system. Microsoft cannot guarantee that problems resulting from editing the registry incorrectly can be resolved. Before editing the registry, back up the registry and any valuable data.

3. Ensure that the shared disk is available to the node being removed.
4. Launch **Add or Remove Programs** in Control Panel.
5. Select the SQL Server 2005 instance to uninstall, and then click **Remove**. This will launch the SQL Server 2005 Installation Wizard.
6. On the **Component Selection** page, select **Remove SQL Server 2005 instance components** and then click the instance name and common components to uninstall. To continue, click **Next**.
7. On the **Remote Login Information** page, provide the password for the administrator account on all nodes of the clustered instance. To continue, click **Next**.
8. On the **Confirmation** page, verify the selections for SQL Server components that will be removed. To continue with the uninstallation process, click **Finish**.
9. The SQL Server 2005 Installation Wizard will remove SQL Server 2005 components from your computer. If Setup requires access to the SQL Server 2005 installation media, you will be prompted to insert the CD or DVD into the CD or DVD drive. The CD or DVD used to uninstall SQL Server must have the same build number as the SQL Server instance you are trying to uninstall. For network installations, Setup must have access to the network build folder.

10. The SQL Server 2005 System Configuration Checker will scan your computer's existing configuration. On the **Welcome** page, click **Next**. After the System Configuration Checker finishes its scan, click **Next** to continue.
11. The **Setup Progress** page will display status as it removes SQL Server 2005 from your computer. On the **Completing the Microsoft SQL Server Installation Wizard** page, you can view the Setup summary log by clicking the link provided on this page. To exit the SQL Server Installation Wizard, click **Finish**.
 1. **Note** Do not attempt to uninstall MSXML or SQLXML; they will be automatically removed when all dependent components are uninstalled.
12. Move the shared disk to the next node to be uninstalled, and on this node, repeat steps 1 - 11. Continue to backup the registry and any other important data on each node, edit the registry to uncluster the cluster, move the shared disk so that it is available to the node being removed, and then use **Add or Remove Programs** to remove SQL Server 2005 from that node.
13. Repeat this move-and-uninstall process for each remaining node.

▣ **See Also**

Tasks

[How to: Remove a SQL Server 2005 Failover Clustered Instance \(Setup\)](#)

[How to: Add or Remove Nodes in a SQL Server 2005 Failover Cluster \(Setup\)](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Recover from Failover Cluster Failure in Scenario 1



[See Also](#)



[Configuring High Availability](#) > [Configuring Availability How-to Topics](#) > [Failover Clustering How-to Topics](#) >

In this scenario, failure is caused by hardware failure in Node 1 of a two-node cluster. This hardware failure could be caused, for example, by the failure of a disk controller or the operating system.

To recover from a failover cluster hardware failure

1. After Node 1 fails, the Microsoft SQL Server failover cluster fails over to Node 2.
2. Evict Node 1 from Microsoft Cluster Service (MSCS). To evict a node from MSCS, from Node 2, open Cluster Administrator, right-click the node you want to remove, and then click **Evict Node**.
3. Verify that Node 1 has been evicted from the cluster definition.
4. Install new hardware to replace the failed hardware in Node 1.
5. Using the Cluster Administrator, configure Microsoft Cluster Server (MSCS) to add Node 1 to the existing cluster. For more information, see [Before Installing Failover Clustering](#).
6. Ensure that the administrator accounts are the same on all cluster nodes. For more information, see [How to: Create a New SQL Server 2005 Failover Cluster \(Setup\)](#).
7. Run SQL Server Setup to add Node 1 to the failover cluster. For more information, see [How to: Add or Remove Nodes in a SQL Server 2005 Failover Cluster \(Setup\)](#).

▣ **See Also**

Tasks

[How to: Read a SQL Server 2005 Setup Log File](#)

Concepts

[How to: View SQL Server 2005 Setup Log Files](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Recover from Failover Cluster Failure in Scenario 2



[See Also](#)



[Configuring High Availability](#) > [Configuring Availability How-to Topics](#) > [Failover Clustering How-to Topics](#) >

In Scenario 2, failure is caused by Node 1 being down or offline but not irretrievably broken. This could be caused by an operating system failure or a hardware failure.

▣ Procedure

▣ To recover from a failover cluster failure

1.

After Node 1 fails, the Microsoft SQL Server failover cluster fails over to Node 2.

- Resolve the problem with Node 1.
- Ensure that the Microsoft Cluster Service (MSCS) cluster is working and all nodes are online.
- Attempt to failover SQL Server to the recovered node. If this succeeds, no further action is required, with the exception of normal testing. If SQL Server fails to come online on the recovered node, see [Failover Cluster Troubleshooting](#).

▣ **See Also**

Tasks

[How to: Read a SQL Server 2005 Setup Log File](#)

Concepts

[Failover Cluster Troubleshooting](#)

[How to: Rebuild the Master Database for SQL Server 2005](#)

[How to: Rebuild the Registry for SQL Server 2005](#)

[How to: View SQL Server 2005 Setup Log Files](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Collation Options and International Support



[See Also](#)



[Preparing to Install SQL Server 2005](#) >

Microsoft SQL Server 2005 supports all languages supported by Microsoft Windows operating systems. This topic presents details about language support in SQL Server 2005, and how SQL Server supports Unicode and non-Unicode data formats.

▣ Collation Changes in SQL Server 2005

The following changes and enhancements are new in SQL Server 2005:

- Support for new and updated collations from Windows Server 2003 and Windows XP, including Far East collations that support supplementary characters.
- Support for supplementary character string comparison based on code point with new collations in SQL Server 2005.
- Addition of a binary flag for true code point comparisons: Binary-code point.
- Improved migration path from SQL Server 7.0 and SQL Server 2000 to SQL Server 2005.
- Deprecation of the Hindi and Lithuanian_Classic Windows collations and deprecation of the SQL_ALTDiction_CP1253_CS_AS SQL collation. These collations are supported but will not show up in **::fn_helpcollations()** builtin function. Support for these collations will be removed in a future SQL Server release. For more information on the SQL Server deprecation policy, see [Backward Compatibility](#).

For more information, see [Collation Settings in Setup](#)

▣ **See Also**

Reference

[Collation Settings in Setup](#)
[Selecting a SQL Collation](#)

Concepts

[Collation Terminology](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Collation Terminology



[See Also](#)



[Preparing to Install SQL Server 2005](#) > [Collation Options and International Support](#) >

To make the best use of language support in SQL Server 2005, you should understand the terms defined in this topic.

▣ Terms

- [Code page](#)
- [Collation](#)
- [Data type](#)
- [Globalization](#)
- [Locale](#)
- [Reading order](#)
- [Sort order](#)
- [Unicode](#)

Code Page

A code page is an ordered set of characters of a given script in which a numeric index, or code point value, is associated with each character. A Microsoft Windows code page is commonly referred to as a *character set* or a *charset*. Code pages are used to provide support for character sets and keyboard layouts used by different Windows locales.

Related Topics: [Setting Client Code Pages](#)

[Back to Top](#)

Collation

A collation specifies the bit patterns that represent each character in a data set. Collations also determine the rules that sort and compare data. SQL Server 2005 supports storing objects with different collations in a single database—each column in a SQL Server database can have its own collation. For non-Unicode columns, the collation setting specifies the code page for the data and, as a result, which characters can be represented. Data can be moved between Unicode columns seamlessly. Data moved between non-Unicode columns cannot be moved seamlessly, and must be converted by the current code page.

The result of a Transact-SQL statement can vary when the statement is run in the context of different databases that each has a different collation setting. Best practices include use of a standardized collation for your organization, if

possible. Using a standard collation setting across all systems in your organization will help to eliminate the need to explicitly specify the collation in every character or Unicode expression. If you must work with objects that have different collation and code page settings, you must code your queries to consider the rules of collation precedence. For more information, see **Collation Precedence (Transact-SQL)**.

The characteristics of a collation are language sensitive, case sensitive, accent sensitive, Kana sensitive, and width sensitive.

SQL Server 2005 collations include the following groupings:

Windows collations

Windows collations define rules for storing character data based on an associated Windows locale. For a Windows collation, comparison of non-Unicode data is implemented using the same algorithm as Unicode data. The base Windows collation rules specify which alphabet or language is used when dictionary sorting is applied, as well as the code page used to store non-Unicode character data. Both Unicode and non-Unicode sorting are compatible with string comparisons in a particular version of Windows. This provides consistency across data types within SQL Server, and it also provides developers with the ability to sort strings in their applications using the same rules that are used by SQL Server; that is, by calling the **CompareStringW** function of the Microsoft Win32 API. For more information, see [Collation Settings in Setup](#).

Binary collations

Binary collations sort data based on the sequence of coded values defined by the locale and data type. A binary collation in SQL Server defines the language locale and the ANSI code page to be used, enforcing a binary sort order. Binary collations are useful in achieving improved application performance due to their relative simplicity. For non-Unicode data types, data comparisons are based on the code points defined in the ANSI code page. For Unicode data types, data comparisons are based on the Unicode code points. For binary collations on Unicode data types, the locale is not considered in data sorts. For example, `Latin_1_General_BIN` and `Japanese_BIN` yield identical sorting results when used on Unicode data.

Previous binary collations in SQL Server performed an incomplete code-point to code-point comparison for Unicode data, in that binary collations in previous versions of SQL Server compared the first character as WCHAR, followed by a byte-by-byte comparison. For backwards compatibility reasons, existing binary collation semantics will not be changed.

Binary collations in this release of SQL Server include a new set of pure code-point comparison collations. Customers can choose to migrate to the new binary collations to take advantage of true code-point comparisons, and they should utilize the new binary collations for development of new applications. The new BIN2 suffix identifies collation names that implement the new code-point collation semantics. In addition, a new comparison flag is added corresponding to BIN2 for the new binary sort. For more information, see [Using Binary Collations](#).

SQL Server collations

SQL Server collations provide sort order compatibility with earlier versions of SQL Server. SQL Server collations are based on legacy SQL Server sort orders for non-Unicode data—for example, **char** and **varchar** data types—defined by SQL Server. The dictionary sorting rules for non-Unicode data are not compatible with any sorting routine provided by Windows operating systems, but sorting of Unicode data is compatible with a particular version of Windows sorting rules. Because SQL Server collations use different comparison rules for non-Unicode and Unicode data, you may see different results for comparisons of the same data, depending on the underlying data type. For more information, see [Using SQL Collations](#).

Note:

When you upgrade an instance of SQL Server, SQL Server collations can be specified for compatibility with existing instances of SQL Server. Because the default collation for an instance of SQL Server is defined during Setup, it is important to specify collation settings carefully when:

- Your application code depends in some way on the behavior of previous SQL Server collations.
- You are going to use SQL Server 2005 replication with existing installations of SQL Server 6.5 or SQL Server 7.0.

- You must store character data that reflects multiple languages.

SQL Server 2005 supports setting collations at the following levels of a SQL Server 2005 instance:

Server-level collations

The default collation of a SQL Server instance is set during Setup. The default collation of the instance also becomes the default collation of the system databases: **master**, **model**, **tempdb**, **msdb**, and **distribution**. After a collation has been assigned to any object other than a column or a database, you cannot change the collation except by dropping and re-creating the object. Instead of changing the default collation of a SQL Server instance, you can specify the collation at the time you create a new database or database column.

To query the server collation for a SQL Server instance, use the following Transact-SQL SERVERPROPERTY function:

 [Copy Code](#)

```
SELECT CONVERT (varchar, SERVERPROPERTY('collation'))
```

To query the server for all available collations, use the following **fn_helpcollations()** built-in function:

 [Copy Code](#)

```
SELECT * from ::fn_helpcollations()
```

Database-level collations

When a database is created, the COLLATE clause of the CREATE DATABASE statement can be used to specify the default collation of the database. If no collation is specified during database creation, the database is assigned the default collation of the **model** database. The default collation for the **model** database is the same as the default collation of the SQL Server instance.

The collation of a user database can be changed with an ALTER DATABASE statement like the following one:

 [Copy Code](#)

```
ALTER DATABASE myDB COLLATE Greek_CS_AI
```

The current collation of a database can be retrieved by using a statement like the following one:

```
SELECT CONVERT (varchar, DATABASEPROPERTYEX('database_name', 'col
```

 **Note:**

Altering the database-level collation does not affect user-, table-, or column-level collations.

Column-level collations

When creating a table, collations for each character-string column can be specified using the COLLATE clause of the CREATE TABLE statement. If no collation is specified during table creation, the column is assigned the default collation of the database.

The collation of a column can be changed with the ALTER TABLE statement like the following one:



```
ALTER TABLE myTable ALTER COLUMN mycol NVARCHAR(10) COLLATE Gree
```

Expression-level collations

Expression-level collations are set at the time a statement is run, and they affect the way a result set is returned. This allows sorting of a result so that the ORDER BY clause can be language-specific. Use a COLLATE clause like the following one to implement expression-level collations:

 [Cop](#)

```
SELECT name FROM customer ORDER BY name COLLATE Latin1_General_C
```

[Back to Top](#)

Data Type

Data type is a definition that specifies a range of values, the operations that can be performed on the values, and the way that the values are stored in memory on your computer. Defining data types enables SQL Server to manipulate data in predictable ways. Non-Unicode character data types are **char**, **varchar**, and **text**. Unicode data types use the Unicode character representation; they are **nchar**, **nvarchar**, and **ntext**. Using Unicode data types in your applications is recommended, especially if you store character data that reflects multiple languages.

Related Topics: Data Types (Database Engine), Data Types (Transact-SQL), Integration Services Data Types

[Back to Top](#)

Globalization

Globalization is the process of developing a software application with features and code design that consider multiple spoken languages and locales. Globalized application design accommodates multiple locales and Unicode-supported languages for data input, processing, display, and output.

Related Topics: SQL Server Globalization

[Back to Top](#)

Locale

A locale is a set of information associated with a place or a culture—the name and identifier of the spoken language, the script used to write the language, and cultural conventions. SQL Server 2005 supports all 135 locales supported by Windows XP. Among them are five Chinese language locales (Hong Kong SAR, Macau SAR, the People's Republic of China, Singapore, and Taiwan); thirteen English language locales (Australia, Belize, Canada, Caribbean, Ireland, Jamaica, New Zealand, Philippines, South Africa, Trinidad, the United Kingdom, the United States of America, and Zimbabwe); and six French language locales (Belgium, Canada, France, Luxembourg, Monaco, and Switzerland).

The following table illustrates several differences between four common locales

supported by Windows.

Locale	English (U.S.A.)	French (France)	Japanese	United Arab Emirates
Country/region	United States	France	Japan	United Arab Emirates
Language	English	French	Japanese	Arabic
Written scripts	Latin	Latin	Kana, kanji	Arabic
Reading order	Left to right	Left to right	Left to right	Right to left
Windows-defined code page	1252	1252	932	1256
Time format	1:00 pm	13:00	13:00	1:00 p
Calendar	Gregorian	Gregorian	Gregorian (Localized)	Gregorian (Localized)
Default paper size	U.S. Letter	A4	A4	A4
Decimal separator	.	,	.	,
List separator	,	;	,	;
Thousands separator	,	space	,	,

[Back to Top](#)

Reading Order

Reading order is the overall direction of an ordered sequence of text, relating to the word order, not the order of the entered characters. For example, using Arabic as the keyboard language means that new characters will always flow from right to left. When Latin is the keyboard language, new characters flow from left to right.

[Back to Top](#)

Sort Order

Sort order specifies the way that data values are sorted, affecting the results of data comparison. The sorting of data is accomplished through collations, and it can be optimized using indexes.

Related Topics: [Windows Collation Sorting Styles](#), **Indexes**

[Back to Top](#)

Unicode

Unicode represents the characters of a language with two bytes rather than one, enabling a single Unicode character set to represent almost all of the written languages of the world. Unicode was developed, and is maintained and promoted by the Unicode Consortium, a nonprofit computer industry organization. For more information, see the [Unicode Consortium Web site](#).

If you store character data that reflect multiple languages, always use Unicode data types (**nchar**, **nvarchar**, and **ntext**) instead of the non-Unicode data types (**char**, **varchar**, and **text**). You may experience a significant performance gain through use of Unicode because fewer code-page conversions will be required. There are significant limitations associated with non-Unicode data types, because a non-Unicode computer will be limited to use of a single code page. To completely evaluate issues related to use of Unicode or non-Unicode data types, you must test your scenario to measure performance differences in your particular environment. At a minimum, standardize the site collation, and deploy Unicode servers and clients wherever possible.

In most cases, your SQL Server instance will interact with other servers or clients, and your instance may use multiple data access standards. SQL Server clients are one of two main types:

- **Unicode clients** that use OLE DB and Open Database Connectivity (ODBC) versions 3.7 or later.
- **Non-Unicode clients** that use DB-Library and ODBC versions 3.6 or earlier.

The following table presents considerations for use of multilingual data with various combinations of Unicode and non-Unicode servers.

Server	Client	Benefits or Limitations
		An ideal configuration because Unicode data will be used throughout the system. This scenario provides the best

Unicode	Unicode	performance and protection from corruption of retrieved data. This is the case with Microsoft ActiveX Data Objects (ADO), OLE DB, and ODBC versions 3.7 or later.
Unicode	Non-Unicode	In this scenario, data storage may not be a problem, but there may be limitations when you move data to a client computer. At a minimum, Unicode data will have to be converted using the code page on the non-Unicode client.
Non-Unicode	Unicode	This is not an ideal configuration for use of multilingual data. You will not be able to write Unicode data to the non-Unicode server. Problems are likely to occur when data is sent to servers that are outside the server's code page.
Non-Unicode	Non-Unicode	This is a very limiting scenario for multilingual data. You will be limited to use of a single code page. The ideal configuration is a Unicode server with Unicode clients.

Related Topics: Unicode Basics

[Back to Top](#)

▣ **See Also**

Reference

[Collation Options and International Support](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Collation Settings in Setup

 [See Also](#)



[Preparing to Install SQL Server 2005](#) > [Collation Options and International Support](#) >

Updated: **14 April 2006**

Collation settings, which include character set, sort order, and other locale-specific settings, are fundamental to the structure and function of Microsoft SQL Server databases. Within your organization, you should develop a standard for collation settings, and apply these settings at the time that you install SQL Server. Many server-to-server activities can fail or yield inconsistent results if collation settings are not consistent across servers. Select a Microsoft Windows locale to match the collation settings in other instances of SQL Server 2005; or select **SQL Collations** to match settings with the sort orders in earlier versions of SQL Server.

SQL Server 2005 supports setting collations at the following levels of a SQL Server 2005 instance:

- Server level
- Database level
- Column level
- Expression level

For more information on collation levels, see [Collation Terminology](#). For more information on rebuilding system databases to specify a new system collation, see [How to: Install SQL Server 2005 from the Command Prompt](#).

▣ Windows system locale

Change the default settings for Windows collation only if your installation of SQL Server must match the collation settings used by another instance of SQL Server, or if the collations setting must match the Windows system locale of another computer.

Collation Designator

Select the name of a specific Windows collation from the **Collation Designator** list. For example:

- Use **Latin1_General** for the U.S. English character set (code page 1252).
- Use **Modern_Spanish** for all variations of Spanish that use the same character set as U.S. English (code page 1252).
- Use **Arabic** for all variations of Arabic, which use the Arabic character set (code page 1256).
- Use **Japanese_Unicode** for the Unicode version of Japanese (code page 932), which has a different sort order from **Japanese**, but the same code page (932).

Sort Order

Select sort order options to use with the selected collation designator. Binary is the fastest sorting order, and it is case sensitive, but it can yield unexpected sort orders. If **Binary** is selected, the **Case-sensitive**, **Accent-sensitive**, **Kana-sensitive**, and **Width-sensitive** options are not available. For more information, see [Windows Collation Sorting Styles](#).

▣ **Binary Collations**

Binary collations sort and compare data in SQL Server based on the bit pattern for each character. Each binary collation in SQL Server maps to a specific language locale and ANSI code page, and each performs case-sensitive and accent-sensitive data sorts. Binary collations provide the fastest data sorts. For more information, see [Windows Collation Sorting Styles](#) and [Using Binary Collations](#).

▣ **SQL Collations**

The **SQL Collations** option is used for compatibility with earlier versions of SQL Server. Select this option to match settings with SQL Server 2000, SQL Server 7.0, or earlier. For more information, see [Using SQL Collations](#).

▣ New Collation Versions

The following improvements have been made to improve several collations over previous collation behavior:

- Supplementary character comparison support for Far East collations
- Addition of a new Indic_General_90_CI_AS (Unicode only) collation version to support changes to sorting behavior

▣ **Important:**

Both of these version updates significantly impact sort and comparison behavior. During upgrade of these collations to Microsoft SQL Server 2005, migration to new versions will not be forced. If users migrate to new collations in order to support supplementary characters, all databases and SQL Server objects will have to be reindexed. Because the reindex process may be time consuming, consider the cost-benefit of supplementary character support. Collations from SQL Server 2000 will be supported in SQL Server 2005 in cases where customers choose not to migrate to new collations.

The following collations have been updated for this release of SQL Server 2005 to use the Windows Server 2003 code-point sorting behavior.

Old collation name	New collation name
Japanese	Japanese_90
Chinese	Chinese_PRC_90
Chinese_PRC_Stroke	Chinese_PRC_Stroke_90
Chinese_Taiwan_Bopomofo	Chinese_Taiwan_Bopomofo_90
Chinese_Taiwan_Stroke	Chinese_Taiwan_Stroke_90
Korean	Korean_90
Hindi (deprecated in this release)	Indic_General_90_CI_AS (Unicode only)

▣ Default Collations in SQL Server Setup

In Control Panel, find the Microsoft Windows locale name in **Regional Options** in (Windows 2000) or **Regional and Language Options** (Windows XP), and then use the following table to find the corresponding collation designator and code page to match collation settings with an existing Windows locale in the table of SQL Server 2005 collations below.

Windows System Locale	LCID (Locale ID)	Default SQL Collation	Code page
Afrikaans	0x436	Latin1_General_CI_AS	1252
Albanian	0x41c	Albanian_CI_AS	1250
Arabic (Algeria)	0x1401	Arabic_CI_AS	1256
Arabic (Bahrain)	0x3c01	Arabic_CI_AS	1256
Arabic (Egypt)	0xc01	Arabic_CI_AS	1256
Arabic (Iraq)	0x801	Arabic_CI_AS	1256
Arabic (Jordan)	0x2c01	Arabic_CI_AS	1256
Arabic (Kuwait)	0x3401	Arabic_CI_AS	1256
Arabic (Lebanon)	0x3001	Arabic_CI_AS	1256
Arabic (Libya)	0x1001	Arabic_CI_AS	1256
Arabic (Morocco)	0x1801	Arabic_CI_AS	1256
Arabic (Oman)	0x2001	Arabic_CI_AS	1256
Arabic (Qatar)	0x4001	Arabic_CI_AS	1256
Arabic (Saudi Arabia)	0x401	Arabic_CI_AS	1256
Arabic (Syria)	0x2801	Arabic_CI_AS	1256
Arabic (Tunisia)	0x1c01	Arabic_CI_AS	1256
Arabic (U.A.E.)	0x3801	Arabic_CI_AS	1256
Arabic (Yemen)	0x2401	Arabic_CI_AS	1256
Armenian (Republic of Armenia)	0x42b	Latin1_General_CI_AS	Unicode
Azeri-Cyrillic (Azerbaijan) ¹	0x82c	Azeri_Cyrillic_90_CI_AS	1251

Azeri-Latin (Azerbaijan) ¹	0x42c	Azeri_Latin_90_CI_AS	1254
Basque	0x42d	Latin1_General_CI_AS	1252
Belarusian	0x423	Cyrillic_General_CI_AS	1251
Bulgarian	0x402	Cyrillic_General_CI_AS	1251
Catalan	0x403	Latin1_General_CI_AS	1252
Chinese (Hong Kong S.A.R.) ¹	0xc04	Chinese_Hong_Kong_Stroke_90_CI_AS	950
Chinese (Macau S.A.R.) ²	0x1404	Chinese_PRC_90_CI_AS	950
Chinese (PRC) ²	0x804	Chinese_PRC_CI_AS	936
Chinese (PRC)	0x20804	Chinese_PRC_Stroke_CI_AS	936
Chinese (Singapore)	0x1004	Chinese_PRC_90_CI_AS	936
Chinese (Taiwan)	0x404	Chinese_Taiwan_Stroke_CI_AS	950
Chinese (Taiwan)	0x30404	Chinese_Taiwan_Bopomofo_CI_AS	950
Croatian	0x41a	Croatian_CI_AS	1250
Czech	0x405	Czech_CI_AS	1250
Danish	0x406	Danish_Norwegian_CI_AS	1252
Divehi ¹	0x465	Divehi_90_CI_AS (Unicode only)	Unicode
Dutch (Belgium)	0x813	Latin1_General_CI_AS	1252
Dutch (Netherlands)	0x413	Latin1_General_CI_AS	1252
English (Australia)	0xc09	Latin1_General_CI_AS	1252
English (Belize)	0x2809	Latin1_General_CI_AS	1252
English (Canada)	0x1009	Latin1_General_CI_AS	1252
English (Caribbean)	0x2409	Latin1_General_CI_AS	1252
English (Hong Kong S.A.R.)	0x3c09	Latin1_General_CI_AS	1252
English (India)	0x4009	Latin1_General_CI_AS	1252
English (Indonesia)	0x3809	Latin1_General_CI_AS	1252
English (Ireland)	0x1809	Latin1_General_CI_AS	1252
English (Jamaica)	0x2009	Latin1_General_CI_AS	1252
English (Malaysia)	0x4409	Latin1_General_CI_AS	1252

English (New Zealand)	0x1409	Latin1_General_CI_AS	1252
English (Philippines)	0x3409	Latin1_General_CI_AS	1252
English (Singapore)	0x4809	Latin1_General_CI_AS	1252
English (South Africa)	0x1c09	Latin1_General_CI_AS	1252
English (Trinidad)	0x2c09	Latin1_General_CI_AS	1252
English (United Kingdom)	0x809	Latin1_General_CI_AS	1252
English (United States)	0x409	SQL_Latin1_General_CP1_CI_AS	1252
English (Zimbabwe)	0x3009	Latin1_General_CI_AS	1252
Estonian	0x425	Estonian_CI_AS	1257
Faeroese	0x438	Latin1_General_CI_AS	1252
Farsi	0x429	Arabic_CI_AS	1256
Finnish	0x40b	Finnish_Swedish_CI_AS	1252
French (Belgium)	0x80c	French_CI_AS	1252
French (Canada)	0xc0c	French_CI_AS	1252
French (France)	0x40c	French_CI_AS	1252
French (Luxembourg)	0x140c	French_CI_AS	1252
French (Monaco)	0x180c	French_CI_AS	1252
French (Switzerland)	0x100c	French_CI_AS	1252
Galician (Spain)	0x456	Latin1_General_CI_AS	1252
Georgian	0x437	Latin1_General_CI_AS	Unicode
German (Austria)	0xc07	Latin1_General_CI_AS	1252
German (Germany)	0x407	Latin1_General_CI_AS	1252
German (Liechtenstein)	0x1407	Latin1_General_CI_AS	1252
German (Luxembourg)	0x1007	Latin1_General_CI_AS	1252
German (Phone Book Sort)	0x10407	German_PhoneBook_CI_AS	1252

German (Switzerland)	0x807	Latin1_General_CI_AS	1252
Greek	0x408	Greek_CI_AS	1253
Gujarati (India) ³	0x447	Indic_General_90_CI_AS (Unicode only)	Unicode
Hebrew	0x40d	Hebrew_CI_AS	1255
Hindi ³	0x0439	Indic_General_90_CI_AS (Unicode only)	Unicode
Hungarian	0x40e	Hungarian_CI_AS	1250
Hungarian (Technical)	0x104e	Hungarian_Technical_CI_AS	1250
Icelandic	0x40f	Icelandic_CI_AS	1252
Indonesian	0x421	Latin1_General_CI_AS	1252
Italian (Italy)	0x410	Latin1_General_CI_AS	1252
Italian (Switzerland)	0x810	Latin1_General_CI_AS	1252
Japanese	0x411	Japanese_CI_AS	932
Japanese (Unicode)	0x10411	Japanese_Unicode_CI_AS	932
Kannada (India) ³	0x44b	Indic_General_90_CI_AS (Unicode only)	Unicode
Kazakh (Kazakstan) ¹	0x43f	Kazakh_90_CI_AS	1251
Konkani (India) ³	0x457	Indic_General_90_CI_AS (Unicode only)	Unicode
Korean (Extended Wansung)	0x0412	Korean_Wansung_CI_AS	949
Kyrgyz-Cyrillic (Republic of Kyrgyztan)	0x440	Cyrillic_General_CI_AS	1251
Latvian	0x426	Latvian_CI_AS	1257
Lithuanian	0x427	Lithuanian_CI_AS	1257
Macedonian (FYROM)	0x42f	Macedonian_FYROM_90_CI_AS	1251
Malay (Brunei Darussalam)	0x83e	Latin1_General_CI_AS	1252
Malay (Malaysia)	0x43e	Latin1_General_CI_AS	1252

Marathi (India) ³	0x44e	Indic_General_90_CI_AS (Unicode only)	Unicode
Mongolian-Cyrillic (Mongolia)	0x450	Cyrillic_General_CI_AS	1251
Norwegian (Bokmal)	0x414	Danish_Norwegian_CI_AS	1252
Norwegian (Nynorsk)	0x814	Danish_Norwegian_CI_AS	1252
Polish	0x415	Polish_CI_AS	1250
Portuguese (Brazil)	0x416	Latin1_General_CI_AS	1252
Portuguese (Portugal)	0x816	Latin1_General_CI_AS	1252
Punjab-Gurmuhki (India) ³	0x446	Indic_General_90_CI_AS (Unicode only)	Unicode
Romanian	0x418	Romanian_CI_AS	1250
Russian	0x419	Cyrillic_General_CI_AS	1251
Sanskrit (India) ³	0x44f	Indic_General_90_CI_AS (Unicode only)	Unicode
Serbian (Cyrillic)	0xc1a	Cyrillic_General_CI_AS	1251
Serbian (Latin)	0x81a	Cyrillic_General_CI_AS	1250
Slovak	0x41b	Slovak_CI_AS	1250
Slovenian	0x424	Slovenian_CI_AS	1250
Spanish (Argentina)	0x2c0a	Modern_Spanish_CI_AS	1252
Spanish (Bolivia)	0x400a	Modern_Spanish_CI_AS	1252
Spanish (Chile)	0x340a	Modern_Spanish_CI_AS	1252
Spanish (Colombia)	0x240a	Modern_Spanish_CI_AS	1252
Spanish (Costa Rica)	0x140a	Modern_Spanish_CI_AS	1252
Spanish (Dominican Republic)	0x1c0a	Modern_Spanish_CI_AS	1252
Spanish (Ecuador)	0x300a	Modern_Spanish_CI_AS	1252
Spanish (El Salvador)	0x440a	Modern_Spanish_CI_AS	1252

Spanish (Guatemala)	0x100a	Modern_Spanish_CI_AS	1252
Spanish (Honduras)	0x480a	Modern_Spanish_CI_AS	1252
Spanish (International Sort)	0xc0a	Modern_Spanish_CI_AS	1252
Spanish (Mexico)	0x80a	Modern_Spanish_CI_AS	1252
Spanish (Nicaragua)	0x4c0a	Modern_Spanish_CI_AS	1252
Spanish (Panama)	0x180a	Modern_Spanish_CI_AS	1252
Spanish (Paraguay)	0x3c0a	Modern_Spanish_CI_AS	1252
Spanish (Peru)	0x280a	Modern_Spanish_CI_AS	1252
Spanish (Puerto Rico)	0x500a	Modern_Spanish_CI_AS	1252
Spanish (Traditional Sort)	0x40a	Traditional_Spanish_CI_AS	1252
Spanish (Uruguay)	0x380a	Modern_Spanish_CI_AS	1252
Spanish (Venezuela)	0x200a	Modern_Spanish_CI_AS	1252
Swahili	0x441	Latin1_General_CI_AS	1252
Swedish	0x41d	Finnish_Swedish_CI_AS	1252
Swedish (Finland)	0x81d	Finnish_Swedish_CI_AS	1252
Syriac (Syria) ¹	0x45a	Syriac_90_CI_AS (Unicode only)	Unicode
Tamil (India) ¹	0x449	Indic_General_90_CI_AS (Unicode only)	Unicode
Tatar (Tatarstan) ¹	0x444	Tatar_90_CI_AS	1251
Telugu (India) ¹	0x44a	Indic_General_90_CI_AS (Unicode only)	Unicode
Thai	0x41e	Thai_CI_AS	874
Turkish	0x41f	Turkish_CI_AS	1254
Ukrainian	0x422	Ukrainian_CI_AS	1251
Urdu	0x420	Arabic_CI_AS	1256
Uzbek (Latin) ¹	0x443	Uzbek_Latin_90_CI_AS	1254
Uzbek-Cyrillic (Republic of Uzbekistan)	0x843	Cyrillic_General_CI_AS	1251

Vietnamese	0x42a	Vietnamese_CI_AS	1258
------------	-------	------------------	------

¹New collations for SQL Server 2005; these collations use the Windows 2003 sorting tables and, therefore, support supplementary characters. Both BIN and BIN2 binary sorts are supported for the new 90 Windows collations.

²Use Pinyin or Pronunciation for sorting.

³Major change to sorting behavior; use the new default collation, Indic_General_90_CI_AS (Unicode only).

⁴ The Hindi collation is deprecated in SQL Server 2005 because the Windows 2000 sorting table is used in this SQL Server release. The collation still exists in the server, but it will not be supported in a future SQL Server release, and it does not show up in **::fn_helpcollations()**.

⁵ The Hindi and Lithuanian_Classic collations are deprecated in SQL Server 2005. These collations still exist in the server, but they will not be supported in a future SQL Server release, and they do not show up in **::fn_helpcollations()**.

▣ Change History

Release	History
5 December 2005	Changed content: <ul style="list-style-type: none"><li data-bbox="565 499 1386 583">• Documented support for Macedonian_FYROM_90 collation.

▣ **See Also**

Reference

[Collation Options and International Support](#)

[Collation Settings](#)

[Windows Collation Sorting Styles](#)

Concepts

[Using SQL Collations](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Windows Collation Sorting Styles

 [See Also](#)



[Preparing to Install SQL Server 2005](#) > [Collation Options and International Support](#) >

During SQL Server Setup, you will be prompted to select either Windows collations or binary collations. Your choice of collations affects the data comparison and sort order behaviors of your instance of Microsoft SQL Server. This topic provides considerations for choosing Microsoft Windows collations or binary collations for your SQL Server implementation.

Note:

For Windows collations, the **nchar**, **nvarchar**, and **ntext** Unicode data types have the same sorting behavior as **char**, **varchar**, and **text** non-Unicode data types.

Windows collations define rules for storing character data based on an associated Windows locale. The base Windows collation rules specify which alphabet or language is used when dictionary sorting is applied, as well as the code page used to store non-Unicode character data. Binary collations sort data based on the sequence of coded values defined by the locale and data type. A binary collation in SQL Server defines the language locale and the ANSI code page to be used, enforcing a binary sort order. Binary collations are useful in achieving improved application performance due to their relative simplicity. For non-Unicode data types, data comparisons are based on the code points defined in the ANSI code page. For Unicode data types, data comparisons are based on the Unicode code points. For binary collations on Unicode data types, the locale is not considered in data sorts. For example, `Latin_1_General_BIN` and `Japanese_BIN` yield identical sorting results when used on Unicode data.

Previous binary collations in SQL Server performed an incomplete code-point-to-code-point comparison for Unicode data, in that older SQL Server binary collations compared the first character as `WCHAR`, followed by a byte-by-byte comparison. For backward compatibility reasons, existing binary collation

semantics will not be changed.

Binary collations in this release of SQL Server also include a new set of pure code-point comparison collations. Customers can choose to migrate to the new binary collations to take advantage of true code-point comparisons, and they should utilize the new binary collations for development of new applications. The new BIN2 suffix identifies collation names that implement the new code-point collation semantics. In addition, a new comparison flag is added corresponding to BIN2 for the new binary sort. For more information, see [Using Binary Collations](#).

The following table describes Windows collation sort order options for SQL Server 2005.

Sort order (suffix)	Sort order description
Binary (_BIN) ¹	<p>Sorts and compares data in SQL Server tables based on the bit patterns defined for each character. Binary sort order is case sensitive and accent sensitive. Binary is also the fastest sorting order. For more information, see Using Binary Collations.</p> <p>If this option is not selected, SQL Server follows sorting and comparison rules as defined in dictionaries for the associated language or alphabet.</p>
Binary-code point (_BIN2) ¹	<p>Sorts and compares data in SQL Server tables based on Unicode code points for Unicode data. For non-Unicode data, Binary-code point will use comparisons identical to binary sorts.</p> <p>The advantage of using a Binary-code point sort order is that no data resorting is required in applications that compare sorted SQL Server data. As a result, a Binary-code point sort order provides simpler application development and possible performance increases. For more information, see Using Binary Collations.</p>

Case-sensitive (_CS)	<p>Distinguishes between uppercase and lowercase letters. If selected, lowercase letters sort ahead of their uppercase versions.</p> <p>If this option is not selected, SQL Server considers the uppercase and lowercase versions of letters to be identical for sorting purposes.</p>
Accent-sensitive (_AS)	<p>Distinguishes between accented and unaccented characters. For example, 'a' is not equal to 'ã'.</p> <p>If this option is not selected, SQL Server considers the accented and unaccented versions of letters to be identical for sorting purposes.</p>
Kana-sensitive (_KS)	<p>Distinguishes between the two types of Japanese kana characters: Hiragana and Katakana.</p> <p>If this option is not selected, SQL Server considers Hiragana and Katakana characters to be equal for sorting purposes.</p>
Width-sensitive (_WS)	<p>Distinguishes between a single-byte character and the same character when represented as a double-byte character.</p> <p>If this option is not selected, SQL Server considers the single-byte and double-byte representation of the same character to be identical for sorting purposes.</p>

¹ If selected, the case-sensitive, accent-sensitive, kana-sensitive, and width-sensitive options are not available.

▣ Examples of Windows Collations

Each Windows collation is combined as a series of suffixes to define case, accent, width, or kana sensitivity. The following examples describe sort order behavior for various combinations of suffixes.

Windows collation suffix	Sort order description
_BIN ¹	Binary sort.
_BIN2 ¹	Binary-code point sort order, new in SQL Server 2005.
_CI_AI	Case-insensitive, accent-insensitive, kana-insensitive, width-insensitive.
_CI_AI_KS	Case-insensitive, accent-insensitive, kana-sensitive, width-insensitive
_CI_AI_KS_WS	Case-insensitive, accent-insensitive, kana-sensitive, width-sensitive
_CI_AI_WS	Case-insensitive, accent-insensitive, kana-insensitive, width-sensitive
_CI_AS	Case-insensitive, accent-sensitive, kana-insensitive, width-insensitive
_CI_AS_KS	Case-insensitive, accent-sensitive, kana-sensitive, width-insensitive
_CI_AS_KS_WS	Case-insensitive, accent-sensitive, kana-sensitive, width-sensitive
_CI_AS_WS	Case-insensitive, accent-sensitive, kana-insensitive, width-sensitive
_CS_AI	Case-sensitive, accent-insensitive, kana-insensitive, width-insensitive
_CS_AI_KS	Case-sensitive, accent-insensitive, kana-sensitive, width-insensitive
_CS_AI_KS_WS	Case-sensitive, accent-insensitive, kana-sensitive, width-sensitive
_CS_AI_WS	Case-sensitive, accent-insensitive, kana-insensitive, width-sensitive
_CS_AS	Case-sensitive, accent-sensitive, kana-insensitive, width-

	insensitive
_CS_AS_KS	Case-sensitive, accent-sensitive, kana-sensitive, width-insensitive
_CS_AS_KS_WS	Case-sensitive, accent-sensitive, kana-sensitive, width-sensitive
_CS_AS_WS	Case-sensitive, accent-sensitive, kana-insensitive, width-sensitive

¹ If BIN or Binary-code point is selected, the Case-sensitive, Accent-sensitive, Kana-sensitive, and Width-sensitive options are not available.

▣ **See Also**

Reference

[Collation Settings in Setup](#)

Concepts

[Using Binary Collations](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Using Binary Collations



[See Also](#)



[Preparing to Install SQL Server 2005](#) > [Collation Options and International Support](#) >

The following considerations will help you to decide whether old or new binary collations are appropriate for your Microsoft SQL Server implementation. Support for both BIN and BIN2 collations will continue in future SQL Server releases.

Binary collations sort data based on the sequence of coded values defined by the locale and data type. A binary collation in SQL Server defines the language locale and the ANSI code page to be used, enforcing a binary sort order. Binary collations are useful in achieving improved application performance due to their relative simplicity. For non-Unicode data types, data comparisons are based on the code points defined in the ANSI code page. For Unicode data types, data comparisons are based on the Unicode code points. For binary collations on Unicode data types, the locale is not considered in data sorts. For example, `Latin_1_General_BIN` and `Japanese_BIN` yield identical sorting results when used on Unicode data.

Previous binary collations in SQL Server performed an incomplete code-point-to-code-point comparison for Unicode data. Older SQL Server binary collations compared the first character as `WCHAR`, followed by a byte-by-byte comparison. For backward compatibility reasons, existing binary collation semantics will not be changed.

▣ **Guidelines for Using Binary Collations**

If your Microsoft SQL Server 2005 applications interact with older versions of SQL Server that use binary collations, continue to use binary. Binary collations may be a more suitable choice for mixed environments.

▣ **Guidelines for Using BIN2 Collations**

Binary collations in this release of SQL Server include a new set of pure code-point comparison collations. Customers can choose to migrate to the new binary collations to take advantage of true code-point comparisons, and they should utilize the new binary collations for development of new applications. The new BIN2 suffix identifies collation names that implement the new code-point collation semantics. In addition, a new comparison flag is added corresponding to BIN2 for the new binary sort. Advantages include simpler application development and clearer semantics.

▣ **See Also**

Reference

[Collation Options and International Support](#)
[Collation Settings in Setup](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Using SQL Collations



[See Also](#)



[Preparing to Install SQL Server 2005](#) > [Collation Options and International Support](#) >

Microsoft SQL Server collation settings depend on the type of installation. In general, choose a SQL Server collation that supports the Microsoft Windows system locale most commonly used at your site. To identify your Windows system locale on computers running Microsoft Windows 2000, click **Regional Settings** in Control Panel, and then click the **General** tab if necessary to display the current system locale. To identify your Windows system locale on computers running Windows Server 2003, click **Regional and Language Options** in Control Panel, and then click the **Regional Options** tab to display the current system locale. In most cases, a computer runs the Windows system locale that matches the language requirements of the user, so Microsoft SQL Server Setup automatically detects the Windows system locale and chooses the appropriate collation.

SQL Server collations control:

- The code page used for storing non-Unicode data in SQL Server.
- The rules governing how SQL Server sorts and compares characters stored in non-Unicode data types. SQL Server Setup will detect the Windows collation on the computer where SQL Server is being installed and automatically select the collation for your SQL Server instance. Sorts and comparisons may be different for Unicode and non-Unicode characters.

Choose a SQL Server collation if:

- You use the replication feature with existing instances of SQL Server version 6.5, SQL Server version 7.0, or SQL Server 2000.
- Your application code depends on the behaviors of previous SQL Server collations.

Use the following table to determine if you need to make a collation choice, and if so, which collation you should choose.

 **Note:**

Upgrading SQL Server 7.0 or SQL Server 2000 to SQL Server 2005 keeps the previous SQL Server collation settings; no collation choice is required.

Task	Collation
Install on a new system with no requirements for synchronizing with any existing system	<p>Use the collation identified by Setup, and then choose the desired binary, case, or other option.</p> <p>When SQL Server Setup detects that the computer is running the U.S. English system locale, Setup automatically selects the SQL Server collation: Dictionary order, case-insensitive, for use with 1252 character set.</p> <p>To select the equivalent Windows collation, select Collation designator, select Latin1_General, clear case-sensitive, and select accent-sensitive.</p>
Upgrade an installation of SQL Server 6.5 or SQL Server 7.0 to a default instance of SQL Server 2005, or to install a default instance of SQL Server 2005 that will version switch with an installation of SQL Server 6.5	<p>Use the SQL Server collation chosen by Setup.</p>
	<p>Select SERVERPROPERTY(N'Collation') from the Code Editor on the existing instance, and specify <i>Collation</i>. For more information, see</p>

<p>Synchronize (for example, to replicate) with an existing instance of SQL Server 2005</p>	<p>SERVERPROPERTY. If the collation name of the existing instance starts with "SQL", select the same SQL Server collation in Setup. If the collation name of the existing instance does not start with "SQL", the collation name refers to a Windows collation name and consists of the collation designator name followed by a description of what binary, case, accent, kana, and width-sensitivity options are specified. Select the same Windows collation designator and sorting options in SQL Server Setup.</p> <p>For more information on collation behaviors, see Windows Collation Sorting Styles.</p>
<p>Synchronize with an existing installation of SQL Server 6.5, SQL Server 7.0, or SQL Server 2000</p>	<p>Run sp_helpsort on the existing system, and then use the sort ID to select a SQL Server collation to make your instance of SQL Server 2005 compatible with an existing installation.</p> <p>For more information, see Selecting a SQL Collation.</p>
<p>Synchronize with the Windows system locale of another computer</p>	<p>In Control Panel, find the locale name under Regional Options (Windows 2000, or Windows 2003), or Regional and Language Options (Windows XP), and then use the table provided in the topic Collation Settings in Setup. Set the sorting options, as explained in the Books Online topic Windows Collation Sorting Styles.</p>

 **Note:**

When you perform an action that depends on collations, the SQL Server collation used by the referenced object must use a code page supported by the operating system running on the computer.

▣ **See Also**

Reference

[Selecting a SQL Collation](#)

[Setting Client Code Pages](#)

Other Resources

SQL Collation Name

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Selecting a SQL Collation

 [See Also](#)



[Preparing to Install SQL Server 2005](#) > [Collation Options and International Support](#) > [Using SQL Collations](#) >

This topic presents SQL Server collation options for Microsoft SQL Server instances that require compatibility with earlier versions of SQL Server that do not use collations.

The following SQL Server collations are listed on the **Collation Settings** page of the SQL Server Installation Wizard.

SQL sort order ID	Sort order name	Description	SQL collation
31	diction.437	Dictionary order, case-sensitive	SQL_Latin1_General_Cp437_CS
32	nocase.437	Dictionary order, case-insensitive	SQL_Latin1_General_Cp437_CI
33	nocasepr.437	Dictionary order, case-insensitive, uppercase preference	SQL_Latin1_General_Pref_Cp43
34	noaccent.437	Dictionary order, case-insensitive, accent-insensitive	SQL_Latin1_General_Cp437_CI
40	cp437bin.437	Binary	SQL_Latin1_General_Cp437_BI
40	cp850bin.850	Binary	SQL_Latin1_General_Cp850_BI
41	diction.850	Dictionary order, case-sensitive	SQL_Latin1_General_Cp850_CS
42	nocase.850	Dictionary order, case-insensitive	SQL_Latin1_General_Cp850_CI
		Dictionary order, case-	

44	noaccent.850	insensitive, accent-insensitive	SQL_Latin1_General_Cp850_CI
44	nocase34.850	Dictionary order, case-insensitive, uppercase preference	SQL_Latin1_General_Pref_Cp85
49	nocase34.850	Strict compatibility with Version 1.x case-insensitive databases	SQL_1xCompat_Cp850_CI_AS_
51	diction.iso	Dictionary order, case-sensitive	SQL_Latin1_General_Cp1_CS_A
52	nocase.iso	Dictionary order, case-insensitive	SQL_Latin1_General_Cp1_CI_A
53	nocasepr.iso	Dictionary order, case-insensitive, uppercase preference	SQL_Latin1_General_Pref_Cp1_
54	noaccent.iso	Dictionary order, case-insensitive, accent-insensitive	SQL_Latin1_General_Cp1_CI_A
55	altdict.850	Alternate dictionary order, case-sensitive	SQL_AltDiction_Cp850_CS_AS
56	altnocsp.850	Alternate dictionary order, case-insensitive, uppercase preference	SQL_AltDiction_Pref_Cp850_CI
57	altnoacc.850	Alternate dictionary order, case-insensitive, accent-insensitive	SQL_AltDiction_Cp850_CI_AI_
58	scan.850	Scandinavian dictionary sort order, case-insensitive, uppercase preference	SQL_Scandinavian_Pref_Cp850
59	scandict.850	Scandinavian dictionary sort order, case-sensitive	SQL_Scandinavian_Cp850_CS_
60	scannocase.850	Case-insensitive Scandinavian dictionary sort order, without case preference	SQL_Scandinavian_Cp850_CI_
		Alternate dictionary	

61	altnocs.850	order, case-insensitive	SQL_AltDiction_Cp850_CI_AS_
80	binary.250	Binary	SQL_Latin1_General_1250_BIN
81	diction.250	Dictionary order, case-sensitive	SQL_Latin1_General_Cp1250_C
82	nocase.250	Dictionary order, case-insensitive	SQL_Latin1_General_Cp1250_C
83	csydic.250	Czech dictionary order, case-sensitive	SQL_Czech_Cp1250_CS_AS_K
84	csync.250	Czech dictionary order, case-insensitive	SQL_Czech_Cp1250_CI_AS_KI
85	hundic.250	Hungarian dictionary order, case-sensitive	SQL_Hungarian_Cp1250_CS_AS
86	hunnc.250	Hungarian dictionary order, case-insensitive	SQL_Hungarian_Cp1250_CI_AS
87	plkdic.250	Polish dictionary order, case-sensitive	SQL_Polish_Cp1250_CS_AS_K
88	plknc.250	Polish dictionary order, case-insensitive	SQL_Polish_Cp1250_CI_AS_KI
89	romdic.250	Romanian dictionary order, case-sensitive	SQL_Romanian_Cp1250_CS_AS
90	romnc.250	Romanian dictionary order, case-insensitive	SQL_Romanian_Cp1250_CI_AS
91	shldic.250	Croatian dictionary order, case-sensitive	SQL_Croatian_Cp1250_CS_AS_
92	shlnc.250	Croatian dictionary order, case-insensitive	SQL_Croatian_Cp1250_CI_AS_
93	skydic.250	Slovak dictionary order, case-sensitive	SQL_Slovak_Cp1250_CS_AS_K
94	skync.250	Slovak dictionary order, case-insensitive	SQL_Slovak_Cp1250_CI_AS_K
95	slvdic.250	Slovenian dictionary order, case-sensitive	SQL_Slovenian_Cp1250_CS_AS
96	slvnc.250	Slovenian dictionary order, case-insensitive	SQL_Slovenian_Cp1250_CI_AS
104	binary.250	Binary	SQL_Latin1_General_1251_BIN
105	diction.251	Dictionary order, case-	SQL_Latin1_General_Cp1251_C

		sensitive	
106	nocase.251	Dictionary order, case-insensitive	SQL_Latin1_General_Cp1251_CS_AS
107	ukrdic.251	Ukrainian dictionary order, case-sensitive	SQL_Ukrainian_Cp1251_CS_AS
108	ukrnc.251	Ukrainian dictionary order, case-insensitive	SQL_Ukrainian_Cp1251_CI_AS
112	binary.253	Binary	SQL_Latin1_General_1253_BIN
113	diction.253	Dictionary order, case-sensitive	SQL_Latin1_General_Cp1253_CS_AS
114	nocase.253	Dictionary order, case-insensitive	SQL_Latin1_General_Cp1253_CI_AS
121	graltdct.253	Dictionary order, case-sensitive, accent-sensitive	SQL_Latin1_General_Cp1253_CS_AS_KS_WS
124	grnoacce.253	Dictionary order, case-insensitive, accent-insensitive	SQL_Latin1_General_Cp1253_CI_AS_KS_WS
128	binary.254	Binary	SQL_Latin1_General_1254_BIN
129	diction.254	Dictionary order, case-sensitive	SQL_Latin1_General_Cp1254_CS_AS
130	nocase.254	Dictionary order, case-insensitive	SQL_Latin1_General_Cp1254_CI_AS
136	binary.255	Binary	SQL_Latin1_General_1255_BIN
137	diction.255	Dictionary order, case-sensitive	SQL_Latin1_General_Cp1255_CS_AS
138	nocase.255	Dictionary order, case-insensitive	SQL_Latin1_General_Cp1255_CI_AS
144	binary.256	Binary	SQL_Latin1_General_1256_BIN
145	diction.256	Dictionary order, case-sensitive	SQL_Latin1_General_Cp1256_CS_AS
146	nocase.256	Dictionary order, case-insensitive	SQL_Latin1_General_Cp1256_CI_AS
152	binary.257	Binary	SQL_Latin1_General_1257_BIN
153	diction.257	Dictionary order, case-sensitive	SQL_Latin1_General_Cp1257_CS_AS
		Dictionary order, case-	

154	nocase.257	insensitive	SQL_Latin1_General_Cp1257_C
155	etidic.257	Estonian dictionary order, case-sensitive	SQL_Estonian_Cp1257_CS_AS_
156	etinc.257	Estonian dictionary order, case-insensitive	SQL_Estonian_Cp1257_CI_AS_
157	lvindic.257	Latvian dictionary order, case-sensitive	SQL_Latvian_Cp1257_CS_AS_I
158	lvinc.257	Latvian dictionary order, case-insensitive	SQL_Latvian_Cp1257_CI_AS_K
159	lthdic.257	Lithuanian dictionary order, case-sensitive	SQL_Lithuanian_Cp1257_CS_A
160	lthnc.257	Lithuanian dictionary order, case-insensitive	SQL_Lithuanian_Cp1257_CI_AS
183	danonocp.iso	Danish/Norwegian dictionary order, case-insensitive, uppercase preference	SQL_Danish_Pref_Cp1_CI_AS_
184	islanocp.iso	Swedish/Finnish[Phone] dictionary order, case-insensitive, uppercase preference	SQL_SwedishPhone_Pref_Cp1_C
185	svf2nocp.iso.iso	Swedish/Finnish[Std] dictionary order, case-insensitive, uppercase preference	SQL_SwedishStd_Pref_Cp1_CI_
186	islanocp.iso	Icelandic dictionary order, case-insensitive, uppercase preference	SQL_Icelandic_Pref_Cp1_CI_AS

Note:

Each SQL Server collation name has an equivalent string in Transact-SQL code. For a list of sort order identifiers and their Transact-SQL equivalents, see **SQL Collation Name (Transact-SQL)**.

▣ **See Also**

Reference

[Collation Settings in Setup](#)

Concepts

[Using SQL Collations](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Setting Client Code Pages



[See Also](#)



[Preparing to Install SQL Server 2005](#) > [Collation Options and International Support](#) >

The code pages a client uses are determined by your operating system settings. To set client code pages on the Microsoft Windows 2000, Microsoft Windows XP, or Microsoft Windows Server 2003 operating systems, use **Regional Settings** in Control Panel, as described in your Windows documentation.

▣ **See Also**

Reference

[Collation Options and International Support](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Working with Multiple Versions of SQL Server



[See Also](#)



[Preparing to Install SQL Server 2005](#) >

Updated: **12 December 2006**

Microsoft SQL Server 2005 supports multiple instances of the Database Engine, Microsoft SQL Server 2005 Analysis Services (SSAS), and Microsoft SQL Server 2005 Reporting Services (SSRS) on the same computer. You can also upgrade previous versions of SQL Server, or install SQL Server 2005 on a computer where previous SQL Server versions are already installed.

▣ Using SQL Server 2005 Side-By-Side with Previous Versions of SQL Server

You can install SQL Server 2005 on a computer already running instances of a previous SQL Server version. If a default instance already exists on the computer, SQL Server 2005 will be installed as a named instance.

Important For SQL Server 2005 installations in side-by-side configurations with previous versions of SQL Server, SQL Server 2005 services must use accounts found only in the global domains group. Additionally, accounts utilized by SQL Server 2005 services must not appear in the local Administrators group. Failure to comply with this guideline will result in unexpected security behavior.

Side-by-side support is as follows:

Side-by-side support	SQL Server 2000 (32-bit)	SQL Server 2000 (64-bit)	SQL Server 2005 (32-bit)	SQL Server 2005 (64-bit) IA64	SQL Server 2005 (64-bit) X64
SQL Server 7.0	Yes	No	Yes	No	No
SQL Server 2000 (32-bit)	Yes	No	Yes	No ¹	Yes
SQL Server 2000 (64-bit)	No	Yes	No	Yes	No
SQL Server 2005 (32-bit)	Yes	No	Yes	No	Yes
SQL Server 2005 (64-bit) IA64	No	Yes	No	Yes	No
SQL Server 2005 (64-bit) X64	Yes	No	Yes	No	Yes

¹The 32-bit editions of SQL Server are supported in WOW64 only on the X64 versions of 64-bit operating systems. The 32-bit editions of SQL Server are not

supported in WOW64 on the IA64 versions of 64-bit operating systems.

▣ Change History

Release	History
12 December 2006	Changed content: <ul style="list-style-type: none">• Added security guidelines for service accounts in side-by-side configurations.

▣ **See Also**

Tasks

[How to: Install SQL Server 2005 \(Setup\)](#)

Reference

[Hardware and Software Requirements for Installing SQL Server 2005](#)

Concepts

[Using Upgrade Advisor to Prepare for Upgrades](#)

Other Resources

[Upgrading to SQL Server 2005](#)

[Backward Compatibility](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

SQL Server 2005 Preparation How-to Topics



[See Also](#)



[Preparing to Install SQL Server 2005](#) >

This section contains information on how to prepare SQL Server 2005.

▣ In This Section

- [How to: Change Operating System Settings to Support Localized Versions](#)
- [How to: Disable NetBIOS over TCP/IP](#)
- [How to: Disable Server Message Block](#)
- [How to: Verify Permissions for SQL Server Services](#)

▣ **See Also**

Reference

[Collation Options and International Support](#)
[Setting Up Windows Service Accounts](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Change Operating System Settings to Support Localized Versions

 [See Also](#)



[Preparing to Install SQL Server 2005](#) > [SQL Server 2005 Preparation How-to Topics](#) >

Localized versions of Microsoft SQL Server 2005 are supported on English-language versions of supported operating systems through the use of Microsoft Windows Multilingual User Interface Pack (MUI) settings.

However, you must verify certain operating system settings before installing a localized version of SQL Server 2005 on a server that is running an English-language operating system with a non-English MUI setting. You need to verify that the following operating system settings match the language of the localized SQL Server to be installed:

- The operating system user interface setting
- The operating system user locale setting
- The system locale setting

If the settings do not match the language of the localized SQL Server to be installed, then use the following procedures to correctly set these operating system settings.



Caution:

Installations of different language versions of SQL Server instances on the same computer are NOT supported. Example: German SQL Server 2005 and Portuguese (Brazil) SQL Server 2005 are NOT supported as multiple instances on the same computer.

 **To change the operating system user interface setting**

1. If not already installed, install the operating system MUI that matches your localized version of SQL Server 2005.
2. In Control Panel, open **Regional and Language Options**.
3. On the **Languages** tab, for **Language used in menus and dialogs**, select a value from the list.

This setting will affect the user interface language of SQL Server 2005, so it must match your localized version of SQL Server 2005.

4. Click **Apply** to confirm the change, and **OK** to close the window.

▣ **To change the operating system user locale setting**

1. If not already installed, install the operating system MUI that matches your localized version of SQL Server 2005.
2. In Control Panel, open **Regional and Language Options**.
3. On the **Regional Options** tab, for **Select an item to match its preferences**, select a value from the list.

This setting will affect culture-specific data formatting.

4. Click **Apply** to confirm the change, and **OK** to close the window.

▣ **To change the system locale setting**

1. If not already installed, install the operating system MUI that matches your localized version of SQL Server 2005.
2. In Control Panel, open **Regional and Language Options**.

3. On the **Advanced** tab, for **Select a language to match the language version of the non-Unicode programs you want to use**, select a value from the list.

This setting will allow SQL Server Setup to choose the best default collation for your SQL Server 2005 installation.

4. Click **Apply** to confirm the change, and **OK** to close the window.

▣ **See Also**

Reference

[Collation Options and International Support](#)

[Hardware and Software Requirements for Installing SQL Server 2005](#)

Other Resources

[SQL Server 2005 Preparation How-to Topics](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Disable NetBIOS over TCP/IP



[See Also](#)



[Preparing to Install SQL Server 2005](#) > [SQL Server 2005 Preparation How-to Topics](#) >

Updated: 5 December 2005

Servers in the perimeter network should have all unnecessary protocols disabled including NetBIOS. Web servers and Domain Name System (DNS) servers do not require NetBIOS. This protocol should be disabled to reduce the threat of user enumeration.

To disable NetBIOS over TCP/IP

1. From the **Start** menu, right-click **My Computer**, and then click **Manage**.
2. Expand **System Tools**, and then clear the **Device Manager** check box.
3. Right-click **Device Manager**, point to **View**, and then select **Show hidden devices**.
4. Expand **Non-Plug and Play Drivers**.
5. Right-click **NetBios over TCP/IP**, and then click **Disable**.

This disables the SMB direct host listener on TCP/445 and UDP 445.

Note:

This procedure disables the netbt.sys driver. The **WINS** tab of the **Advanced TCP/IP Settings** dialog box contains a **Disable NetBIOS over TCP/IP** option. Selecting this option only disables the NetBIOS Session Service (which listens on TCP port 139). It does not disable NetBIOS completely.

▣ **See Also**

Tasks

[How to: Disable Server Message Block](#)

Concepts

[Security Considerations for a SQL Server Installation](#)

Other Resources

[SQL Server 2005 Preparation How-to Topics](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Disable Server Message Block



[See Also](#)



[Preparing to Install SQL Server 2005](#) > [SQL Server 2005 Preparation How-to Topics](#) >

Updated: 5 December 2005

Servers in the perimeter network should have all unnecessary protocols disabled, including server message block (SMB). Web servers and Domain Name System (DNS) servers do not require SMB. This protocol should be disabled to counter the threat of user enumeration.

To disable SMB

1. On the **Start** menu, point to **Settings**, and then click **Network and Dial-up Connections**.

Right-click Internet facing connection, and then click **Properties**.

2. Select the **Client for Microsoft Networks** check box, and then click **Uninstall**.
3. Follow the uninstall steps.
4. Select **File and Printer Sharing for Microsoft Networks**, and then click **Uninstall**.
5. Follow the uninstall steps.

To disable SMB on servers accessible from the Internet

- In the Local Area Connection properties, use the **Transmission Control**

Protocol/Internet Protocol (TCP/IP) properties dialog box to remove **File and Printer Sharing for Microsoft Networks** and **Client for Microsoft Networks**.

▣ **See Also**

Tasks

[How to: Disable NetBIOS over TCP/IP](#)

Concepts

[Security Considerations for a SQL Server Installation](#)

Other Resources

[SQL Server 2005 Preparation How-to Topics](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Verify Permissions for SQL Server Services



[See Also](#)



[Preparing to Install SQL Server 2005](#) > [SQL Server 2005 Preparation How-to Topics](#) >

Updated: 5 December 2005

Typically, both SQL Server and SQL Server Agent are assigned the same user account, either the local system or a domain user account. However, you can customize the settings for each service during the installation process. For more information about how to customize account information for each service during Setup, see [Service Account](#).



Security Note:

Always run SQL Server services with the least possible privileges.

To verify the permissions for SQL Server services

1. Click **Start**, open **Control Panel**, and then open **Administrative Tools**.
2. Open **Local Security Policy**, select the **Local Policies** folder, and then double-click **User Rights Assignment**.
3. Right-click a permission, and then select **Properties**.
4. In the **Properties** dialog box, verify that the account for SQL Server Agent is listed. If it is not listed, click **Add User or Group**, and then add the user account for the SQL Server Agent service.
5. When the desired changes have been made, click **OK**.
6. Repeat the process for each permission and service.

▣ **Security**

Configure services for SQL Server 2005 so that they function with the minimum required permissions.

▣ **See Also**

Reference

[Setting Up Windows Service Accounts](#)
[Service Account](#)
[Service Accounts \(Clusters\)](#)

Concepts

[Security Considerations for a SQL Server Installation](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Upgrading to SQL Server 2005



[See Also](#)



You can directly upgrade instances of SQL Server 2000 Service Pack 3 (SP3) or later, and instances of SQL Server 7.0 SP4 or later, to SQL Server 2005. You can perform most upgrade operations through Setup; however, some components support or require you to migrate applications or solutions after running Setup. For more information, see [Upgrading SQL Server Components](#).

Before running Setup to upgrade to SQL Server 2005, you should first review system requirements and the upgrade matrix. To review system requirements, see [Preparing to Install SQL Server 2005](#). To review the upgrade matrix, see [Version and Edition Upgrades](#).

After reviewing system requirements and the upgrade matrix, run SQL Server Upgrade Advisor to analyze your instances of SQL Server 2000 and SQL Server 7.0. Upgrade Advisor produces lists of issues specific to your installations that you must fix before or after upgrading. SQL Server Setup will detect blocking issues that will prevent you from upgrading to SQL Server 2005, but will not list issues that may affect your applications.

▣ In This Section

Topic	Description
Version and Edition Upgrades	Lists rules for upgrading from one version of SQL Server to another, or upgrading to a SQL Server edition of the same version.
Using Upgrade Advisor to Prepare for Upgrades	Describes Upgrade Advisor, which analyzes instances of SQL Server 2000 and SQL Server 7.0 and reports known upgrade issues.
Upgrading SQL Server Components	Contains information and instructions for upgrading individual SQL Server components to SQL Server 2005.
Backward Compatibility	Lists the significant changes that might affect your applications when you upgrade to SQL Server 2005.

▣ **See Also**

Concepts

[Preparing to Install SQL Server 2005](#)

[Installing SQL Server 2005](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Version and Edition Upgrades

 [See Also](#)



[Upgrading to SQL Server 2005](#) >

Updated: 14 April 2006

The following upgrade scenarios are supported in this release of SQL Server 2005:

Upgrade from	Supported Upgrade Paths
SQL Server 7.0 Enterprise Edition SP4	SQL Server 2005 Enterprise Edition
SQL Server 7.0 Developer Edition SP4	SQL Server 2005 Enterprise Edition SQL Server 2005 Developer Edition
SQL Server 7.0 Standard Edition SP4	SQL Server 2005 Standard Edition SQL Server 2005 Enterprise Edition
SQL Server 7.0 Desktop Edition SP4	SQL Server 2005 Standard Edition SQL Server 2005 Workgroup Edition
SQL Server 7.0 Evaluation Edition SP4	Upgrades not supported
SQL Server Desktop Engine (MSDE) 7.0 SP4	SQL Server 2005 Express Edition

SQL Server 2000 Enterprise Edition SP3 or later ¹	SQL Server 2005 Enterprise Edition
SQL Server 2000 Developer Edition SP3 or later ¹	SQL Server 2005 Developer Edition
SQL Server 2000 Standard Edition SP3 or later ¹	SQL Server 2005 Enterprise Edition SQL Server 2005 Developer Edition SQL Server 2005 Standard Edition
SQL Server 2000 Workgroup Edition SP3 or later ¹	SQL Server 2005 Enterprise Edition SQL Server 2005 Developer Edition SQL Server 2005 Standard Edition SQL Server 2005 Workgroup Edition
SQL Server 2000 Personal Edition SP3 or later ¹	SQL Server 2005 Standard Edition SQL Server 2005 Workgroup Edition SQL Server 2005 Express Edition SQL Server 2005 Express Edition with Advanced Services
SQL Server 2000 Evaluation Edition SP3 or later ¹	SQL Server 2005 Evaluation Edition
SQL Server Desktop Engine (MSDE)	SQL Server 2005 Workgroup Edition SQL Server 2005 Express Edition

2000 SP3 or later ¹	SQL Server 2005 Express Edition with Advanced Services
SQL Server 2000 IA64 (64-bit) Enterprise Edition	SQL Server 2005 IA64 (64-bit) Enterprise Edition
SQL Server 2000 IA64 (64-bit) Developer Edition	SQL Server 2005 IA64 (64-bit) Enterprise Edition SQL Server 2005 IA64 (64-bit) Developer Edition
SQL Server 2005 Developer Edition ¹	SQL Server 2005 Enterprise Edition SQL Server 2005 Standard Edition SQL Server 2005 Workgroup Edition
SQL Server 2005 Standard Edition ¹	SQL Server 2005 Enterprise Edition SQL Server 2005 Developer Edition
SQL Server 2005 Workgroup Edition ¹	SQL Server 2005 Enterprise Edition SQL Server 2005 Developer Edition SQL Server 2005 Standard Edition
SQL Server 2005 Evaluation Edition ¹	SQL Server 2005 Enterprise Edition SQL Server 2005 Developer Edition SQL Server 2005 Standard Edition

	<p>SQL Server 2005 Workgroup Edition</p> <p>SQL Server 2005 Express Edition</p>
SQL Server 2005 Express Edition ¹	<p>SQL Server 2005 Enterprise Edition</p> <p>SQL Server 2005 Developer Edition</p> <p>SQL Server 2005 Standard Edition</p> <p>SQL Server 2005 Workgroup Edition</p> <p>SQL Server 2005 Express Edition with Advanced Services</p>
SQL Server 2005 Express Edition with Advanced Services ¹	<p>SQL Server 2005 Enterprise Edition</p> <p>SQL Server 2005 Developer Edition</p> <p>SQL Server 2005 Standard Edition</p> <p>SQL Server 2005 Workgroup Edition</p>
SQL Server 2005 IA64 (64-bit) Developer Edition	<p>SQL Server 2005 IA64 (64-bit) Enterprise Edition</p> <p>SQL Server 2005 IA64 (64-bit) Standard Edition</p>
SQL Server 2005 X64 (64-bit) Developer Edition	<p>SQL Server 2005 X64 (64-bit) Enterprise Edition</p> <p>SQL Server 2005 X64 (64-bit) Standard Edition</p>
	SQL Server 2005 IA64 (64-bit)

SQL Server 2005 IA64 (64-bit) Standard Edition	Enterprise Edition SQL Server 2005 IA64 (64-bit) Developer Edition
SQL Server 2005 X64 (64-bit) Standard Edition	SQL Server 2005 X64 (64-bit) Enterprise Edition SQL Server 2005 X64 (64-bit) Developer Edition
SQL Server 2005 IA64 (64-bit) Evaluation Edition	SQL Server 2005 IA64 (64-bit) Enterprise Edition SQL Server 2005 IA64 (64-bit) Developer Edition SQL Server 2005 IA64 (64-bit) Standard Edition
SQL Server 2005 X64 (64-bit) Evaluation Edition	SQL Server 2005 X64 (64-bit) Enterprise Edition SQL Server 2005 X64 (64-bit) Developer Edition SQL Server 2005 X64 (64-bit) Standard Edition

¹This SQL Server edition can be upgraded to SQL Server 2005 on the 32-bit subsystem (WOW64) of a 64-bit server.

Upgrade Notes

- After upgrading to SQL Server 2005 from another edition of SQL Server 2005, you must re-apply any hotfix or service pack updates to the

upgraded SQL Server instance.

- Cross-version instances on SQL Server 2005 are not supported. Version numbers of the Database Engine, Analysis Services, and Reporting Services components must be the same within an instance of SQL Server 2005.
- Before upgrading to SQL Server 2005, enable Windows Authentication for SQL Server Agent and verify that the SQL Server Agent service account is a member of the SQL Server **sysadmin** group.
- Before upgrading from one edition of SQL Server 2005 to another, verify that the functionality you are currently using is supported in the edition to which you are upgrading. For more information, see the section for your components in [Features Supported by the Editions of SQL Server 2005](#).
- To upgrade a SQL Server instance to a SQL Server 2005 failover cluster, the instance being upgraded must be a failover cluster. To upgrade a stand-alone instance of SQL Server to a SQL Server 2005 failover cluster, install a new SQL Server 2005 failover cluster and then migrate user databases from the stand-alone instance using the Copy Database Wizard. For more information on upgrade, see [How to: Upgrade to a SQL Server 2005 Failover Cluster \(Setup\)](#). For more information on database migration, see **Using the Copy Database Wizard**.
- Upgrading a 32-bit instance of SQL Server 2000 from the 32-bit subsystem (WOW64) of a 64-bit server to SQL Server 2005 (64-bit) on the X64 platform is not supported. However, you can upgrade a 32-bit instance of SQL Server to SQL Server 2005 on the WOW64 of a 64-bit server as noted in the table above. You can also backup or detach databases from a 32-bit instance of SQL Server 2000, and restore or attach them to an instance of SQL Server 2005 (64-bit) if the databases are not published in replication. In this case, you must also recreate any logins and other user objects in **master**, **msdb**, and **model** system databases.

▣ **Adding Components to an Instance of SQL Server 2005**

You can add components to an existing installation of SQL Server 2005 from **Add or Remove Programs** in Control Panel and then select the instance of SQL Server to which you want to add or remove features. For more information, see [How to: Change Components in an Existing Instance of SQL Server 2005 \(Setup\)](#).

▣ **Cross-Language Support**

For upgrades:

- English-language versions of SQL Server can be upgraded to any localized version of SQL Server 2005.
- Localized versions of SQL Server can be upgraded to localized versions of SQL Server 2005 of the same language.
- Localized version of SQL Server cannot be upgraded to the English-language version of SQL Server 2005.
- Localized versions of SQL Server cannot be upgraded to localized SQL Server 2005 versions of a different localized language.

Additional cross-language support:

- The English-language version of SQL Server 2005 is supported on all localized versions of supported operating systems.
- Localized versions of SQL Server 2005 are supported on localized operating systems that are the same language as the localized SQL Server version.
- Localized versions of SQL Server 2005 are also supported on English-language versions of supported operating systems through the use of Windows Multilingual User Interface Pack (MUI) settings. However, you must verify certain operating system settings before installing a localized version of SQL Server 2005 on a server that is running an English-language operating system with a non-English MUI setting. You must verify that the following operating system settings match the language of the localized SQL Server to be installed:
 - The operating system user interface setting
 - The operating system user locale setting
 - The system locale setting

If these operating system settings do not match the language of the localized SQL Server, then you must correctly set these operating system settings as described in [How to: Change Operating System Settings to Support Localized Versions](#).

▣ **See Also**

Reference

[Installation Options](#)

[Hardware and Software Requirements for Installing SQL Server 2005](#)

Concepts

[Preparing to Install SQL Server 2005](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Using Upgrade Advisor to Prepare for Upgrades



[See Also](#)



[Upgrading to SQL Server 2005](#) >

Updated: **17 July 2006**

Microsoft SQL Server 2005 Upgrade Advisor helps you prepare for upgrades to SQL Server 2005. Upgrade Advisor analyzes installed SQL Server 2000 or SQL Server 7.0 components, and then generates reports that identify issues to fix either before or after you upgrade to SQL Server 2005.

▣ **How Upgrade Advisor Works**

When you run Upgrade Advisor, the Upgrade Advisor Home page appears. From the Home page, you can run the following tools:

- Upgrade Advisor Analysis Wizard
- Upgrade Advisor Report Viewer
- Upgrade Advisor Help

The first time that you use Upgrade Advisor, run the Upgrade Advisor Analysis Wizard to analyze SQL Server components. When the wizard finishes the analysis, you can view the resulting reports in the Upgrade Advisor Report Viewer. Each report provides links to information in Upgrade Advisor Help that will help you fix or reduce the effect of the known issues.

▣ Upgrade Advisor Analysis

Upgrade Advisor analyzes the following SQL Server components:

- Database Engine
- Analysis Services
- Notification Services
- Reporting Services
- Integration Services, formerly referred to as Data Transformation Services (DTS)

A dedicated analyzer runs in the context of Upgrade Advisor for each SQL Server component. The analysis examines objects that can be accessed by the individual analyzers, such as scripts, stored procedures, triggers, and trace files. Upgrade Advisor cannot analyze desktop applications and encrypted stored procedures.

The output of each analyzer is an XML report for that component. View the XML report using the Upgrade Advisor report viewer.

📌 **Note:**

Reports might contain an "other upgrade issues" item. This item links to a list of issues that are not detected by Upgrade Advisor, but might exist on your server or in your applications. You should review the list of undetectable issues and determine whether you must make changes to your server or applications because of the undetectable issues.

▣ **How to Install and Run Upgrade Advisor**

The location where you install Microsoft SQL Server 2005 Upgrade Advisor depends on what you will be analyzing. Upgrade Advisor supports remote analysis of all supported components except Reporting Services. If you are not scanning instances of Reporting Services, you can install Upgrade Advisor on any computer that can connect to your instance of SQL Server, and that meets the Upgrade Advisor prerequisites. If you are scanning instances of Reporting Services, you must install Upgrade Advisor on the report server.

Upgrade Advisor is available in the Servers\redist\Upgrade Advisor folder of the SQL Server 2005 product disc and from the [Microsoft Download Center](#). The prerequisites for installing Upgrade Advisor are as follows:

- Windows 2000 SP4 or a later version, Windows XP SP2 or a later version, or Windows Server 2003 SP1 or a later version.
- Microsoft Windows Installer 3.1 or a later version.
The .NET Framework version 2.0 requires Windows Installer 3.1. You can install Windows Installer from the [Windows Installer 3.1 Web site](#).
- The Microsoft .NET Framework version 2.0 or a later version. The .NET Framework version 2.0 is available on the SQL Server 2005 product disc, and from the [.NET Framework Developer Center](#).
 - To install the .NET Framework version 2.0 from the SQL Server 2005 disc, locate the root of the CD or DVD drive, double-click the redist folder, double-click the 2.0 folder, and run dotnetfx.exe (for 32 bit) or dotnetfx64.exe (for 64 bit), depending on the operating system you have.
- SQL Server 2000 decision support objects (DSO) are required to scan upgrade issues in Analysis Services . To install DSO, insert the SQL Server 2000 disc into the CD or DVD drive. This starts the SQL Server 2000 Setup program. Click **Install SQL Server 2000 Components**. Click **Analysis Services** to start the Analysis Services Setup program. In **Select Components**, make sure that the **Decision Support Objects** component is selected.
- SQL Server 2000 client components are required to scan DTS packages. If your computer does not have these components, insert the SQL Server 2000 disc into the CD or DVD drive and install them.

To install Upgrade Advisor from the Web, click the download button on the download page. You can then run installation immediately, or save the SQLUASetup.msi file to run later. If you are installing from the product disc, run SQLUASetup.msi directly from the product disc.

After you install Upgrade Advisor, you can start Upgrade Advisor from the Start menu:

- Click **Start**, point to **All Programs**, point to **Microsoft SQL Server 2005**, and then click **SQL Server 2005 Upgrade Advisor**.

For more information about how to run Upgrade Advisor, see the Upgrade Advisor Help and the Readme. These are installed with Upgrade Advisor.

▣ Change History

Release	History
17 July 2006	Changed content: <ul style="list-style-type: none">• Added note about upgrade issues that are not detected by Upgrade Advisor.• Added the location of SQLUASetup.msi on the product media.

▣ **See Also**

Concepts

[Working with Multiple Versions of SQL Server](#)

Other Resources

[Upgrading to SQL Server 2005
Backward Compatibility](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Upgrading SQL Server Components



[See Also](#)



[Upgrading to SQL Server 2005](#) >

This section provides information about upgrading and migrating components from previous versions of Microsoft SQL Server to SQL Server 2005 and between editions of SQL Server 2005. Upgrading is typically performed by SQL Server Setup. Migrating is typically done after installing or upgrading SQL Server 2005.

▣ In This Section

Topic	Description
Upgrading the Database Engine	Provides information about upgrading the Database Engine from SQL Server 7.0 or SQL Server 2000.
Upgrading or Migrating Analysis Services	Contains information about the upgrade options for Analysis Services and how to upgrade or migrate Analysis Services databases.
Upgrading or Migrating Data Transformation Services	Contains information about upgrading to SQL Server 2005 Integration Services (SSIS) and how to migrate packages to SQL Server 2005.
Migrating Notification Services Instances	Contains information about migrating instances of Notification Services from Notification Services 2.0 or upgrading to a higher edition of SQL Server 2005.
Upgrading Reporting Services	Describes the requirements and steps for upgrading Reporting Services, and provides information about which components are removed.
Upgrading Replicated Databases	Provides information about upgrading replicated databases.
Upgrading SQL Server Components How-to Topics	Contains step-by-step instructions for upgrading SQL Server components.

▣ **See Also**

Concepts

[Installing SQL Server 2005](#)

Other Resources

[Upgrading to SQL Server 2005](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Upgrading the Database Engine



[See Also](#)



[Upgrading to SQL Server 2005](#) > [Upgrading SQL Server Components](#) >

Updated: **12 December 2006**

You can upgrade the Microsoft SQL Server Database Engine from Microsoft SQL Server version 7.0 or Microsoft SQL Server 2000 to Microsoft SQL Server 2005. For information about supported upgrade paths, see [Version and Edition Upgrades](#).

The SQL Server 2005 Setup program can upgrade a previous version of SQL Server with minimal user intervention. However, proper preparation and a familiarity with the upgrade process can make it easier to prevent or solve any issues that arise.

This topic provides the information you will need to prepare for and understand the upgrade process; it covers:

- Known upgrade issues.
- Pre-Upgrade tasks and considerations.
- Links to procedural topics for upgrading the Database Engine to SQL Server 2005.
- Links to procedural topics for migrating databases to SQL Server 2005.
- Considerations for failover clusters.
- Post-upgrade tasks and considerations.

▣ **Known Upgrade issues**

Before upgrading the Database Engine to SQL Server 2005, review [SQL Server 2005 Database Engine Backward Compatibility](#). For backward compatibility content for other SQL Server components, see [Backward Compatibility](#).

▣ **Important:**

Before upgrading from one edition of SQL Server 2005 to another, verify that the functionality you are currently using is supported in the edition to which you are upgrading. For more information, see the section for your components in [Features Supported by the Editions of SQL Server 2005](#).

▣ Pre-Upgrade Checklist

Upgrading to SQL Server 2005 from a previous SQL Server version is supported by the SQL Server 2005 Setup program. You can also migrate databases from previous SQL Server versions to an instance of SQL Server 2005. Migration can be from one SQL Server instance to another on the same computer, or from a SQL Server instance on another server. Migration options include use of the Copy Database Wizard, Backup and restore functionality, use of the SQL Server 2005 Integration Services Import and Export Wizard, and bulk export/bulk import methods.

Before upgrading the Database Engine, make sure to do the following:

Review documentation related to upgrading to SQL Server 2005:

- Review [Hardware and Software Requirements for Installing SQL Server 2005](#).
- Review [Check Parameters for the System Configuration Checker](#).
- Review [Security Considerations for a SQL Server Installation](#).
- Review [Using Upgrade Advisor to Prepare for Upgrades](#).
- Review [SQL Server 2005 Database Engine Backward Compatibility](#).

Review the following issues and make changes, as necessary, before upgrading to SQL Server 2005:

- When upgrading from a 64-bit edition of Microsoft SQL Server 2000 to a 64-bit edition of Microsoft SQL Server 2005, you must upgrade Analysis Services before upgrading the Database Engine.
- Back up all SQL Server database files from the instance to be upgraded, so you can completely restore them, if necessary.
- Run the appropriate Database Console Commands (DBCC) on databases to be upgraded to ensure they are in a consistent state.
- Estimate the disk space required to upgrade SQL Server components, as well as user databases. For disk space required by SQL Server 2005 components, see [Hardware and Software Requirements for Installing SQL Server 2005](#).
- Ensure that existing SQL Server system databases - **master**, **model**, **msdb**, and **tempdb** - are configured to autogrow, and ensure that they have

adequate hard disk space.

- Ensure that all database servers have logon information in the **master** database. This is important for restoring a database, as system logon information resides in **master**.
- Disable all startup stored procedures, as the upgrade process will stop and start services on the SQL Server instance being upgraded. Stored procedures processed at startup time may block the upgrade process.
- Stop Replication and make sure that the replication log is empty.
- Upgrading an instance of the SQL Server 2000 Database Engine to SQL Server 2005 retains the configuration value for **max worker threads**. However, we recommend changing the **max worker threads** value to 0 before upgrading, to allow the Database Engine to calculate the optimal number of threads. For more information, see **max worker threads Option**.
- Quit all applications, including all services with SQL Server dependencies. Upgrade may fail if local applications are connected to the instance being upgraded. For more information, see [Troubleshooting an Installation of the SQL Server Database Engine](#).

▣ **Upgrading the Database Engine**

You can overwrite an installation of SQL Server 7.0 or SQL Server 2000 with a version upgrade to SQL Server 2005. If a previous version of SQL Server is detected when you run SQL Server 2005 Setup, all previous SQL Server program files are upgraded, and all data stored in the previous SQL Server instance is preserved. In addition, previous versions of SQL Server Books Online will remain intact on the machine. To access previous versions of SQL Server Books Online, see [How to: Access Books Online for SQL Server 7.0](#), or [How to: Access Books Online for SQL Server 2000](#).

Caution:

When you upgrade to SQL Server 2005, the previous SQL Server instance will be overwritten and will no longer exist on your computer. Before upgrading, back up SQL Server databases and other objects associated with the previous SQL Server instance.

You can upgrade the Database Engine using the SQL Server Installation Wizard (Setup.exe in User Interface mode) or from the command prompt (Setup.exe).

Caution:

If your computer has SQL Server 2000 Management Tools and a default instance of SQL Server 2005 installed, SQL Server Setup will permit you to install a SQL Server 2000 default instance. However, doing so will disable the installed instance of SQL Server 2005. Therefore, do not install a default instance of SQL Server when SQL Server 2000 Management Tools and a default instance of SQL Server 2005 already exist on the computer.

Database Compatibility Level After Upgrade

After an upgrade, SQL Server 2005 automatically sets the database compatibility level to the level of the previous version of SQL Server. Therefore, if you upgrade from SQL Server 7.0 through SQL Server 2000 and then to SQL Server 2005, the compatibility level will be set to SQL Server 2000.

To upgrade the Database Engine to SQL Server 2005

- [How to: Upgrade to SQL Server 2005 \(Setup\)](#)
- [How to: Upgrade to a SQL Server 2005 Failover Cluster \(Setup\)](#)
- [Upgrading Replicated Databases](#)

▣ **Migrating Databases to SQL Server 2005**

You can move user databases to an instance of SQL Server 2005 using backup and restore or detach and attach functionalities in SQL Server 2005.

▣ **Important:**

A database with the identical name on both source and destination servers cannot be moved or copied. In this case, it will be noted as "Already exists."

To migrate user databases to SQL Server 2005

- **Using the Copy Database Wizard**
- **How to: Upgrade to SQL Server 2005 with the Copy Database Wizard**

▣ Failover Clustering Considerations when Upgrading the Database Engine

SQL Server 2005 failover clustering provides high-availability support for an entire SQL Server instance. A failover cluster is a combination of one or more physical disks in a Microsoft Cluster Service (MSCS) cluster group, known as a resource group, that are participating nodes of the cluster. The resource group is configured as a virtual server that hosts an instance of SQL Server.

A SQL Server virtual server appears on the network as if it were a single computer, but has functionality that provides failover from one node to another if one node becomes unavailable. For example, during a hardware failure, operating system failure, or planned upgrade, you can configure an SQL Server instance on one node of a failover cluster to fail over to any other node in the disk group. You can use failover clustering to reduce system downtime and provide higher application availability.

To upgrade a SQL Server instance to a SQL Server 2005 failover cluster, the instance being upgraded must be a failover cluster. To upgrade a stand-alone instance of SQL Server to a SQL Server 2005 failover cluster, install a new SQL Server 2005 failover cluster and then migrate user databases from the stand-alone instance using the Copy Database Wizard. For more information on upgrade, see [How to: Upgrade to a SQL Server 2005 Failover Cluster \(Setup\)](#). For more information on database migration, see **Using the Copy Database Wizard**.

For more information about failover clustering, see [Failover Clustering](#) and [Installing a Failover Cluster](#). For more information about upgrading to a SQL Server 2005 failover cluster, see [How to: Upgrade to a SQL Server 2005 Failover Cluster \(Setup\)](#).

▣ After Upgrading the Database Engine

After upgrading the Database Engine to SQL Server 2005, complete the following tasks:

- After upgrading to SQL Server 2005 from another edition of SQL Server 2005, you must re-apply any hotfix or service pack updates to the upgraded SQL Server instance.
- **Register your servers** - Upgrade removes registry settings for the previous SQL Server instance. After upgrading, you must reregister your servers. For more information about registering servers, see the **Registering Servers** topic in SQL Server 2005 Books Online.
- **Update statistics** - To help optimize query performance, we recommend that you update statistics on all databases following upgrade. Use the **sp_updatestats** stored procedure to update statistics in user-defined tables in SQL Server 2005 databases.
- **Update usage counters** - In earlier versions of SQL Server, the values for the table and index row counts and page counts can become incorrect. To correct any invalid row or page counts, we recommend that you run **DBCC UPDATEUSAGE** on all databases following upgrade.
- **Configure your new SQL Server installation** - To reduce the attackable surface area of a system, SQL Server 2005 selectively installs and activates key services and features. For more information on how to activate SQL Server 2005 features, see [SQL Server Surface Area Configuration](#).

Any databases that were marked full-text enabled or disabled before the upgrade will maintain that status after upgrade. After the upgrade, the full-text catalogs will be rebuilt and populated automatically for all full-text enabled databases. This is a time-consuming and resource-consuming operation. You can pause the full-text indexing operation temporarily by running the following statement:

 [Copy Code](#)

```
EXEC sp_fulltext_service 'pause_indexing', 1
```

To resume full-text index population, run the following statement:

 [Copy Code](#)

```
EXEC sp_fulltext_service 'pause_indexing', 0
```

▣ Change History

Release	History
12 December 2006	Changed content: <ul style="list-style-type: none">• In "After Upgrading the Database Engine," removed bulleted item about repopulating full-text catalogs. The paragraph about full-text catalogs being automatically rebuilt explains the correct behavior.
17 July 2006	New content: <ul style="list-style-type: none">• Added links to topics for registering servers and repopulating full-text catalogs.

▣ **See Also**

Concepts

[Working with Multiple Versions of SQL Server](#)
[Upgrading Replicated Databases](#)

Other Resources

[Backward Compatibility](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Upgrading or Migrating Analysis Services



[See Also](#)



[Upgrading to SQL Server 2005](#) > [Upgrading SQL Server Components](#) >

Note:

We recommend that you review [Version and Edition Upgrades](#) for general information about upgrade scenarios for Microsoft SQL Server 2005 products and technologies.

If you are using Microsoft SQL Server 2000 Analysis Services, you have the following options when you install Microsoft SQL Server 2005 Analysis Services (SSAS):

- [Install Analysis Services side-by-side with an earlier version.](#)
- [Upgrade an earlier version of Analysis Services.](#)
- [Migrate existing Analysis Services databases.](#)

Important:

Regardless of which installation option you choose, you should be aware of which SQL Server 2000 Analysis Services features and functionality have been discontinued, deprecated, or changed. For more information, see [SQL Server 2005 Analysis Services Backward Compatibility](#).

▣ **Known Upgrade Issues**

Before upgrading to SQL Server 2005 Analysis Services, review [Known Upgrade Issues for SQL Server 2005 Analysis Services](#).

▣ **Pre-Upgrade Checklist**

Before upgrading to SQL Server 2005 Analysis Services, review the following requirements:

- [Hardware and Software Requirements for Installing SQL Server 2005](#)
- [Check Parameters for the System Configuration Checker](#)
- [Security Considerations for a SQL Server Installation](#)
- [Using Upgrade Advisor to Prepare for Upgrades](#)

▣ **Installing Analysis Services Side-by-Side with an Earlier Version**

SQL Server 2000 Analysis Services does not support named instances. Therefore, on a computer that is running instances of both SQL Server 2000 Analysis Services and SQL Server 2005 Analysis Services, the instance of SQL Server 2000 Analysis Services must always be the default instance. In this case, SQL Server Setup permits only the installation of named instances of SQL Server 2005 Analysis Services. SQL Server 2000 Analysis Services can continue to run as the default instance, side-by-side with named instances of SQL Server 2005 Analysis Services.

For more information, see [How to: Install Analysis Services Side-by-Side with an Earlier Version](#).

Modifying a Side-by-Side Installation

You can use the Analysis Services Migration Wizard to migrate databases from an instance of SQL Server 2000 Analysis Services to SQL Server 2005 Analysis Services. For more information about running the Migration Wizard, see [Migrating Existing Analysis Services Databases](#).

If you remove the default instance of SQL Server 2000 Analysis Services after you have migrated its databases, you can use the Analysis Services Instance Rename tool to make a named instance of SQL Server 2005 Analysis Services the default instance on the server.

For more information, see [How to: Migrate Analysis Services Databases](#) and [How to: Rename an Instance of Analysis Services](#).

▣ **Upgrading an Earlier Version of Analysis Services**

Upgrading SQL Server 2000 Analysis Services involves the installation of SQL Server 2005 Analysis Services and the migration of existing Analysis Services databases to the SQL Server 2005 Analysis Services database format. You can upgrade SQL Server 2000 Analysis Services and migrate existing databases during Setup, or you can migrate databases after you install SQL Server 2005 Analysis Services, by using other software tools in addition to SQL Server Setup.

▣ **Important:**

When upgrading from a 64-bit edition of Microsoft SQL Server 2000 to a 64-bit edition of Microsoft SQL Server 2005, you must upgrade Analysis Services before upgrading the Database Engine.

To upgrade an instance of SQL Server 2000 Analysis Services during Setup

1. On the **Instance Name** page of the SQL Server Installation Wizard, select **Default Instance**.
2. To confirm the presence of the SQL Server 2000 Analysis Services instance, click the **Installed Instances** button.

To migrate an instance of SQL Server 2000 Analysis Services after Setup

1. Install a named instance of SQL Server 2005 Analysis Services by using the SQL Server Installation Wizard or an alternative installation method. For more information, see [Installing SQL Server Analysis Services by Using Setup](#) or [Installing SQL Server Analysis Services from the Command Prompt](#).
2. Use the Analysis Services Migration Wizard to migrate SQL Server 2000 Analysis Services database metadata to SQL Server 2005 Analysis Services. This step re-creates the database structures in the new SQL Server 2005 Analysis Services database format while leaving the original databases intact. For more information, see [Migrating Existing Analysis Services Databases](#).

3. Process the new SQL Server 2005 Analysis Services databases. This step populates the new database structures with data. For more information, see **Processing Analysis Services Objects**.
4. Verify the database upgrade process. Review the upgraded databases in SQL Server Management Studio, and test them with your client applications.
5. Uninstall SQL Server 2000 Analysis Services if it is no longer needed.
6. If you remove the SQL Server 2000 Analysis Services default instance, you can use the Analysis Services Instance Rename tool to rename the named instance of SQL Server 2005 Analysis Services as the default instance.

For more information, see [How to: Upgrade an Instance of Analysis Services](#), [How to: Migrate Analysis Services Databases](#), and [How to: Rename an Instance of Analysis Services](#).

▣ **Migrating Existing Analysis Services Databases**

You can use the Analysis Services Migration Wizard to upgrade SQL Server 2000 Analysis Services databases to the SQL Server 2005 Analysis Services format. During migration, the wizard copies SQL Server 2000 Analysis Services database objects and then re-creates them on the specified instance of SQL Server 2005 Analysis Services. The source databases are left intact and are not modified.

For more information about migrating to SQL Server 2005 Analysis Services, see [Migrating Existing Analysis Services Databases](#) and [How to: Migrate Analysis Services Databases](#).

After you migrate your databases, you must process them from the original data source before you can run queries against them. For more information, see **Processing Analysis Services Objects**.

▣ **See Also**

Concepts

[Installing SQL Server Analysis Services](#)

Other Resources

[Upgrading SQL Server Components](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Migrating Existing Analysis Services Databases



[See Also](#)



[Upgrading to SQL Server 2005](#) > [Upgrading SQL Server Components](#) > [Upgrading or Migrating Analysis Services](#) >

You can use the Analysis Services Migration Wizard to upgrade Microsoft SQL Server 2000 Analysis Services databases to Microsoft SQL Server 2005 Analysis Services (SSAS). During migration, the wizard copies SQL Server 2000 Analysis Services database objects and then re-creates them on an instance of SQL Server 2005 Analysis Services. The source databases are left intact and are not modified. After you verify that the new databases are fully operational, you can manually delete the old databases.

As a best practice, migrate your databases one at a time or in small batches so that you can verify that each database object appears as you expect on the destination server, before you migrate additional objects. When you use the Migration Wizard, the **MSSQLServerOLAPService** service must be running on both the source and the destination server. (If you are migrating a named instance, the **MSOLAP\$instancename** service must be running on both servers.)

After you migrate your databases, you must process them from the original data source before you can run queries against them. For more information, see **Processing Analysis Services Objects**.

▣ Using the Migration Wizard

You can start the Migration Wizard from an Analysis Services server node in the Object Browser in SQL Server Management Studio. You can also start the wizard at the command prompt, by running the program **MigrationWizard.exe**.

Specifying a Source and a Destination

After the initial **Welcome** page in the wizard, the **Specify Source and Destination** page appears. On this page, you specify the source server and the destination server for the databases that you are migrating. The destination instance of Analysis Services can be either local or remote. If you specify a destination computer that is running more than one instance of SQL Server 2005 Analysis Services, use the format *computername\instancename*.

You can also choose to save the database schema to a script file and complete the migration later by using an Analysis Services Execute DDL task in a SQL Server 2005 Integration Services (SSIS) package. For more information about the Analysis Services Execute DDL task, see **Analysis Services Execute DDL Task**.

Selecting Databases to Migrate

The next page of the wizard, **Select Databases to Migrate**, displays a list of all SQL Server 2000 Analysis Services databases and the corresponding SQL Server 2005 Analysis Services databases to be created. By default, all databases are selected, but you can clear the check box next to the name of any database that you do not want to migrate.

By default, the names of a source database and its corresponding destination database are identical. However, if another database with the same name already exists on the destination server, the Migration Wizard assigns a unique name for the new database. (For example, a second copy of AdventureWorks becomes AdventureWorks 1.) You can use the name provided in the wizard, type a new name in the wizard, or rename the database after you complete the wizard.

For more information about how SQL Server 2000 Analysis Services objects

correspond to SQL Server 2005 Analysis Services objects, see [Migration Considerations \(SSAS\)](#).

 **Note:**

The Migration Wizard does not support the migration of Microsoft SQL Server 7.0 Analysis Services databases.

Validating Databases

On the **Validating Databases** page, the Migration Wizard validates the structure of the databases to be migrated. The wizard generates a log as the databases are validated. To view the log or a subset of the log entries, click **View Log**, and then select from the following options:

- **Show All**
- **Show Errors**
- **Show Warnings**
- **Show Successes**

Validation stops if an error occurs. To continue the migration process, fix the database that caused the error, and then click **Restart**.

You can stop the validation process at any time by clicking **Stop**. The wizard displays a message that indicates the number of databases that have not been validated.

Migrating Databases

On the **Migrating Databases** page, the wizard migrates SQL Server 2000 Analysis Services objects to SQL Server 2005 Analysis Services format. You can stop the process and roll back the whole transaction by clicking **Stop** and then **Cancel**.

After migration is complete, you can change selections you made earlier by clicking **Back** to return to earlier pages of the wizard. However, if you do this, you must run the migration process again. Running the migration process again deletes the databases that have already been migrated.

Completing the Wizard

The **Completing the Wizard** page displays a summary of the SQL Server 2005 Analysis Services databases that have been created. Click **Finish** to complete the wizard.

After you migrate a database, you must process the database from the original data source before you can query the database. For more information, see **Processing Analysis Services Objects**.

▣ **See Also**

Concepts

[Upgrading or Migrating Analysis Services](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Migration Considerations (SSAS)



[See Also](#)



[Upgrading to SQL Server 2005](#) > [Upgrading SQL Server Components](#) > [Upgrading or Migrating Analysis Services](#) >

When you migrate from earlier versions of Microsoft SQL Server Analysis Services to SQL Server 2005 Analysis Services (SSAS), you must be aware of changes in features, functionality, structure, and behavior. This overview topic provides information about common migration considerations that relate to data mining and the Analysis Services engine. For more information about specific areas of migration considerations, see the following topics:

- [Known Upgrade Issues for SQL Server 2005 Analysis Services](#)
- [Migrating Existing Analysis Services Databases](#)
- [Discontinued Analysis Services Functionality in SQL Server 2005](#)
- [Deprecated Analysis Services Functionality in SQL Server 2005](#)

For more information about new and enhanced features in SQL Server 2005 Analysis Services (SSAS), see **Analysis Services Enhancements (SSAS)**.

▣ Analysis Services Migration Considerations

Analysis Services Objects

The following table describes how SQL Server 2000 Analysis Services objects correspond to SQL Server 2005 Analysis Services (SSAS) objects.

SQL Server 2000 Analysis Services object	SQL Server 2005 Analysis Services (SSAS) object	More information
Aggregation	Aggregation	Only aggregations for hybrid OLAP (HOLAP) and multidimensional OLAP (MOLAP) partitions are migrated.
Analysis Services database	Analysis Services database	None.
Command	MDX script	A Multidimensional Expressions (MDX) script is written for the object that contains the command collection.
Cube	Cube	One data source view is also created for all dimensions, virtual dimensions, and cubes that share the same data source. A data source view is then created for each mining model and data mining dimension.
Cube and database role commands	Not migrated	None.
Cube role	Cube permission	Dimension and cell permissions are not migrated.
Data mining dimension	Data mining dimension	A data source view is also created for each data mining dimension.
Data source	Data source	None.

Database role	Database role and database permission	None.
Dimension	Dimension with a single hierarchy	None.
Dimensions that share the same name prefix (representing multiple hierarchies)	Separate dimensions, each with a single hierarchy.	MDX expressions that refer to the original hierarchy names no longer function.
Level in a hierarchy	Attribute in the dimension, and also a level in a user-defined hierarchy.	None.
Linked cubes	Not migrated	None.
Member property on a level	Member property on a level	None.
Mining model	Mining model	A data source view is created for each mining model in a database.
Mining model role	Migrated as two permissions: one for the mining structure and one for the mining model.	None.
	Single parent-child	

Multiple parent-child hierarchies in a dimension	hierarchy, which is defined by settings on the parent attribute.	MDX expressions that refer to the original hierarchies no longer function. A SQL Server 2005 Analysis Services (SSAS) dimension can only have a single parent-child hierarchy.
Remote partitions	Not migrated.	Remote partitions are unavailable until the server on which they are located is upgraded or the database that contains the remote partition is migrated. Databases that contain remote partitions can be migrated individually. The definition of the remote partition must be updated to identify the name of the new server instance.
Repository	Object Definition Language (ODL) files in XML format	None.
Virtual cube	Linked measure groups	None.
Virtual dimension	A hierarchy of the source dimension	Virtual parent-child dimensions are migrated as separate dimensions, because parent-child dimensions cannot have multiple hierarchies.

Custom Level Formulas

In SQL Server 2005 Analysis Services (SSAS), assignments replace calculated cells and custom level formulas. Calculated cells are supported in SQL Server 2005 Analysis Services (SSAS) and are migrated from SQL Server 2000 Analysis Services as defined, but custom level formulas are migrated as assignments.

In SQL Server 2005 Analysis Services (SSAS), the calculation pass is automatically incremented with each pass. Therefore, if a cube contains custom level formulas, make sure that you verify that the results are the same as

intended.

Data Source Views

Data source views are new in SQL Server 2005 Analysis Services (SSAS). A data source view differs from the cube schema in SQL Server 2000 Analysis Services in that the data source view is a logical representation of the source data from which multiple cubes can be defined, whereas the cube schema in SQL Server 2000 Analysis Services is a physical representation of relationships between tables for each cube.

Migrating Multiple-Hierarchy Dimensions

The model for multiple-hierarchy dimensions has changed from SQL Server 2000 Analysis Services to SQL Server 2005 Analysis Services (SSAS). Wherever possible, such hierarchies are migrated to the same dimension, but there are circumstances when this is not possible. When this occurs, calculations that rely on the original dimension hierarchy might fail.

SQL Syntax

The means of stating SQL queries has changed in SQL Server 2005 Analysis Services (SSAS). The Backus-Naur Form (BNF) for the new syntax is used in SQL Server 2005 Books Online.

Schema Rowset Changes

Many schema rowsets have been augmented in SQL Server 2005 Analysis Services (SSAS) to provide additional information. Most of the time, the position and meaning of existing rowsets and columns remain exactly as in SQL Server 2000 Analysis Services. The following table describes the schema rowset changes that have occurred in SQL Server 2005 Analysis Services (SSAS).

Schema	Column	Change
DBSCHEMA_TABLES_INFO	All	This schema is no longer

		supported.
MDSHEMA_CELL_FORMULAS	All	This schema is no longer supported.
MDSHEMA_DIMENSIONS	DIMENSION_CARDINALITY	This column returns the number of dimension key attribute members.
MDSHEMA_MEMBERS	LEVELS_LIST	This schema column contains null.
MDSHEMA_MEMBERS	MEMBER_ORDINAL	This column is deprecated. The value 0 is returned instead of the member ordinal.

Unique Member Names

Unique member names differ in SQL Server 2005 Analysis Services (SSAS) from those in SQL Server 2000 Analysis Services. Because of restrictions on the member unique name, retrieving results from the members schema by using the original unique name might not return the same results as in SQL Server 2000 Analysis Services.

▣ Data Mining Migration Considerations

The Migration Wizard transfers all the mining models from the SQL Server 2000 Analysis Services database to the destination SQL Server 2005 Analysis Services (SSAS) database. For each mining model in the source database, the Migration Wizard creates a new mining structure in the destination database. The mining structure contains exactly one model, and that model has the same name as the mining model in the source database.

To migrate individual mining models from SQL Server 2000 Analysis Services to an instance of SQL Server 2005 Analysis Services (SSAS), you can use the Predictive Model Markup Language (PMML). The following statement will retrieve the PMML representation, in the form of an XML script, of a SQL Server 2000 Analysis Services mining model:

```
SELECT * FROM [Model_Name].PMML
```

You can obtain the same information by retrieving the OLE DB schema, `DMSHEMA_MINING_MODEL_XML`.

You can use the PMML script to create a new mining model in an instance of SQL Server 2005 Analysis Services (SSAS) by using a statement similar to the following:

```
CREATE MINING MODEL [Model_Name] FROM PMML '<PMML script>'
```

If you use the Analysis Services Migration Wizard to migrate your mining models, be aware that the wizard only supports the migration of metadata. Therefore, migrated mining models are unprocessed. However, the models still contain original data bindings, so that you can process them in SQL Server 2005 Analysis Services (SSAS) after migration. On the other hand, if you use the PMML to migrate the models, the original data bindings are not preserved. Therefore, you cannot reprocess the migrated models. Nevertheless, the models are ready to use, because the PMML includes the patterns and trends that were discovered by the source model.

Known data mining migration issues:

- Mining models that have multiple case-level keys with multiple keys in a nested table cannot be migrated.
- Mining models that were created by using aggregated data mining providers cannot be migrated.

Change History

Release	History
17 July 2006	Changed content: <ul style="list-style-type: none">• Updated conceptual material to improve clarity.

▣ **See Also**

Concepts

[Upgrading or Migrating Analysis Services](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Upgrading Reporting Services



[See Also](#)



[Upgrading to SQL Server 2005](#) > [Upgrading SQL Server Components](#) >

Updated: **14 April 2006**

You can upgrade Reporting Services from SQL Server 2000 Reporting Services to the version that ships in SQL Server 2005. How you upgrade depends on the complexity of your installation, whether you are using custom extensions, and whether you have server availability requirements.

Custom installations, Internet-facing installations, and production servers that have up-time requirements require a migration plan that determines in advance the optimal time to perform the upgrade, how best to upgrade the report server database, verification steps that confirm whether the upgrade succeeded, and a backup plan in case the upgrade fails for some unexpected reason. If you are using custom extensions for data processing, delivery, security, or rendering you might need to recompile the extensions before upgrading. If you have a production server, you might want to migrate your installation rather than run an in-place upgrade. You should also consider installing a new instance side-by-side with your existing Reporting Services installation. This allows you to test the installation thoroughly before removing the SQL Server 2000 instance.

▣ Upgrading From SQL Server 2000 to SQL Server 2005

Use the following table to view upgrade options for different deployment scenarios.

Initial state	Action	State after upgrade
Reporting Services and Database Engine on the same SQL Server 2000 default instance.	Run Upgrade. Choose the default instance.	Reporting Services and the Database Engine are upgraded to SQL Server 2005.
<p>One instance of SQL Server 2000 Reporting Services that runs as the default instance.</p> <p>One instance of the SQL Server 2000 Database Engine that runs as a named instance.</p> <p>Both instances run on the same computer.</p>	Run Upgrade. Choose the instance you want to upgrade. You can upgrade each instance independently of the other. You can also upgrade all instances at once.	<p>Separate instances of Reporting Services and the Database Engine, upgraded to SQL Server 2005. Valid combinations include:</p> <p>SQL Server 2005 Reporting Services with SQL Server 2000 Database Engine instance.</p> <p>SQL Server 2000 Reporting Services and SQL Server 2005 Database Engine instance.</p> <p>Reporting Services and the Database Engine both upgraded to SQL Server 2005.</p>

▣ **Upgrading a Single-Server Installation**

If you are currently running a default installation that has not been modified, you can upgrade the entire Reporting Services installation by running Setup once. You must be using a local instance of SQL Server Database Engine to host the report server databases. For more information, see [Upgrading a Default Installation of Reporting Services](#).

If you are running a customized installation, you might need to migrate Reporting Services. For more information about migration and upgrade considerations, see [Migrating Reporting Services](#), [How to: Install Reporting Services Side-by-Side with an Earlier Version](#), [Using Upgrade Advisor to Prepare for Upgrades](#), and [Breaking Changes in SQL Server 2005 Reporting Services](#).

▣ **Upgrading a Distributed Server Installation**

You can configure a Reporting Services installation to host the report server database on a remote computer. You can also configure an installation to run multiple report servers in a scale-out deployment. If you install the report server database on a remote computer, Setup will update the schema for you automatically. If Setup cannot update the schema due to insufficient permissions or connection errors, you must upgrade the report server database manually. For more information, see [Upgrading a Report Server Database](#).

You can upgrade the report server database without upgrading the Database Engine instance that hosts it. If you do upgrade the Database Engine instance, be aware that there are edition requirements for hosting the report server database. For more information about edition requirements, see [Creating a Report Server Database](#). For more information about upgrading multiple report server instances, see [Upgrading a Report Server Web Farm](#).

▣ In This Section

Topic	Description
Upgrading a Default Installation of Reporting Services	Explains how to upgrade an existing Reporting Services installation, where all components are configured to use the default values.
Upgrading Reports	Explains how to upgrade reports that you created in the previous version of Report Designer.
Updating Role-Based Security for Report Builder Access	Explains how to modify role-based security settings to use new ad hoc reporting functionality.
Upgrading a Report Server Database	Provides information about updating the report server database schema.
Upgrading Editions of Reporting Services	Describes the functional differences between the Evaluation and Standard Editions. You can upgrade any edition of Reporting Services, including the Evaluation Edition if it has not expired.
Upgrading a Report Server Web Farm	Explains how to upgrade multiple report servers that share a single report server database.
Migrating Reporting Services	Provides instructions for upgrading an installation that you have modified and installations that consist of components installed on multiple computers.

▣ **See Also**

Tasks

[How to: Install Reporting Services Side-by-Side with an Earlier Version](#)

Concepts

[Installing SQL Server Reporting Services](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Upgrading a Default Installation of Reporting Services



[See Also](#)



[Upgrading to SQL Server 2005](#) > [Upgrading SQL Server Components](#) > [Upgrading Reporting Services](#) >

When you run Setup on a computer that already has a default installation of Reporting Services, you can choose whether to upgrade the existing instance in-place or install a new one. Requirements for an in-place upgrade are as follows:

- The virtual directories used to access the report server and Report Manager must have the default configuration settings. The virtual directories must be created under the default Web site. Application and content mappings, virtual directory security settings, and permissions must have default values.
- The ASP.NET process identity has not been encrypted in the machine.config file. Encrypting the process identity is a security best practice for some organizations. SQL Server setup cannot upgrade a Reporting Services installation when the process identity is encrypted.
- No custom extensions have been added and the report server is deployed with the default security extension.

The items described above will block an upgrade and are triggered when you run Upgrade Advisor on an existing installation. For more information about these issues and corrective actions, see the Upgrade Advisor product documentation. You can migrate a Reporting Services installation if upgrading is not an option. For more information about migration and using Upgrade Advisor, see [Migrating Reporting Services](#) and [Using Upgrade Advisor to Prepare for Upgrades](#).

▣ **Pre-Upgrade Checklist**

Use this checklist to prepare your Reporting Services installation for an upgrade.

1. Back up the report server database and configuration files. If you modified the sample files or style sheets, and you want to keep the revised versions, back up those as well.
2. Back up the symmetric key.
3. Uninstall the Reporting Services Books Online Documentation Refresh, if you installed it.
4. Run Setup on the computer that hosts the report server. Setup detects the existing installation and provides options to upgrade.

▣ **Upgrading the Report Server Database**

During an upgrade, Setup connects to the report server databases installed on the local default SQL Server Database Engine instance, and then updates the existing table structure to the new schema. Once you upgrade a report server database to the new format, you cannot undo it. The database is migrated in place. Setup uses the database connection information in the RSReportServer.config file to find and connect to the report server database.

Upgrading an installation does not modify the name of the report server database or the connection information. The account information that the report server uses to access the report server database is unchanged. For more information about upgrading the database, see [Upgrading a Report Server Database](#).

▣ **Upgrading Report Server Content**

Upgrading a report server also upgrades reports, but it does not upgrade other content that is stored in a report server. Published reports and snapshots are upgraded in a long running process at the end of Setup. Subscriptions, schedules, shared data sources, and resources are identical to pre-upgrade versions. For more information about in-place upgrades to reports, see [Upgrading Reports](#).

▣ **Upgrading Role Assignments to Use New Ad Hoc Reporting Features**

Role definitions and role assignments are not modified during an upgrade. This means that new tasks and role definitions that were introduced in SQL Server 2005 are not implemented in an upgraded installation. To use the new tasks and roles, you must update roles and role assignments manually.

New tasks and roles support the ad hoc reporting features that allow users to create models and use Report Builder. Before you can use the new functionality, you must update existing roles and role assignments. For more information, see [Updating Role-Based Security for Report Builder Access](#).

▣ **Upgrading Configuration Files**

The configuration files of the existing installation are used in the upgraded installation. New settings are added and set during the upgrade. Not all new settings are present once the upgrade is complete. Additional settings may be added when you run the Reporting Services Configuration tool.

If you are upgrading to SQL Server 2005 Service Pack 1, the configuration files will not be updated to include any new settings. The Service Pack 1 Setup does not modify configuration files. The only new configuration settings added in Service Pack 1 are for the SAP NetWeaver BI data processing extension. For more information about how to configure these settings manually, see **Configuring Reporting Services to Use the Microsoft .NET Data Provider for SAP NetWeaver Business Intelligence**.

▣ **Upgrading Reporting Services Components Together**

Reporting Services includes client and server components. If you upgrade server components, you should also upgrade all client components accordingly. This is especially true of the Report Designer. Although it is possible to run the SQL Server 2000 version of Report Designer with a SQL Server 2005 report server, you might see a disparity between report preview in Report Designer and how the report is rendered at run time. Note that once you upgrade Report Designer, you can no longer use it to publish reports to a SQL Server 2000 report server. Report namespace differences prevent publishing to both versions of the report server.

▣ **Upgrading SQL Server Components Together**

You can choose whether to upgrade the report server and Database Engine separately or together. The following list summarizes how the components are upgraded:

- If you upgrade just the Database Engine, the report server and report server database are not affected. The upgraded Database Engine hosts the existing report server database, which is defined by the old schema.
- If you upgrade just the report server, setup modifies the table structure of the report server database to the new format.
- If you upgrade both components, the Database Engine is upgraded first, followed by the report server.

For a Reporting Services installation, the upgrade process is initiated only when Setup detects the SQL Server 2000 version of the Report Server component or Report Designer.

▣ **Components Removed During an Upgrade**

During an upgrade, Setup removes any components that are replaced by new components in the SQL Server 2005 release. The following list describes which components are upgraded or removed. For more information about how to choose a component in Setup, see [Installing Reporting Services Using Setup](#).

- Report server, Report Manager, and the report server databases are upgraded.
- Report Designer is uninstalled and replaced with an installation of Business Intelligence Development Studio. In SQL Server 2005, Report Designer runs within Business Intelligence Development Studio. The previous version, which runs in Visual Studio 2003, is uninstalled and the program files are removed from your system.
- Reporting Services Books Online is removed. In the new release, product documentation is included in SQL Server 2005 Books Online.

▣ **Important:**

If you installed the documentation refresh, be sure to uninstall it before upgrading. You must do this step manually. Although the upgrade feature removes the documentation files, it does not remove the program item from **Add / Remove programs**.

- Samples are removed. New versions are available through Setup.
- Administrative command line tools are removed. Note that **rsactivate.exe** is no longer used or installed in this release.

▣ **See Also**

Concepts

[Upgrading Reporting Services](#)

Other Resources

[Reporting Services Backward Compatibility](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Upgrading Reports



[See Also](#)



[Upgrading to SQL Server 2005](#) > [Upgrading SQL Server Components](#) > [Upgrading Reporting Services](#) >

You can upgrade report definitions, published reports, and report snapshots when you upgrade a SQL Server 2000 Reporting Services installation to SQL Server 2005. SQL Server 2005 introduces changes to the Report Definition Language (RDL) and report object model that affects reports created in earlier versions of the software. Existing reports must be upgraded if they are to run on a SQL Server 2005 Reporting Services report server.

Report definition (.rdl) files are upgraded when you open them in Report Designer. Published reports and snapshots are upgraded on first use after you upgrade the report server. When a report is upgraded, it is modified in the following ways:

- The report definition namespace is upgraded to `http://schemas.microsoft.com/sqlserver/reporting/`
- **Actions** element has been modified to support multiple actions. It can now contain more than one child element.
- **CustomReportItem** element has been modified to support data-bound controls and report processing extensions. It includes child elements that describe the data that is used by the control, properties, and dimensions of the control in the report. The XML structure that you use with **CustomReportItem** is HTML-encoded upon upgrade in Report Designer, and will be decoded during report publishing to a report server.
- **Custom** element has been replaced by a custom properties collection that contains name-value pairs. Upon upgrade, all instances of a custom element are mapped to a custom property in the custom properties collection.

After you upgrade a report, you might notice additional errors, warnings, and messages. This is the result of improvements to the internal report object model and processing components, which cause messages to appear when underlying problems in the report are detected.

▣ **Report Definition (.rdl) Files and Report Designer**

Report definitions can be upgraded to a new namespace when you open an .rdl in the SQL Server 2005 version of Report Designer. You are prompted whether to upgrade the report definition when you open the report. Once you upgrade the report definition, you cannot open it in previous versions of Report Designer.

Report definitions are also upgraded when you publish or upload an .rdl file created in an earlier version of Reporting Services to a SQL Server 2005 report server. The report is published in the SQL Server 2005 version of RDL. The original file is not upgraded unless you open it in Report Designer.

▣ **Note:**

You cannot publish a report that has the SQL Server 2005 report definition namespace to a SQL Server 2000 report server.

You can run different versions of the Report Designer side-by-side on the same computer. For more information, see [How to: Install Reporting Services Side-by-Side with an Earlier Version](#).

▣ **Published Reports and Report Snapshots**

Published and stored reports are upgraded automatically, requiring no specific action on your part. Upgrading reports is performed on a first use basis. When a user opens a report or report snapshot, or when the report server processes a subscription, upgrading occurs. Once a published report or snapshot has been upgraded, you cannot run it on earlier versions of the product.

▣ **Versions Supported by Upgrade**

You can upgrade reports that were created in the following versions of Reporting Services:

- SQL Server 2000
- SQL Server 2000 with Service Pack 1
- SQL Server 2000 with Service Pack 2.
- SQL Server 2005 Beta 3
- SQL Server 2005 product release to SQL Server 2005 Service Pack 1.

▣ **See Also**

Concepts

[Upgrading Editions of Reporting Services](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Updating Role-Based Security for Report Builder Access



[See Also](#)



[Upgrading to SQL Server 2005](#) > [Upgrading SQL Server Components](#) > [Upgrading Reporting Services](#) >

New: 14 April 2006

SQL Server 2005 provides new tasks that allow users to create models and use Report Builder. If you are upgrading an existing installation, Setup will not modify existing role definitions to include the new tasks that grant access to these features. After you upgrade your installation, you must manually update existing role definitions and role assignments to include the new tasks. The following tasks are new in SQL Server 2005 Reporting Services:

Name	Type	Description
View models	Item-level task	Allows a user to view models in the folder hierarchy, use models as data sources for a report, and run queries against the model to retrieve data. This task is necessary for viewing ad hoc reports that are auto-generated by the report server and for using Report Builder.
Manage models	Item-level task	Allows a user to create, view, and delete models, and view and modify model properties. Users who can perform this task also have permission to assign model item security and map predefined static reports to parts of a model.
Consume reports	Item-level task	Allows a user to load a published report from the report server into a local Report Builder instance. Users who can perform this task can open and edit published, model-based reports.

Execute report definitions	System-level task	<p>Allows a user to process a report definition that is not currently stored on the report server.</p> <p>In the tools that are included with Reporting Services, this task is used to access Report Builder. Users who can perform this task can click the Report Builder button in Report Manager to launch Report Builder.</p> <p>When used programmatically, this task can support a broader implementation. Users who can perform this task have permission to process an unpublished report definition (.rdl) file.</p>
Browse model	Model item task	Allows a user to view a portion of a model. You can use this task to specify which parts of the model are visible to a user or group.

New installations of SQL Server 2005 Reporting Services include a new role definition named Report Builder. This role defines a collection of tasks necessary for using Report Builder. The Report Builder role is not created for you when you upgrade an existing installation. However, you can create the role definition manually if you want to have the complete set of role definitions that are available in a new Reporting Services installation.

▣ **How to Update Existing Item-Level Role Definitions**

If you are using the predefined role definitions, you can edit them to include the new tasks. Adding new tasks to existing role definitions is the easiest way to grant current users permission to view and manage Report Builder reports and models. When you add a new task to an existing role definition, all role assignments that include that role definition will get the new tasks automatically.

1. Find the role definitions that are defined on the report server. In Management Studio, the role definitions are located under the Security node in the Roles folder. In Report Manager, open the Site Settings page and click **Configure item-level role definitions**.
2. Expand the Browser role definition. Add the "View models" task to the role, and then click **OK**.
3. Expand the Content Manager role definition. Add "Manage models", "View models", and "Consume reports" to grant full permission over models, and the ability to create and modify reports in Report Builder. Click **OK**.
4. Expand the Publisher role definition. Add "Manage models" to grant permission to create, view, and delete models on the report server. Click **OK**.

▣ **How to Update Existing System-Level Role Definitions**

To make Report Builder available to users, you must add tasks to the system-level roles.

1. In Management Studio, expand the System Roles folder. In Report Manager, open the Site Settings page and click **Configure system-level role definitions**.
2. Expand the System Administrator role definition. Add the "Execute report definitions" task. Click **OK**.
3. Expand the System User role definition. Add the "Execute report definitions" task. Click **OK**.

▣ **Creating a New Report Builder Role Definition**

New installations of SQL Server 2005 Reporting Services include a new role definition that organizes the set of tasks used for working with Report Builder into a single collection of tasks. If you are upgrading an existing installation, you must create this role definition manually. Creating a separate role definition ensures that Report Builder users have only those tasks that are necessary for creating ad hoc reports.

1. Create a new role definition using either Management Studio or Report Manager. For more information, see **How to: Create, Delete, or Modify a Role (Report Manager)** or **How to: Create, Delete, or Modify a Role (Management Studio)**.
2. Name the role Report Builder.
3. Add the following tasks: Consume reports, View reports, View models, View resources, View folders, and Manage individual subscriptions.
4. Save the role definition.

▣ **Tips for Using the New Report Builder Role Definition**

If you modified existing role definition using the instructions provided earlier in this topic, only users who are mapped to the Content Manager role have sufficient permission to use Report Builder to create and save reports. To make Report Builder functionality available to more users, you must create or modify role assignments that map a group or user account to the Report Builder role that you just created.

- If possible, create a domain group account that includes all of users who need access to Report Builder. If you have a group account, you can create one role assignment for all users.
- If you cannot create a group account, you can create role assignments that map a specific user account to the role definition. If you already have an existing role definition for a specific user, you can modify it to include the new role definition. For more information, see **Creating, Modifying, and Deleting Role Assignments**.
- You can write a script that creates role assignments and run it on the report server. For more information, see **Script Samples (Reporting Services)**.

▣ **See Also**

Other Resources

Browser Role

Content Manager Role

Publisher Role

System Administrator Role

System User Role

Report Builder Role

Securing Report Builder

Securing Models

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Upgrading a Report Server Database



[See Also](#)



[Upgrading to SQL Server 2005](#) > [Upgrading SQL Server Components](#) > [Upgrading Reporting Services](#) >

New: 14 April 2006

The report server database schema can change with each new release of Reporting Services. This topic explains how to upgrade the database schema to use the latest format. Note that once you update the schema, you cannot rollback the upgrade to an earlier version. Always backup the report server database before upgrading in case you need to recreate a previous installation.

Note:

You can upgrade a report server database that runs on a SQL Server 2000 Database Engine instance without having to upgrade the Database Engine to SQL Server 2005. You can also use a SQL Server 2000 Database Engine instance to host a new report server database that you create for a SQL Server 2005 Reporting Services installation.

The report server database is upgraded in three stages:

1. The schema is upgraded automatically during setup or manually through the Reporting Services Configuration tool.
2. Metadata and security descriptions are upgraded on first use of the report server database after the schema is updated.
3. Published reports and compiled report snapshots are updated on first use. For more information, see [Upgrading Reports](#).

In addition to the report server database, a report server also uses a temporary database. The temporary database is upgraded automatically when you upgrade the report server database.

▣ **Ways to Upgrade a Report Server Database**

The report server database can be upgraded automatically or manually.

- Setup will upgrade a report server database automatically if you are running a default configuration and the report server database is installed locally.
- Setup can also upgrade a report server database automatically if you run Setup from the command prompt. If the report server database is installed on a remote SQL Server instance, or runs on a different named instance on the same computer, you can run Setup from the command prompt and use the `RSUPGRADEDATABASEACCOUNT` and `RSUPGRADEPASSWORD` arguments to specify credentials that have permission to update the schema.
- You can manually upgrade any report server database regardless of whether it runs locally or on a remote server. To upgrade a report server database manually, use the Reporting Services Configuration tool. An upgrade option on the Database Setup page runs a script that updates the schema to the most recent version.

▣ **Upgrading a Report Server Database on a Remote Server**

If you are upgrading a Reporting Services installation that includes a remote report server database, you might encounter problems when upgrading the database schema. By default, Setup uses the security token of the user who is running the Setup program to connect to the remote SQL Server instance and update the schema. If you have administrator permissions on both the local and remote computer, the database upgrade will succeed. Similarly, if you run Setup from the command prompt and specify the `RSUPGRADEDATABASEACCOUNT` and `RSUPGRADEPASSWORD` arguments for an account that has permission to modify the schema on the remote computer, the database upgrade will succeed. However, if you do not have permission to update the schema on the remote computer, the connection will be refused with the following error:

"Setup was not able to upgrade the report server database schema. You must update the database schema manually after setup is finished. To update the schema, run the Reporting Services Configuration tool, open the Database Setup page, and click Upgrade. For more information, see article 91310 in the Microsoft Knowledge Base."

The report server program files will be upgraded, but the report server database will be in the format of the previous version. The report server will be unavailable while the report server database is in the older format.

To upgrade the database manually

If Setup did not upgrade the report server database automatically, your report server installation is only partially upgraded. To complete the upgrade, you must upgrade the report server database manually by running the Reporting Services Configuration tool after Setup is finished.

1. Run the Reporting Services Configuration tool and connect to the report server that you just upgraded. For more information about how to start the tool and connect to a server, see [How to: Start Reporting Services Configuration](#).

2. Open the Database Setup page, and then click **Connect** to connect to the SQL Server instance that hosts the report server database.
3. Select the report server database that you want to upgrade.
4. Click **Upgrade** to update the database schema.

The report server will be available once these steps are complete.

▣ **See Also**

Tasks

[How to: Start Reporting Services Configuration](#)

Concepts

[Creating a Report Server Database](#)

[Upgrading Reporting Services](#)

[Upgrading a Default Installation of Reporting Services](#)

[Migrating Reporting Services](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Upgrading Editions of Reporting Services



[See Also](#)



[Upgrading to SQL Server 2005](#) > [Upgrading SQL Server Components](#) > [Upgrading Reporting Services](#) >

Updated: **14 April 2006**

This topic describes requirements and considerations when upgrading between some editions of Reporting Services. The upgrading scenarios that are discussed in this topic include the following:

- Upgrade paths for different versions and editions of Reporting Services.
- Upgrading to editions that have more features (for example, from Standard Edition to Developer or Enterprise Editions).
- Upgrading to editions that have fewer features (for example, from Evaluation to Standard Edition).

You can also upgrade from SQL Server 2005 Express Edition with Advanced Services (SQL Server Express) to other editions. For more information, see [How to: Upgrade a SQL Server Express Report Server to Other SQL Server Editions](#) and **Reporting Services in SQL Server 2005 Express Edition with Advanced Services** in SQL Server Books Online.

▣ **Upgrading from SQL Server 2000**

The following table shows the possible upgrade paths from SQL Server 2000 editions to SQL Server 2005 Reporting Services.

SQL Server 2000 Edition	SQL Server 2005 Edition
Evaluation	Evaluation
Standard	Standard, Developer, Enterprise
Developer	Developer
Enterprise	Enterprise

Other upgrade paths are possible (for example, upgrading from the Developer Edition directly to the Enterprise Edition of SQL Server 2005), but Setup will not perform all of the steps for you automatically. In this case, you can install a new instance of the edition you want to use, and then move the report server database to the new installation. For more information, see [Migrating Reporting Services](#).

Upgrading the Evaluation Edition

To upgrade the Evaluation Edition, it must not be expired. If your evaluation copy has expired, you can install a new edition and then configure the new report server instance to use the existing report server database. If your evaluation copy of the software has not expired, you can use Setup, but the upgrade must be performed as a two-step operation:

1. Upgrade SQL Server 2000 Evaluation Edition to SQL Server 2005 Evaluation Edition.
2. Upgrade SQL Server 2005 Evaluation Edition to any other SQL Server 2005 Edition.

▣ Upgrading Editions Within SQL Server 2005

You can upgrade editions within the same version of SQL Server. The following table shows the upgrade path for different editions of SQL Server 2005 Reporting Services. Each edition has software requirements that must be satisfied. An upgrade might not succeed if your system does not meet the minimum software requirements of the new edition. For more information, see [Version and Edition Upgrades](#).

SQL Server 2005	SQL Server 2005
Evaluation	Express, Workgroup, Standard, Developer, Enterprise
Express	Workgroup, Standard, Developer, Enterprise
Workgroup	Standard, Developer, Enterprise
Standard	Developer, Enterprise
Developer	Workgroup, Standard, Enterprise

▣ **Upgrading to Editions that Have More Features**

If you have an existing installation of Reporting Services Standard Edition, Express Edition, or Workgroup Edition, you can run Setup to upgrade to the Developer or Enterprise Editions. These editions include additional features. There is no feature reduction in the edition upgrade. All reports, models, and applications that you created in the Standard Edition will continue to work in the upgraded edition. For more information, see [Features Supported by the Editions of SQL Server 2005](#).

▣ **Upgrading to Editions that Have Fewer Features**

If you are upgrading the Evaluation Edition to the Standard Edition or Workgroup Edition, the following features will no longer be available in the upgraded edition:

- Data-driven subscriptions
- Report server scale-out deployment
- Support for more than four processors

If you used any of these features in the Evaluation Edition, read the following section to determine the effect on existing files and functionality when you upgrade to the Standard Edition.

Data-Driven Subscriptions

Data-driven subscriptions are deactivated (that is, they no longer run) but not deleted. You must delete the subscriptions manually. For more information, see **Creating, Modifying, and Deleting Data-Driven Subscriptions**.

Report server scale-out deployment

If you upgrade a report server instance in scale-out deployment to the Standard Edition, the report server instance you upgraded will no longer work. If you had two report server instances sharing a single report server database, you must now have a separate report server database for each instance. To use the upgraded report server, you can choose between these options:

- Create a new report server database for the report server instance that is no longer part of the scale-out deployment. For more information, see [Creating a Report Server Database](#).
- Make a copy of the existing report server database, attach it, remove the key information, and then configure the upgraded report server to use it. For more information, see [Migrating Reporting Services](#).
- Continue to use the existing report server database with the upgraded report server. You can do this by removing the key information that was created for other report server instances in the deployment. To remove key

information for report server instances that are no longer part of a scale-out deployment, you can use the Initialization page in the Reporting Services Configuration tool. For more information, see [Adding and Removing Encryption Keys for Scale-Out Deployment](#) and [Initialization \(Reporting Services Configuration\)](#).

Support for more than four processors

Setup will not upgrade an existing installation running on more than four processors to the Standard Edition. You must install a new instance and then migrate your installation.

▣ Change History

Release	History
14 April 2006	New content: <ul style="list-style-type: none"><li data-bbox="526 499 1325 579">• SQL Server 2005 Express Edition with Advanced Services

▣ **See Also**

Concepts

[Upgrading Editions of Reporting Services](#)

[Upgrading a Report Server Web Farm](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Upgrading a Report Server Web Farm



[See Also](#)



[Upgrading to SQL Server 2005](#) > [Upgrading SQL Server Components](#) > [Upgrading Reporting Services](#) >

Updated: **14 April 2006**

This topic explains how to upgrade multiple report server instances that share the same report server database. In the previous release, this deployment model was called "Report Server Web farm". That term is no longer used. Report servers that share a single database are referred to as a report server scale-out deployment. You can run a scale-out deployment on a Network Load Balanced (NLB) cluster. Configuring report server nodes for scale-out deployment is a prerequisite for doing so. If you have deployed Reporting Services in a scale-out deployment, you must upgrade each report server in the scale-out deployment. You cannot have a scale-out deployment that includes different versions of Reporting Services on individual nodes.

The following approaches can be used to upgrade a scale-out deployment:

- Perform an in-place upgrade to replace the existing installation with a newer version. You can upgrade the nodes in any order. The report server database is updated on the first upgrade. You should take the servers offline until all of the report server nodes and the database are upgraded (that is, stop IIS and the Report Server Windows service). Once you start upgrading the servers, you need to upgrade all nodes. Otherwise, the remaining report server nodes will not work until they have been upgraded to use the new database format.
- Perform a side-by-side upgrade to install a new SQL Server 2005 Reporting Services instance on the same computer as the existing installation. If you have up-time requirements, or if your installation does not meet the requirements for an in-place upgrade, you can install new instances of SQL Server 2005 Reporting Services that run alongside your existing installations. Users can continue to use the existing scale-out deployment while you configure and test the new software.

Before upgrading production servers, it is important to verify the upgrade steps on test servers so that you know what to expect.

▣ **Preparing for Upgrade**

Use the following checklist to prepare for the upgrade:

1. Back up the report server database and the symmetric key.
2. Verify that you know the user account and password used for connecting to the SQL Server Database Engine that hosts the report server database. The account must have local administrator permissions in order to perform the upgrade.
3. Back up of all configuration files (including the Machine.config and Web.config files) on each report server instance.
4. Back up log files.
5. Prepare the server for down time. If you are using Microsoft Windows Network Load Balancing, you can set host priority levels to isolate a node from new connections. This step minimizes the impact on existing user sessions.
 1. Log on to each computer.
 2. Start Network Load Balancing Manager on the first computer and open the host list.
 3. In Host priority, note the priority level of each node. Use the priority level to determine the order in which you upgrade the nodes. Upgrade the lowest priority node first.
 4. On the lowest priority node, right-click the host name, select Control Host, and click Stop. This routes all new connections to other nodes. Existing sessions continue to run. If the report server is used by many users, you might want to stop new connections several hours before you begin the upgrade to minimize the number of open sessions.
 5. Close the Network Load Balancing Manager if it is open. Close down any other MMC snap-in that is open.
6. When you are ready to bring down the report server, do the following:
 1. Stop the Windows Management Instrumentation (WMI) service.
 2. Stop the report server Windows service.
 3. Reset Internet Information Services (IIS).
 4. Stop IIS.

▣ **How to Upgrade to a SQL Server 2005 Report Server Scale-Out Deployment**

These steps assume you are installing a new instance alongside an existing installation. If you want to perform an in-place upgrade, you can follow the instructions for installing Service Pack 1.

These steps are used to upgrade a report server. If the computer also has a database engine instance, it can be upgraded at the same time, at a later date, or not at all.

1. Run Setup. In **Instance Name**, type a new name for the SQL Server 2005 instance you are installing. Do not specify **Default Instance**. If you do, you will perform an in-place upgrade of the existing installation rather than create a new report server instance.
2. After setup is complete, view the timestamp information on the program files to verify that the files were copied successfully.
3. Start the Report Server Windows service.
4. Open the reportserverservice.log files and search for "the action completed successfully". This entry confirms that the upgrade succeeded.
5. Start IIS.
6. Open a browser window and type the report server URL. Verify that the version information at the bottom of the page reflects the product version.
7. Start WMI service.

▣ How to Upgrade to Service Pack 1

Service Pack 1 is applied by running a package on an existing SQL Server 2005 installation. If you are upgrading a SQL Server 2005 scale-out deployment to Service Pack 1, you must apply the service pack to each node. Upgrading the first node will upgrade the report server database, so be sure to take all other nodes offline until all nodes are upgraded. Use the following steps to perform an in-place upgrade of a report server node to Service Pack 1.

📌 **Note:**

Upgrading the report server database does not upgrade the Database Engine that hosts it. For more information, see [Upgrading a Report Server Database](#).

1. Choose a node to upgrade first. Take all other nodes offline.
2. Stop the Windows Management Instrumentation (WMI) service, the report server Windows service, and any other SQL Server service that is part of installation you are upgrading.
3. Reset Internet Information Services (IIS), and then stop the service.
4. Run the package that installs Service Pack 1. If the computer has multiple instances, be sure that you choose the one that runs Reporting Services. If the report server database is on a remote computer, you will be prompted to specify credentials that have permission to upgrade the report server database.
5. After the upgrade is complete, view the timestamp information on the program files to verify that the files were copied successfully.
6. Start the Report Server Windows service.
7. Open the ReportServerService_<timestamp>.log files and search for "the action completed successfully". This entry confirms that the upgrade succeeded.
8. Start IIS.
9. Open a browser window and type the report server URL (by default, this value is <http://localhost/reportserver>). Verify that the version information at the bottom of the page reflects the product version you expect to see.
10. Start WMI service.

Repeat the steps for the other nodes in the scale-out deployment.

▣ Change History

Release	History
14 April 2006	Changed content: <ul style="list-style-type: none"><li data-bbox="570 499 1292 541">• Scale-out deployment Default Instance issue

▣ **See Also**

Concepts

[Upgrading Editions of Reporting Services](#)

[Upgrading a Report Server Database](#)

[Upgrading the Database Engine](#)

[Configuring a Report Server Scale-Out Deployment](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Migrating Reporting Services



[See Also](#)



[Upgrading to SQL Server 2005](#) > [Upgrading SQL Server Components](#) > [Upgrading Reporting Services](#) >

This topic provides step-by-step instructions for migrating a SQL Server 2000 Reporting Services deployment to a new SQL Server 2005 Reporting Services instance. Migrating an installation is useful if you encountered an upgrade blocker, have a large scale deployment or up-time requirements, or if you are changing the hardware or topology of your installation. The migration process for Reporting Services includes manual and automated steps. The following tasks are part of a report server migration:

- Backup application and configuration files.
- Install a new instance of SQL Server 2005 side-by-side your existing SQL Server 2000 installation. Because SQL Server 2000 Reporting Services always runs in the default SQL Server instance, the new SQL Server 2005 instance that you install must be configured as a named instance.
- Move the report server database from your SQL Server 2000 installation to your new SQL Server 2005 installation.
- Uninstall SQL Server 2000 Reporting Services after you have confirmed that the new instance is fully operational.

If you are reusing an existing report server database that you created in a previous installation, be aware that there are restrictions on the editions of SQL Server you use to host the report server database. For more information, see [Creating a Report Server Database](#).

▣ Backup Files and Data

Before you install a new instance of Reporting Services, be sure to backup all of the files in your current installation.

1. Back up the symmetric key for the report server database. Use the **rskeymgmt** utility to perform this operation.
2. Back up the report server database using any of the supported methods for backing up a SQL Server database. For more information, see the instructions on how to backup the report server database in **Moving a Report Server Database to Another Computer**.
3. Back up the report server configuration files. Files to back up include:
 1. Rsreportserver.config
 2. Rswbapplication.config
 3. Rssvrpolicy.config
 4. Rsmgrpolicy.config
 5. Reportingservice.exe.config
 6. Web.config for both the Report Server and Report Manager ASP.NET applications.
 7. Machine.config for ASP.NET.
4. Back up the report server virtual directory settings. Use Internet Services Information Manager to backup the settings.
5. For high availability report servers, set the report server database to read-only and modify the RSExec role on the SQL Server Database Engine to stop transactions in the report server databases.

▣ Install SQL Server 2005 Reporting Services

1. Run SQL Server 2005 Setup to install a new instance of Reporting Services.
2. In the Components to Install page, select **Reporting Services**. You can also select SQL Server Database Services and Workstation components, Books Online, and development tools if you want to install a new Database Engine instance and Report Designer on the same computer.
3. In Instance Name, type a new name for the SQL Server 2005 instance you are installing. Do not specify Default Instance. If you do, you will perform an in-place upgrade of the existing installation rather than create a new report server instance.
4. In Service Account, specify a service account for the Report Server Windows service. For more information about service accounts, see [Setting Up Windows Service Accounts](#).
Optionally, in Start services at the end of Setup, select SQL Server Agent (SQL Server Agent must be running to support scheduled operations).
5. In Report Server Installation Options, select the **Install but do not configure the server** option. Selecting this option allows you to configure the report server after setup is finished.
6. Finish setup.
7. If you are applying Service Pack 1, stop all SQL Server 2005 services (including the Report Server Windows service, SQL Server Agent, SQL Server Browser, and SQL Server Fulltext Search).
8. Apply Service Pack 1, and then restart the services.
9. Attach or restore the report server database you backed up from the SQL Server 2000 installation to the new instance. Both the report server database and the temporary database are required and must be moved together. Do not copy the databases; copying does not transfer all of the security settings to the new installation.
If you are using a new SQL Server 2005 Database Engine instance, you can use SQL Server Management Studio to attach or restore the databases. For instructions, see **Moving a Report Server Database to Another Computer**. If you are using SQL Server 2000 to host the report server database, create a copy of the database for the new installation.
10. Copy settings from the rsreportserver.config used in the SQL Server 2000 installation to the rsreportserver.config file of the new SQL Server 2005 installation:

1. Copy the **InstallationID** entry. Copying the installation ID is necessary if you want to avoid having two unused entries in the Keys table of the report server database. The two unused rows are the installation identifiers of the SQL Server 2000 instance. The unused rows are incomplete. Missing from each row is the Machine Name and Instance Name. Both Machine Name and Instance Name are undefined for a SQL Server 2000 instance, and as a result there are no values for those columns in the Keys table after a report server database is upgraded. Note that if an entry is missing a machine name or instance name, you cannot delete it in the Reporting Services Configuration tool. However, you can edit the Keys table to remove the rows or to add a missing machine or instance name. You can also leave any unused entries in the table; the unused entries have no effect on report server operations.
2. Copy any other custom settings that you want to use in the new installation. If you are using custom extensions in your installation, you might need to recompile the extensions before you can use them. For more information, see Upgrade Advisor product documentation.

If you have custom trace settings, copy the entries to the Web.config or Machine.config files of the new installation.

▣ **Configure a Report Server**

1. Start the Reporting Services Configuration tool and connect to the SQL Server 2005 Reporting Services instance you just installed.
2. Create virtual directories for the report server and Report Manager. Virtual directory names must be unique, so if you used the default names in the SQL Server 2000 Reporting Services installation you must choose non-default names for the SQL Server 2005 installation.
When you create the Report Server virtual directory, the Web service identity should be configured automatically. If it is not configured correctly, you can create a new application pool in which to run the service.
The Report Server Windows service should be configured automatically during Setup. If you want to run the service under a different account, you can modify it now. For more information about the service accounts, see [Setting Up Windows Service Accounts](#) and **Accounts in a Reporting Services Deployment** in SQL Server Books Online.
3. On the Server Status page, click **Stop** to stop the Report Server Windows service. This step is very important. If you do not stop the service, the database upgrade operation might time out before the schema is fully updated, leaving the database in an intermediate state.
4. On the Database Setup page, in Server Name, select the SQL Server instance that hosts the report server database that you want to upgrade and click **Connect**.
5. Choose the report server database that you want to use in the SQL Server 2005 installation.
6. Click **Upgrade**. Both the report server database and the temporary database are upgraded to the new schema.
7. Specify the account used to connect the report server to the report server database.
8. On the Encryption Keys page, restore the key that is used to encrypt and decrypt content in the report server database. This is the encryption key that you backed up in step 1 of the "Backup Files and Data" section in this topic.
9. Test the report server and Report Manager virtual directories by opening a browser and typing in the URL address. URLs and database connection information that you specify for that instance must include the instance name.
10. Test reports and verify they contain the data you expect. Review data source

information to see whether the data source connection information is still specified.

11. Uninstall SQL Server 2000 Reporting Services.
12. Rename the virtual directories to the names used in the previous installation. If you used custom virtual directory settings, update the virtual directories you created to use the new values.

▣ **See Also**

Concepts

[Upgrading Reporting Services](#)

Other Resources

Report Server Database

[Reporting Services Backward Compatibility](#)

Reporting Services Configuration Tool

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Upgrade a SQL Server Express Report Server to Other SQL Server Editions

 [See Also](#)



[Upgrading to SQL Server 2005](#) > [Upgrading SQL Server Components](#) > [Upgrading Reporting Services](#) >

New: 14 April 2006

SQL Server 2005 Express Edition with Advanced Services (SQL Server Express) includes a subset of Reporting Services functionality. To use additional features, you must upgrade to a different edition. You can upgrade SQL Server Express to the Standard, Developer, Enterprise, or Workgroup Edition of SQL Server 2005 Reporting Services. Not all Reporting Services features are available in every edition. If you are upgrading SQL Server Express because you want to use a specific feature, verify that the edition supports it. For more information, see [Features Supported by the Editions of SQL Server 2005](#).

The Express edition of Reporting Services is based on SQL Server 2005 Service Pack 1. Because it is based on Service Pack 1, the Express edition is newer than the version of Reporting Services that shipped in other editions of SQL Server 2005. The Express edition includes enhancements from Service Pack 1 and uses the Service Pack 1 version of the report server database schema. When you upgrade from Express to a different edition of SQL Server, the schema of the report server database retains the Service Pack 1 format and is not modified during the upgrade. To use the report server database from SQL Server Express in an upgraded installation, you must apply Service Pack 1 to avoid compatibility problems between the report server and the report server database.

Reporting Services has edition requirements for the SQL Server Database Engine that hosts the report server database. After the report server is upgraded, you can no longer use SQL Server Express to host a report server database. Instead, you can either choose another SQL Server Database Engine instance or you can upgrade the Database Engine instance in SQL Server Express to a compatible edition. You can use any local or remote SQL Server 2000 or SQL Server 2005 Database Engine instance that is valid for the report server edition

you are using. For more information about edition requirements for report server databases, see [Creating a Report Server Database](#).

To upgrade your installation, you must use the **SKUUPGRADE=1** argument. You can specify this argument when you start the SQL Server Setup wizard. Or, you can run Setup from the command prompt and include the **SKUUPGRADE=1** argument on the command line.

To simplify the upgrade and minimize configuration steps after setup is complete, you should upgrade Reporting Services and the Database Engine together. If you choose to upgrade only Reporting Services, you must attach the report server database to the Database Engine instance you want to use, and then configure the report server to use that database.

The following example provides an illustration of the command line that upgrades Reporting Services and Database Engine in the same operation:

```
start /wait setup.exe ADDLOCAL=SQL_Engine,RS_Server PIDKEY=
```

The example assumes you are running Setup from a folder that contains the Setup.exe program file for the edition to which you are upgrading (do not run the SQL Server Express Setup program to upgrade to a different edition). The **PIDKEY** in the example is fictitious; the actual value must be valid for the edition to which you are upgrading. For more information about running Setup from the command prompt, see [How to: Install SQL Server 2005 from the Command Prompt](#).

After upgrade is finished, you must apply Service Pack 1. Otherwise, you will get a database compatibility error message.

Note that once you begin upgrading your SQL Server Express installation, the report server and content will not be available until all of these steps are complete.

□ **Update a SQL Server Express Installation**

1. Backup the report server database and report server temporary database. By default, the databases are named **reportserver\$SQLExpress** and **reportserverTempDB\$SQLExpress**.
2. Backup the RSReportServer.config file. If you modified other configuration files (such as Web.config or RSWebApplication.config), you should back them up.
3. Backup the encryption key using the Encryption Keys page in the Reporting Services Configuration tool. For more information, see [Backing Up and Restoring Encryption Keys](#).
4. Run Setup with the **SKUUPGRADE=1** argument.

To run Setup in unattended mode from the command line, use syntax that is similar to the following example. Note that the **PIDKEY** is fictitious; be sure to replace it with a **PIDKEY** that is valid for the edition you are installing.

```
start /wait setup.exe ADDLOCAL=SQL_Engine,RS_Server PIDKEY=AE
```

To run the SQL Server Setup Wizard:

1. Open a command prompt, navigate to the directory or drive that contains the installation media, and specify `setup.exe SKUUPGRADE=1` on the command line (for example, `d:\setup.exe SKUUPGRADE=1`).
 2. On the Registration Information page, enter identification information used to register your copy of SQL Server.
 3. On the Components to Install page, select **Database Services**, **Reporting Services**, and **Workstation components, Books Online and development tools**.
 4. On the Instance Name page, click **Named instance**, and then choose the SQLEXPRESS instance you are upgrading.
 5. Complete the wizard by specifying options on the remaining pages.
5. Apply Service Pack 1 after Setup is complete.

To verify that the upgrade succeeded, open Report Manager. You should be able to view all of the reports that you previously created. If you have not changed the virtual directory, the URL used to access Report Manager in the upgraded installation will be the same URL you used in the Express installation.

Unless you rename them, an upgraded installation continues to use names from the original installation. If you used default SQL Server Express names, the upgraded SQL Server instance name continues to be `<servername>\SQLExpress`. For the databases, the default names continue to be `reportserver$sqlexpress` and `reportservertempdb$sqlexpress`. For the Reporting Services URLs, the default names continue to be `http://<servername>/reports$SQLExpress` and `http://<servername>/reportserver$SQLExpress`. To avoid confusion, you can reregister the SQL Server instance to change the instance name, and rename the report server virtual directories. Renaming a report server database is not supported. For more information, see **Registering Servers** and [Configuring Report Server Virtual Directories](#) in SQL Server Books Online.

In addition to viewing existing report server content, you should also be able to create subscriptions and scheduled operations, and use other features that are available in the edition you installed. For more information about how to create subscriptions and schedules, see **Report Manager How-to Topics**. If the report server upgrade did not succeed or seems incomplete, you can migrate the installation instead. For more information, see [Migrating Reporting Services](#).

You can specify client components on the command line to upgrade SQL Server Management Studio and Business Intelligence Development Studio. You can verify whether the tools upgraded correctly by doing the following:

- Connect to a Reporting Services instance in SQL Server Management Studio.
- Create additional project types in Business Intelligence Development Studio (specifically, the Report Model project template should be available when you create a new project in Business Intelligence Development Studio).

If the tools do not upgrade correctly, you can use **Add or Remove Programs** to uninstall the tools you no longer need, and then run SQL Server Setup to install the tools that are included in the edition you are upgrading to.

If you are migrating other SQL Server Express databases that provide data to reports to another Database Engine, be sure to update the data source connection information used by published reports after the upgrade is complete. For example, if a data source specified `Data Source=localhost\sqlexpress`, it should be updated to `Data Source=localhost` if you migrate the database to a local default instance.

▣ **See Also**

Tasks

[How to: Start Reporting Services Configuration](#)

Concepts

[Default Configuration for a Report Server Installation](#)

[Upgrading Reporting Services](#)

[Managing Encryption Keys](#)

[Configuring a Report Server Database Connection](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Upgrading or Migrating Data Transformation Services



[See Also](#)



[Upgrading to SQL Server 2005](#) > [Upgrading SQL Server Components](#) >

Note:

We recommend that you review [Version and Edition Upgrades](#) for general information about upgrade scenarios for the Microsoft SQL Server 2005 products and technologies.

If you are a previous or current user of Microsoft SQL Server 2000 Data Transformation Services (DTS), you have the following options when you install Microsoft SQL Server 2005 Integration Services (SSIS):

- [Installing Support for SQL Server 2000 Data Transformation Services Packages](#)
- [Upgrading an earlier version of Data Transformation Services to SSIS](#)
- [Migrating existing Data Transformation Services packages to SSIS](#)

Important:

Regardless of which installation option you choose, you should be aware that some SQL Server 2000 Data Transformation Services features and functionality have been discontinued, deprecated, or changed. For more information, see [SQL Server 2005 Integration Services Backward Compatibility](#).

▣ **Known Upgrade Issues**

Since Integration Services is an entirely new product, and not a version upgrade, you will not encounter software conflicts when upgrading to Integration Services. However, before upgrading Data Transformation Services to SQL Server 2005 Integration Services, we recommend that you review [SQL Server 2005 Integration Services Backward Compatibility](#).

▣ **Pre-Upgrade Checklist**

Before upgrading Data Transformation Services to SQL Server 2005 Integration Services, review the following requirements:

- [Hardware and Software Requirements for Installing SQL Server 2005](#)
- [Check Parameters for the System Configuration Checker](#)
- [Security Considerations for a SQL Server Installation](#)
- [Using Upgrade Advisor to Prepare for Upgrades](#)

▣ Installing Support for SQL Server 2000 Data Transformation Services Packages

You can run SQL Server 2000 Data Transformation Services (DTS) packages and Integration Services packages on the same computer.

When you select **Integration Services** for installation, Setup also installs support for SQL Server 2000 Data Transformation Services (DTS) packages, including the DTS runtime and DTS package enumeration in SQL Server Management Studio. Run-time support has been updated to allow DTS packages to access SQL Server 2005 data sources. If you are not installing Integration Services, but you need support for DTS packages, then you need to make sure that **Legacy Components** is selected on the **Feature Selection** page.

☒ **Note:**

Although support for SQL Server 2000 Data Transformation Services (DTS) is installed automatically when you select Integration Services for installation, some of the registry and environment settings required by DTS are removed if you upgrade the last instance of SQL Server 2000 on the computer to SQL Server 2005 during or after setup. To restore these settings, locate "Microsoft SQL Server 2005 Backward compatibility" in the **Add or Remove Programs** list in the Control Panel, click the **Change** button, and select the **Repair** option on the **Program Maintenance** page of the wizard.

The backward compatibility files that are installed also include the SQL Server 2000 Analysis Services Processing task and its dependency, Decision Support Objects (DSO). However they do not include the SQL Server 2000 Data Mining Prediction Query task.

☒ **Note:**

SQL Server 2005 does not install the DTS package designer for editing DTS packages. For more information, see [SQL Server 2005 Integration Services Backward Compatibility](#). DTS packages cannot be opened or edited in BI Development Studio.

When you select Integration Services for installation, the Package Migration Wizard is also installed. You can optionally use the Package Migration Wizard to upgrade SQL Server 2000 DTS packages to the Integration Services format. During migration, the wizard copies the DTS packages and then re-creates them in the Integration Services format. The source packages are left intact and are not modified. For more information, see [Migrating Data Transformation Services Packages](#).

 **Note:**

The Package Migration Wizard is available in the Standard, Enterprise, and Developer Editions of SQL Server 2005.

For more information, see [How to: Ensure Support for Data Transformation Services Packages](#).

▣ **Upgrading an Earlier Version of Data Transformation Services to Integration Services**

The process of upgrading SQL Server 2000 Data Transformation Services to SQL Server 2005 Integration Services consists of the installation of SQL Server 2005 Integration Services and, optionally, the migration of existing DTS packages to the SQL Server 2005 Integration Services package format. This process uses software tools in addition to Setup itself.

To upgrade an instance of SQL Server 2000 Data Transformation Services to SQL Server 2005 Integration Services, follow these steps:

1. Install SQL Server 2005 Integration Services by using the Installation Wizard or an alternate installation method. For more information, see [Installing SQL Server Integration Services by Using Setup](#) and [Installing SQL Server Integration Services from the Command Prompt](#).
2. Optionally, use the Package Migration Wizard to migrate SQL Server 2000 DTS packages to SQL Server 2005 Integration Services. This step recreates the packages in the new package format while leaving the original packages intact. When you select Integration Services for installation, the Package Migration Wizard is also installed. For more information, see [Migrating Data Transformation Services Packages](#).

For more information, see [How to: Upgrade from Data Transformation Services to Integration Services](#) and [SQL Server 2005 Integration Services Backward Compatibility](#).

▣ **Migrating Existing Data Transformation Services Packages to Integration Services**

You can optionally use the Package Migration Wizard to upgrade SQL Server 2000 DTS packages to the SQL Server 2005 Integration Services format. During migration, the wizard copies SQL Server 2000 DTS packages and then re-creates them in the SQL Server 2005 Integration Services format. The source packages are left intact and are not modified.

When you select Integration Services for installation, the Package Migration Wizard is also installed.

▣ **Note:**

The Package Migration Wizard is available in the Standard, Enterprise, and Developer Editions of SQL Server 2005.

For more information, see [Migrating Data Transformation Services Packages](#), [How to: Migrate Data Transformation Services Packages to Integration Services](#), and [Known Package Migration Issues](#).

▣ **See Also**

Concepts

[Upgrading dtsrun Command Lines](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Migrating Data Transformation Services Packages



[See Also](#)



[Upgrading to SQL Server 2005](#) > [Upgrading SQL Server Components](#) > [Upgrading or Migrating Data Transformation Services](#) >

Updated: 17 July 2006

SQL Server 2005 Integration Services (SSIS) provides several options for preserving solutions created by using the SQL Server 2000 Data Transformation Services (DTS) tools and object model. You can migrate DTS packages to the SQL Server 2005 format, continue to run DTS packages by using the SQL Server 2000 DTS runtime, or incorporate DTS packages into SQL Server 2005 Integration Services solutions by using the Execute DTS 2000 Package task.

When you migrate SQL Server 2000 DTS packages to the SQL Server 2005 Integration Services format, the structure of the migrated package and the completeness of the migration depend on the contents of the original package.

Note:

The Package Migration Wizard leaves the original DTS packages intact and unchanged.

The possible outcomes of package migration include the following:

- Packages can be migrated successfully and completely because they contain only tasks and use only features that map to SQL Server 2005 Integration Services (SSIS) tasks and features.
- Packages contain SQL Server 2000 DTS tasks and features that cannot be migrated directly to SQL Server 2005 Integration Services (SSIS) tasks and features, but package functionality is preserved through the use of encapsulated SQL Server 2000 packages.

Note:

In some cases, such as when the package name contains invalid characters (/ \ : [] . =), or when the package contains unregistered objects, the Package Migration Wizard cannot perform the migration until these issues are resolved.

If all the tasks in the package can be mapped to new tasks in SQL Server 2005, the structure of the migrated package is very similar to the structure of the SQL Server 2000 package. Conversely, if tasks do not map directly to SQL Server 2005 tasks, the migration changes the package structure. Each non-migrated task retains its SQL Server 2000 format and is encapsulated within an Execute DTS 2000 task. Custom tasks in packages are also migrated this way. For more information, see [Migrating Tasks](#).

Other package elements such as precedence constraints, connections, and variables are migrated to the equivalent element in SQL Server 2005. For more information, see [Migrating Precedence Constraints](#), [Migrating Connections](#), and [Migrating Variables](#).

Package passwords are not migrated with packages that are migrated as SQL Server 2005 Integration Services (SSIS) packages. However, SQL Server 2005 supports package passwords in SQL Server 2000 packages and within the Execute DTS 2000 Package task. An Execute DTS 2000 Package task within a SQL Server 2005 package can pass the package password to a SQL Server 2000 package. For more information, see [Migrating Passwords](#).

The ActiveX Script task in SQL Server 2005 is frequently able to run the script code that is migrated from the ActiveX scripts in SQL Server 2000 DTS packages without additional revision. However, references to SQL Server 2000 DTS package objects are not supported. For more information, see [Migrating Scripts](#).

Text annotations entered in the DTS Designer, and package logging and error handling settings, are not migrated.

After migrating packages, the migrated packages should be opened to check for issues, including validation failures. For example, some packages rely on preceding tasks within the same package to create or prepare data sources or data destinations, and may fail validation when these objects cannot be located. A

validation failure of this sort can be resolved by setting the **DelayValidation** property to **true** on the task or other container object, or by setting the **ValidateExternalMetadata** property to **false** on the affected data flow component.

 **Note:**

After migrating packages from Management Studio, connect to the local instance of Integration Services in Object Explorer to see the migrated packages. If you selected SQL Server as the destination, the migrated packages will be visible under the **MSDB** node. If you selected a file system folder as the destination, right-click the **File System** node and then select **Import Package** to display the migrated packages. After you migrate packages from BI Development Studio, the migrated packages are saved to the file system folder that you specified and added to the open project under the **SSIS Packages** node.

DTS packages cannot be opened or edited in BI Development Studio.

▣ In This Section

Topic	Description
Using the Package Migration Wizard	Describes the migration process that the Package Migration Wizard provides.
Known Package Migration Issues	Lists known package migration issues that the Package Migration Wizard or the Upgrade Advisor may identify.
Migrating Tasks	Describes how the Package Migration Wizard migrates tasks, and explains the steps that you can take after migration to repair tasks or enhance task behavior.
Migrating Precedence Constraints	Describes how the Package Migration Wizard migrates precedence constraints, and explains the steps that you can take after migration to enhance precedence constraint behavior.
Migrating Variables	Describes how the Package Migration Wizard migrates variables, and explains the steps that you can take after migration to repair or enhance variable behavior.
Migrating Connections	Describes how the Package Migration Wizard migrates connections, and explains the steps that you can take after migration to edit connection properties.
Migrating Passwords	Describes how the Package Migration Wizard migrates passwords, and explains other steps that you can take after migration to improve security.
Migrating Scripts	Describes how the Package Migration Wizard migrates ActiveX scripts.

Change History

Release	History
17 July 2006	<p>Changed content:</p> <ul style="list-style-type: none"> • Provided note about location of migrated packages.

▣ **See Also**

Concepts

[SQL Server 2005 Integration Services Backward Compatibility](#)

Other Resources

SQL Server Integration Services (SSIS)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Using the Package Migration Wizard



[See Also](#)



[Upgrading SQL Server Components](#) > [Upgrading or Migrating Data Transformation Services](#) > [Migrating Data Transformation Services Packages](#) >

Updated: 17 July 2006

SQL Server 2005 Integration Services (SSIS) provides the Package Migration Wizard for migrating Data Transformation Services (DTS) packages that were created by using the SQL Server 2000 tools.

Note:

The Package Migration Wizard is available in the Standard, Enterprise, and Developer Editions of SQL Server 2005.

Note:

Upgrade Advisor cannot analyze, and the Package Migration Wizard cannot migrate, DTS packages that are saved in the **msdb** database of an instance of SQL Server 7.0. However the Upgrade Advisor can analyze, and the Package Migration Wizard can migrate, SQL Server 7.0 DTS packages that are saved as structured storage files, or that are saved in the **msdb** database of an instance of SQL Server 7.0 that has been upgraded to SQL Server 2005.

Not all packages can be migrated completely. For example, some SQL Server 2000 tasks do not map to Integration Services tasks, custom tasks cannot be migrated, and some scripts in ActiveX Script tasks or in data transformations cannot be migrated. For more information, see [Migrating Tasks](#) and [Known Package Migration Issues](#).

Packages left in the SQL Server 2000 package format can be used in Integration Services solutions and can also continue to run as stand-alone SQL Server 2000

packages.

Many SQL Server 2000 packages are password-protected. If you migrate these packages, the Package Migration Wizard prompts you for their passwords. You cannot migrate a password-protected package unless you provide the correct password.

SQL Server 2005 does not install support for Meta Data Services. The Package Migration Wizard supports Meta Data Services (Repository) packages only if SQL Server 2000, the SQL Server 2000 tools, or the Repository redistributable files are installed on the local computer. When the Repository files are present, the Package Migration Wizard can migrate DTS packages that were saved to Meta Data Services. When the Repository files are not present, the Package Migration Wizard can only migrate DTS packages that were saved to SQL Server or to structured storage files.

▣ Start the Package Migration Wizard

You can start the Package Migration Wizard from three locations.

- From SQL Server Management Studio. Connect to an instance of the SQL Server 2005 Database Engine, right-click the **Data Transformation Services** node in Object Explorer under **Management\Legacy**, and select **Migration Wizard**.
- From Business Intelligence Development Studio. Create or open an Integration Services Project, right-click on the **SSIS Packages** node in Solution Explorer, and select **Migrate DTS 2000 Package**.
- From the command prompt. Start **DTSMigrationWizard.exe** from the **C:\Program Files\Microsoft SQL Server\90\DTS\Binn** folder.

▣ **Select the Package Source**

The Package Migration Wizard can migrate SQL Server 2000 packages that were saved to a SQL Server **msdb** database, to structured storage files, or to Meta Data Services. Packages that were saved to Visual Basic files cannot be migrated.

Important:

SQL Server 2005 does not install support for Meta Data Services. The Package Migration Wizard can only access DTS packages saved to Meta Data Services if SQL Server 2000, the SQL Server 2000 tools, or the Repository redistributable files are present on the computer.

If a package is located in an **msdb** database or in Meta Data Services, the user provides a server name, and the Package Migration Wizard authenticates the user by using Windows Authentication or SQL Server Authentication.

If a package is located in a structured storage file, you select the file to migrate. A structured storage file can contain definitions for many packages and can include multiple versions of a package. You can migrate some or all the packages and package versions in the file.

For more information, see **Choose Source Location (Package Migration Wizard)**.

▣ **Select the Package Destination**

The Package Migration Wizard can migrate packages either to an **msdb** database in an instance of SQL Server 2005, or to a .dtsx file. The .dtsx file is an XML representation of the package definition that is saved to the file system.

If the destination is SQL Server 2005, the user provides a server name, and the Package Migration Wizard authenticates the user by using Windows Authentication or SQL Server Authentication.

For more information, see **Choose Destination Location (Package Migration Wizard)**.

📌 **Note:**

After migrating packages from within Management Studio, connect to the local instance of Integration Services in Object Explorer to see the migrated packages. If you selected SQL Server as the destination, the migrated packages will be visible under the **MSDB** node. If you selected a file system folder as the destination, right-click the **File System** node and select **Import Package** to display the migrated packages. After migrating packages from within BI Development Studio, the migrated packages are saved to the file system folder that you specified and added to the open project under the **SSIS Packages** node.

DTS packages cannot be opened or edited in BI Development Studio.

▣ **Select Packages to Migrate**

After the source and destination have been specified, the Package Migration Wizard lists the packages available in the **msdb** database, in Meta Data Services, or in the specified structured storage file. You can select the packages to migrate and update the default names that the Package Migration Wizard provides for the packages. You can also update the package names in Business Intelligence Development Studio or SQL Server Management Studio after the packages have been migrated.

For more information, see **List Packages (Package Migration Wizard)**.

▣ **Provide Package Passwords**

Packages developed by using earlier versions of SQL Server tools may be password-protected. You must supply the correct password before the wizard can access their package definitions and migrate the packages. If multiple packages share a password, you have to provide the password only one time. The Package Migration Wizard cannot continue until you have provided passwords for all password-protected packages. If no packages are password-protected, the wizard skips this step.

▣ **Log Migration Errors**

You can enable the Package Migration Wizard to log errors that it encounters during migration.

For more information, see **Specify a Log File (Package Migration Wizard)**.

▣ **View Migration Progress**

The Package Migration Wizard migrates the selected packages one by one. If the wizard cannot migrate a package, it provides options to end the migration process for all remaining packages, or to skip the package and continue migration with the next package in the list. As each package is migrated, its migration status is updated.

For more information, see **Complete the Wizard (Package Migration Wizard)** and **Migrating the Packages (Package Migration Wizard)**.

▣ Check Migrated Packages

After migrating packages, the migrated packages should be opened in Business Intelligence Development Studio to check for issues, including validation failures. For example, some packages rely on preceding tasks within the same package to create or prepare data sources or data destinations, and may fail validation when these objects cannot be located. A validation failure of this sort can be resolved by setting the **DelayValidation** property to **true** on the task or other container object, or by setting the **ValidateExternalMetadata** property to **false** on the affected data flow component.

📌 Note:

After migrating packages from within Management Studio, connect to the local instance of Integration Services in Object Explorer to see the migrated packages. If you selected SQL Server as the destination, the migrated packages will be visible under the **MSDB** node. If you selected a file system folder as the destination, right-click the **File System** node and then select **Import Package** to display the migrated packages. After you migrate packages from BI Development Studio, the migrated packages are saved to the file system folder that you specified and added to the open project under the **SSIS Packages** node.

▣ Change History

Release	History
17 July 2006	Changed content: <ul style="list-style-type: none">• Noted that DTS packages cannot be opened in BI Development Studio.
5 December 2005	Changed content: <ul style="list-style-type: none">• Added note about where to find migrated packages.• Added information about support for migration from SQL Server 7.0.

▣ **See Also**

Other Resources

[Migrating Data Transformation Services Packages](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Known Package Migration Issues



[See Also](#)



[Upgrading SQL Server Components](#) > [Upgrading or Migrating Data Transformation Services](#) > [Migrating Data Transformation Services Packages](#) >

Updated: 17 July 2006

This is a summary of the issues that you may experience when migrating Microsoft SQL Server 2000 Data Transformation Services (DTS) packages to the SQL Server 2005 Integration Services (SSIS) format by using the Package Migration Wizard.

The Package Migration Wizard migrates certain SQL Server 2000 DTS features that have no direct equivalent in SQL Server 2005 Integration Services (SSIS) by encapsulating them in an Execute DTS 2000 Package task. Although in many cases you can run these migrated packages without error, you must eventually replace the functionality encapsulated within the Execute DTS 2000 Package task.

After migrating packages, migrated packages should be opened to check for issues, including validation failures. For example, some packages rely on preceding tasks within the same package to create or prepare data sources or data destinations, and may fail validation when these objects cannot be located. A validation failure of this sort can be resolved by setting the **DelayValidation** property to **true** on the task or other container object, or by setting the **ValidateExternalMetadata** property to **false** on the affected data flow component.

▣ Known Issues Detected by Upgrade Advisor

The following list presents the migration issues and a brief description of each as they appear in the report generated by the SQL Server 2005 Upgrade Advisor. The Package Migration Wizard and Upgrade Advisor identify and report the same potential package migration issues.

Issue Title	Issue Description
Reconfigure transaction settings after package migration	<p>The SQL Server 2005 Integration Services (SSIS) Package Migration Wizard does not migrate transaction settings in SQL Server 2000 Data Transformation Services (DTS) packages. After migration, you must manually edit the resulting SSIS package to restore former package behavior.</p> <p>Corrective action: After migration, re-create the desired transaction settings on the package by setting the appropriate transaction-related properties on the package and its components in Business Intelligence Development Studio. For more information, see Incorporating Transactions in Packages.</p>
Replace functionality of ActiveX script attached to package steps after package migration	<p>The SQL Server 2005 Integration Services (SSIS) Package Migration Wizard does not migrate ActiveX scripts associated with steps in SQL Server 2000 Data Transformation Services (DTS) packages. After migration, you must manually edit the resulting SSIS package to restore former package behavior.</p> <p>Corrective action: After migration, replace the missing script with a new Script task or other Integration Services objects as appropriate. For more information, see Script Task.</p>
Replace functionality of Analysis Services task encapsulated	<p>The SQL Server 2005 Integration Services (SSIS) Package Migration Wizard does not migrate the Analysis Services task in SQL Server 2000 Data Transformation Services (DTS) packages. The migrated package will contain encapsulated DTS functionality which must eventually be replaced.</p>

<p>in Execute DTS 2000 Package task after package migration</p>	<p>Corrective action: After migration, convert the Execute DTS 2000 Task that encapsulates the Analysis Services task to an Integration Services Analysis Services Processing task. For more information, see Analysis Services Processing Task.</p>
<p>Replace functionality of complex Data Transformation task encapsulated in Execute DTS 2000 Package task after package migration</p>	<p>The SQL Server 2005 Integration Services (SSIS) Package Migration Wizard does not migrate complex Data Transformation tasks in SQL Server 2000 Data Transformation Services (DTS) packages. The migrated package will contain encapsulated DTS functionality which must eventually be replaced.</p> <p>Corrective action: After migration, rewrite the complex Data Transformation task by using Integration Services data flow sources, transformations, and destinations. For more information, see Creating Package Data Flow and Data Flow Elements.</p>
<p>Replace functionality of custom task encapsulated in Execute DTS 2000 Package task after package migration</p>	<p>The SQL Server 2005 Integration Services (SSIS) Package Migration Wizard does not migrate custom tasks in SQL Server 2000 Data Transformation Services (DTS) packages. The migrated package will contain encapsulated DTS functionality which must eventually be replaced.</p> <p>Corrective action: After migration, re-create the logic of the custom task by using a Script task or Script component, or a custom component, if the desired functionality is not available in a built-in Integration Services component or combination of components. For information about programming solutions, see Integration Services Programming.</p>
<p>Replace functionality of Data Driven Query task encapsulated in Execute DTS</p>	<p>The SQL Server 2005 Integration Services (SSIS) Package Migration Wizard does not migrate Data Driven Query tasks in SQL Server 2000 Data Transformation Services (DTS) packages. The migrated package will contain encapsulated DTS functionality that must eventually be replaced.</p> <p>Corrective action: After migration, re-create the</p>

<p>2000 Package task after package migration</p>	<p>functionality of the Data Driven Query task by using Integration Services data flow components. Consider the Slowly Changing Dimension Transformation, the Conditional Split Transformation, or the OLE DB Command Transformation when replacing the functionality of Data Driven Query tasks.</p>
<p>Replace functionality of Data Mining Prediction Query task encapsulated in Execute DTS 2000 Package task after package migration</p>	<p>The SQL Server 2005 Integration Services (SSIS) Package Migration Wizard does not migrate the Data Mining Prediction Query task in SQL Server 2000 Data Transformation Services (DTS) packages. The migrated package will contain encapsulated DTS functionality which must eventually be replaced. SQL Server 2005 does not install support for the SQL Server 2000 Data Mining Prediction Query task; therefore the migrated package will not run after upgrade in the absence of SQL Server 2000 Analysis Services.</p> <p>Corrective action: After migration, convert the Execute DTS 2000 Task that encapsulates the Data Mining Prediction Query task to an Integration Services Data Mining Query task or Data Mining Query transformation. For more information, see Data Mining Query Task and Data Mining Query Transformation.</p>
<p>Replace functionality of Dynamic Properties task after package migration</p>	<p>The SQL Server 2005 Integration Services (SSIS) Package Migration Wizard does not migrate Dynamic Properties tasks in SQL Server 2000 Data Transformation Services (DTS) packages. After migration, you must manually edit the resulting SSIS package to restore former package behavior.</p> <p>Corrective action: After migration, re-create the functionality of the Dynamic Properties task by using Integration Services features such as variables, property expressions, and package configurations. For more information, see Integration Services Variables and Using Variables in Packages; Using Property Expressions in Packages; and Package Configurations.</p>
	<p>The SQL Server 2005 Integration Services (SSIS) Package Migration Wizard does not migrate Execute Packages tasks that load Meta Data Services packages. The migrated</p>

<p>Replace functionality of Execute Package task (with Meta Data Services package) encapsulated in Execute DTS 2000 Package task after package migration</p>	<p>package will contain encapsulated DTS functionality which must eventually be replaced. SQL Server 2005 does not install support for SQL Server 2000 Meta Data Services; therefore the migrated package will not run after upgrade in the absence of the SQL Server 2000 Client Tools.</p> <p>Corrective action: After migration, move the Meta Data Services package to SQL Server storage (in the msdb database) or to a structured storage file, and convert the Execute DTS 2000 Task that encapsulates the Execute Package task that loads a Meta Data Services package to an Execute DTS 2000 Task that loads the package from SQL Server or from a structured storage file. For more information, see Saving Packages.</p>
<p>Replace functionality of Parallel Data Pump task encapsulated in Execute DTS 2000 Package task after package migration</p>	<p>The SQL Server 2005 Integration Services (SSIS) Package Migration Wizard does not migrate Parallel Data Pump tasks in SQL Server 2000 Data Transformation Services (DTS) packages. The migrated package will contain encapsulated DTS functionality which must eventually be replaced.</p> <p>Corrective action: After migration, re-create the functionality of the Parallel Data Pump task by using Integration Services data flow components. For more information, see Creating Package Data Flow and Data Flow Elements.</p>
<p>Replace functionality of UDL connection encapsulated in Execute DTS 2000 Package task after package migration</p>	<p>The SQL Server 2005 Integration Services (SSIS) Package Migration Wizard does not migrate tasks that obtain their connection information from Microsoft Data Link (.udl) files. The migrated package will contain encapsulated DTS functionality which must eventually be replaced.</p> <p>Corrective action: After migration, convert the Execute DTS 2000 Task that encapsulates the UDL file connection to an equivalent Integration Services task that uses an Integration Services connection manager. For more information, see Connection Managers and Creating Connection Managers.</p>
	<p>The SQL Server 2005 Integration Services (SSIS) Package</p>

Rewrite ActiveX script code that access the DTS object model after package migration

Migration Wizard does not migrate ActiveX Script code that accesses the DTS object model through the Parent property of the GlobalVariables collection in SQL Server 2000 Data Transformation Services (DTS) packages. After migration, you must manually edit the resulting SSIS package to restore former package behavior.

Corrective action: After migration, modify the script code, or rewrite the ActiveX Script task to obtain the required information from the package in a different way: for example, by using variables. For more information, see **Integration Services Variables** and **Using Variables in Packages**.

▣ **Known Issues Not Detected by Upgrade Advisor**

The following issues are not detected or reported by the Package Migration Wizard or Upgrade Advisor.

Issue Title	Issue Description
Delete and recreate ODBC connections after package migration	ODBC connections raise an error if you try to edit the connection after migration. Corrective action: If you have already migrated the package, you can delete and recreate the ODBC connection. This issue has been fixed in SQL Server 2005 Service Pack 1. Install Service Pack 1 or later to prevent this issue from recurring.

Change History

Release	History
17 July 2006	Changed content: <ul style="list-style-type: none">• Added issue with ODBC connections.• Added links for more information to each issue.

▣ **See Also**

Concepts

[SQL Server 2005 Integration Services Backward Compatibility](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Migrating Tasks

 [See Also](#)



[Upgrading SQL Server Components](#) > [Upgrading or Migrating Data Transformation Services](#) > [Migrating Data Transformation Services Packages](#) >

Many SQL Server 2000 Data Transformation Services (DTS) tasks map directly to SQL Server 2005 Integration Services (SSIS) tasks, and migrating a package to SQL Server 2005 Integration Services (SSIS) replaces the SQL Server 2000 version of a task with its SQL Server 2005 Integration Services (SSIS) equivalent.

The following table lists the direct mappings between task versions.

SQL Server 2000 task	SQL Server 2005 task
ActiveX Script task	ActiveX Script Task
Bulk Insert task	Bulk Insert Task
Copy SQL Server Objects task	Transfer SQL Server Objects Task
Data Mining Prediction task	Data Mining Query Task
Execute Package task	Execute DTS 2000 Package Task
Execute Process task	Execute Process Task
Execute SQL task	Execute SQL Task

File Transfer Protocol task	File Transfer Protocol Task
Message Queue task	Message Queue Task
Send Mail task	Send Mail Task
Transfer Databases, Transfer Error Messages, Transfer Jobs, Transfer Logins, and Transfer Master Stored Procedures tasks	Transfer SQL Server Objects Task, Transfer Database Task, Transfer Error Messages Task, Transfer Jobs Task, Transfer Logins Task, and Transfer Master Stored Procedures Task

 **Note:**

ActiveX Script tasks that try to access the package object model by using the **Parent** property of the **GlobalVariables** collection will fail after package migration.

Some SQL Server 2000 tasks do not map directly to SQL Server 2005 Integration Services (SSIS) tasks:

- Analysis Services task (because the SQL Server 2005 Integration Services (SSIS) Analysis Services task does not work with a SQL Server 2000 instance of Analysis Services)
- Data Driven Query task
- Dynamic Properties task
- Transform Data task

In SQL Server 2005, some of these tasks are replaced by non-task features. The following table lists the mapping between the SQL Server 2000 tasks and their equivalent SQL Server 2005 non-task feature.

SQL Server 2000 task	SQL Server 2005 feature

Dynamic Properties	Configurations and property expressions. For more information, see Creating Configurations .
Transform Data	Data Flow Task and Data Flow Components .
Data Driven Query	Consider the Slowly Changing Dimension Transformation , the Conditional Split Transformation , and the OLE DB Command Transformation when replacing the functionality of Data Driven Query tasks.

▣ Custom Tasks

A package that contains custom tasks is migrated to SQL Server 2005 in a special manner. A SQL Server 2000 package is created for each custom task, and an Execute DTS 2000 Package task that calls the SQL Server 2000 package is added to the migrated package. The new SQL Server 2000 packages, called intermediate packages, are embedded within the migrated package and are not saved separately. For example, a SQL Server 2000 package with three custom tasks becomes a SQL Server 2005 package that contains three Execute DTS 2000 Package tasks, where each task points to an embedded SQL Server 2000 package that encapsulates a custom task.

▣ **Data Pump Tasks**

Three SQL Server 2000 tasks — Data Driven Query, Transform Data, and ParallelDataPump — are based on the SQL Server 2000 DTS data pump. With the exception of simple data transformation tasks, these tasks are migrated in the same way as custom tasks. The Package Migration Wizard creates a new SQL Server 2000 package for each task, and adds an Execute DTS 2000 Package task for each data pump task to the new package.

▣ **Dynamic Properties Task**

Dynamic Properties tasks cannot be migrated to SQL Server 2005. The Package Migration Wizard warns you if you try to migrate a package with a Dynamic Properties task. If you choose to migrate the package, the Dynamic Properties task is replaced by a placeholder task in the migrated package. The placeholder task is a Script task that lists the property-value pairs from the Dynamic Properties task that you may want to convert to script in the Script task. This information is written to the Script task in the form of comments in the script.

In SQL Server 2005, configurations replace and enhance the functionality available in the Dynamic Properties task. For more information, see **Creating Configurations**.

▣ **See Also**

Other Resources

[Migrating Data Transformation Services Packages](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Migrating Precedence Constraints



[See Also](#)



[Upgrading SQL Server Components](#) > [Upgrading or Migrating Data Transformation Services](#) > [Migrating Data Transformation Services Packages](#) >

Precedence constraints in a SQL Server 2000 Data Transformation Services (DTS) package migrate to the **PrecedenceConstraints** collection of the new package.

▣ **Extending Functionality of Precedence Constraints**

After the package is migrated, you can extend precedence constraints in the following ways:

- Use an expression with, or instead of, the execution result of the preceding executable to specify the conditions under which the constrained task runs. For more information, see **Adding Expressions to Precedence Constraints**.
- Combine multiple precedence constraints to specify the conditions under which the constrained task runs. For more information, see **Applying Multiple Precedence Constraints**.

▣ **See Also**

Other Resources

[Migrating Data Transformation Services Packages
Precedence Constraints](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Migrating Variables



[See Also](#)



[Upgrading SQL Server Components](#) > [Upgrading or Migrating Data Transformation Services](#) > [Migrating Data Transformation Services Packages](#) >

In SQL Server 2005 Integration Services (SSIS), a variables collection is available on all containers: packages, looping structures such as the Foreach container, and the TaskHost container that encapsulates each task. When a package is migrated, the global variables and the variables from SQL Server 2000 DTS tasks are all migrated to the variables collection of the package. For example, the variables used by a SQL Server 2000 Execute SQL task are added to the variables collection of the migrated package.

SQL Server 2005 Integration Services (SSIS) uses namespaces to organize variables. All DTS variables are migrated to the **User** namespace.

▣ **Extending Functionality of Variables**

After the package is migrated, you can add functionality to variables in the following ways:

- Create additional namespaces and organize variables by namespace.
- Use expressions to set variable values.
- Specify whether updating the variable value raises an event.

For more information, see **DTS Variables** and **Using Variables in Packages**.

▣ **See Also**

Other Resources

[Migrating Data Transformation Services Packages](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Migrating Connections



[See Also](#)



[Upgrading SQL Server Components](#) > [Upgrading or Migrating Data Transformation Services](#) > [Migrating Data Transformation Services Packages](#) >

Connections in SQL Server 2000 Data Transformation Services (DTS) packages are migrated to SQL Server 2005 and converted to connection managers of the appropriate connection type.

A connection manager is a logical representation of a connection to a data source. The connection manager specifies the attributes of the physical connection that the run-time engine creates when the package runs. A migrated package can have multiple connection managers, all of the same connection type, that are configured differently. For more information, see **Connection Managers**.

The only SQL Server 2000 task that uses a connection and can be migrated to SQL Server 2005 without issues is the Execute SQL task. Its connections are migrated as OLE DB or ODBC connection managers. Any properties of the existing connection managers that were not previously set are set to the default values in the new connection manager. For more information, see **OLE DB Connection Manager** and **ODBC Connection Manager**.

Note:

In some circumstances, the SQL Server 2000 data transformation task can be migrated successfully together with its connections. The Data Driven Query task and the Parallel Data Pump task are migrated into Execute DTS 2000 Package tasks. For more information, see [Migrating Tasks](#) and [Known Package Migration Issues](#).

Any connections that were used in SQL Server 2000 tasks but cannot be migrated to SQL Server 2005 remain part of the SQL Server 2000 intermediate packages that are created as part of the migration. For example, a connection in the SQL Server 2000 Transform Data task remains part of the SQL Server 2000

intermediate package that encapsulates the Transform Data task.

▣ **Updating Properties of Connection Managers**

The connection managers include properties that are set to default values.

The following table lists these properties.

Property	Description
DelayValidation	A value that specifies whether the validation of the connection is delayed until the package runs. This property is set to False by the Package Migration Wizard.
Description	A description of the connection manager. This property is left blank by the Package Migration Wizard.

▣ **See Also**

Other Resources

[Migrating Data Transformation Services Packages](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Migrating Passwords



[See Also](#)



[Upgrading SQL Server Components](#) > [Upgrading or Migrating Data Transformation Services](#) > [Migrating Data Transformation Services Packages](#) >

SQL Server 2005 only uses passwords on packages in connection with certain package protection level options. Therefore, password-protected packages that are fully migrated from SQL Server 2000 to SQL Server 2005 do not retain passwords, although you must provide the password during the migration process. For information about package protection levels in SQL Server 2005, see **Setting the Protection Level of Packages**.

 **Note:**

Password-protected packages cannot be migrated unless you provide their passwords.

If the DTS run-time engine for SQL Server 2000 is installed, you can continue to run SQL Server 2000 packages. You can also include these packages in SQL Server 2005 data transformation solutions. SQL Server 2005 continues to support package passwords in this scenario. For example, the Execute DTS 2000 Package task includes a password property for passing a password to a SQL Server 2000 package.

▣ **Adding Security Features to Packages**

After a package is migrated, you can add the following security features:

- Set the protection level on the package to encrypt all or only sensitive property values when the package is saved. For more information, see **DTS Package**.
- Digitally sign the package and set properties to ensure that the signature is checked when the package is loaded. For more information, see **How to sign a DTS package**.
- Encrypt checkpoints on packages that use checkpoints. For more information, see **Using Checkpoints in Packages**.

For more information about securing the Integration Services service and Integration Services packages, see **Security Considerations for Integration Services**.

▣ **See Also**

Other Resources

[Migrating Data Transformation Services Packages](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Migrating Scripts



[See Also](#)



[Upgrading SQL Server Components](#) > [Upgrading or Migrating Data Transformation Services](#) > [Migrating Data Transformation Services Packages](#) >

The SQL Server 2000 version of the ActiveX Script task can be migrated to the SQL Server 2005 ActiveX Script task. However the SQL Server 2005 ActiveX Script task is provided only for backward compatibility. After migration, you should replace the VBScript with managed Visual Basic .NET code in the SQL Server 2005 Integration Services (SSIS) Script Task.

ActiveX Script tasks that try to access the package object model by using the **Parent** property of the **GlobalVariables** collection will fail after package migration. The former functionality must be replaced by using different script code or new SQL Server 2005 features.

▣ **See Also**

Other Resources

ActiveX Script Task

Script Task

[Migrating Data Transformation Services Packages](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Upgrading dtsrun Command Lines



[See Also](#)



[Upgrading to SQL Server 2005](#) > [Upgrading SQL Server Components](#) > [Upgrading or Migrating Data Transformation Services](#) >

New: 5 December 2005

The SQL Server 2000 **dtsrun.exe** command prompt utility has been replaced in SQL Server 2005 by **dtexec.exe**.

Similarly, the SQL Server 2000 **dtsrunui.exe** utility that helps build command lines visually has been replaced in SQL Server 2005 by **dtexecui.exe**.

▣ **Upgrading Command Lines**

If you want to convert your **dtsrun** command lines by mapping each command prompt option to the corresponding **dtexec** option, see **dtsrun to dtexec Command Option Mapping**.

- Some **dtsrun** command-line options have direct **dtexec** equivalents, such as the options for providing a server name or package name, or for setting the value of a variable.
- Some **dtsrun** command-line options do not have direct **dtexec** equivalents, such as the option to specify a trusted connection, which is now the implicit default, or the option to load from the Repository, which is no longer supported.
- There are some new **dtexec** command-line options that support new features in SQL Server 2005 Integration Services (SSIS), such as the options to pass in connection strings and to manage checkpoints.

If you want to create new command lines visually with the assistance of a user interface, use the **dtexecui** tool, which displays all the available options and ensures the use of the correct syntax.

For more information about the **dtexec** command options and examples of **dtexec** syntax, see **dtexec Utility**.

▣ See Also

Other Resources

dtexec Utility

How to: Run a Package Using the dtexec Utility

How to: Run a Package Using the dtexecui Utility

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Migrating Notification Services Instances



[See Also](#)



[Upgrading to SQL Server 2005](#) > [Upgrading SQL Server Components](#) >

Microsoft SQL Server Setup does not automatically upgrade instances of Notification Services 2.0. Instead, you upgrade the Database Engine and install Microsoft SQL Server 2005 Notification Services, and then migrate the instances of Notification Services. Also, Setup does not automatically update instance metadata when you upgrade to a higher edition of SQL Server 2005. Instead, you must "upgrade" the metadata using Notification Services tools.

This topic describes migration options and provides links to additional information based on your migration scenario.



Important:

We strongly recommend that you test the migration steps on test servers and then back up your production databases before migrating production instances of Notification Services.

▣ **Known Migration Issues**

Before migrating Notification Services 2.0 instances to SQL Server 2005 Notification Services, review any upgrade and migration issues. For more information, see [Notification Services Backward Compatibility](#).

▣ **Preparing to Migrate to SQL Server 2005**

Before upgrading the Database Engine and migrating instances of Notification Services, review the following topics:

- [Hardware and Software Requirements for Installing SQL Server 2005](#)
- [Using Upgrade Advisor to Prepare for Upgrades](#)
- [Check Parameters for the System Configuration Checker](#)
- [Security Considerations for a SQL Server Installation](#)

▣ **Migrating Instances of Notification Services**

Notification Services stores instance and application metadata and operational data in databases. The Notification Services instance version must match the Database Engine instance version:

- If you have instances of Notification Services 2.0, and you want to upgrade to SQL Server 2005, you must upgrade the Database Engine, install SQL Server 2005 Notification Services, and then migrate the instances of Notification Services. For more information, see [Migrating Notification Services 2.0 Instances to SQL Server 2005](#).
- If you have instances of SQL Server 2005 Notification Services, and you want to upgrade to a higher edition of SQL Server 2005, you must "upgrade" the instance metadata. For more information, see [Migrating Notification Services Instances to a Higher Edition of SQL Server 2005](#).

▣ Side-by-Side Installations

You can install Notification Services 2.0 and SQL Server 2005 Notification Services side by side. However, within a Notification Services instance, the Database Engine version and Notification Services version must match. Also, custom components must be built referencing the same version of Notification Services and the associated version of Microsoft .NET Framework.

If the version information in the Microsoft Windows Registry and the database do not match, the instance of Notification Services does not start. Therefore, after migrating an instance of Notification Services, you should verify the version information. You can display version information in SQL Server Management Studio and using the command prompt utilities.

- [How to: View Notification Services Version Information](#)
- **nscontrol listversions Command**

▣ **Downgrading Instances of Notification Services**

If, after migrating instances of Notification Services, you need to revert back to an earlier version, you must restore the previous databases from backups and then reregister the instance of Notification Services using the matching version of the Notification Services utilities.

▣ **See Also**

Concepts

[Installing SQL Server 2005](#)

[Installing SQL Server Notification Services](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Migrating Notification Services 2.0 Instances to SQL Server 2005



[See Also](#)



[Upgrading to SQL Server 2005](#) > [Upgrading SQL Server Components](#) > [Migrating Notification Services Instances](#) >

The version of a Notification Services instance and its databases must match. Instances of Notification Services 2.0 must use the Microsoft SQL Server 2000 databases. Instances of Microsoft SQL Server 2005 Notification Services must use the SQL Server 2005 databases.

SQL Server 2000 databases are upgraded by upgrading the Database Engine or by moving the databases to a SQL Server 2005 database server. SQL Server 2005 Notification Services is installed side by side with Notification Services 2.0. Instances of Notification Services are not automatically upgraded when you run Setup. Instead, you must migrate Notification Services instances to SQL Server 2005 by performing the following steps:

- Upgrade the Database Engine.
- Install SQL Server 2005 Notification Services.
- Upgrade Notification Services instance metadata.
- Update the Notification Services instance.

▣ Migration Matrix

The following table shows possible migration paths from Notification Services 2.0 to SQL Server 2005 Notification Services. "Yes" indicates that you can migrate from the version in the left column to the version at the top of the column. "No" indicates that the migration fails.

Current version	SQL Server 2005 Evaluation Edition	SQL Server 2005 Developer Edition	SQL Server 2005 Workgroup Edition	SQL Server 2005 Standard Edition	SQL Server 2005 Enterprise Edition
Notification Services 2.0 Standard Edition	Yes ¹	Yes ²		Yes	Yes
Notification Services 2.0 Enterprise Edition	Yes ¹	Yes ²		No	Yes

¹ SQL Server 2005 Evaluation Edition has a time limit. When the time limit expires, the instance of Notification Services will not start.

² Note the license restrictions with SQL Server 2005 Developer Edition.

Note Notification Services does not support SQL Server 2005 Express Edition and SQL Server Workgroup Edition.

▣ Preparing to Migrate Instances

Before upgrading an existing instance of Notification Services to SQL Server 2005, perform the following tasks:

- Recompile custom event providers, content formatters, and delivery protocols using the SQL Server 2005 Notification Services assemblies and Microsoft .NET Framework 2.0. Custom components compiled with the Notification Services 2.0 assembly and .NET Framework 1.1 are not compatible with SQL Server 2005 Notification Services.
The SQL Server 2005 assemblies for Notification Services are located in the *%ProgramFiles%\Microsoft SQL Server\90\NotificationServices*n.n.nnn*\bin* folder, where *n.n.nnn* is the build number.
- Uninstall the Notification Services 2.0 sample applications.
Unmodified Notification Services 2.0 samples will not run against SQL Server 2005. The syntax of the rules have changed; rules that use the **Notify()** function are not supported in SQL Server 2005.
Also, some sample applications use the same names in Notification Services 2.0 and SQL Server 2005; therefore, you must remove the Notification Services 2.0 samples before using SQL Server 2005 samples of the same name.
Run the **RemoveSamplesDB.cmd** utility, which is located in the *v2.0.NNNN.0\Samples* folder, to remove the 2.0 sample applications.
- Ensure that applications do not have multiple distributors on the same server. This is not allowed with SQL Server 2005. If necessary, update the ADF to move one of the distributors and then update the application before migrating to SQL Server 2005.
- Determine the best time to perform the upgrade.
Upgrade instances during periods of low activity to minimize the impact on event collection, notification generation and distribution, and subscription management. You should also try to avoid upgrading during a scheduled vacuuming interval, which often occurs during periods of low activity. The vacuuming schedule is defined in each application definition file (ADF) in Coordinated Universal Time (UTC).
- Prepare a backup device for the instance and application databases.
During upgrade, you are instructed to back up your instance and application

databases. Having backup devices defined will reduce the time required to perform upgrades.

▣ **How to Migrate Instances**

You can migrate Notification Services instances to SQL Server 2005 using SQL Server Management Studio or command prompt utilities:

- [How to: Migrate Notification Services 2.0 Instances to SQL Server 2005 \(SQL Server Management Studio\)](#)
- [How to: Migrate Notification Services 2.0 Instances to SQL Server 2005 \(Command Prompt\)](#)

▣ **See Also**

Concepts

[Migrating Notification Services Instances](#)

[Installing SQL Server Notification Services](#)

[How to: View Notification Services Version Information](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Migrating Notification Services Instances to a Higher Edition of SQL Server 2005

 [See Also](#)



[Upgrading to SQL Server 2005](#) > [Upgrading SQL Server Components](#) > [Migrating Notification Services Instances](#) >

If you are running Notification Services on Microsoft SQL Server 2005, you can migrate your Notification Services instances to a higher edition of SQL Server 2005. The following list shows the editions of SQL Server 2005 supported by Notification Services in order from lowest to highest edition:

- SQL Server 2005 Evaluation Edition
- SQL Server 2005 Developer Edition
- SQL Server 2005 Standard Edition
- SQL Server 2005 Enterprise Edition

You can migrate SQL Server 2005 Standard Edition instances to Evaluation Edition or Developer Edition even though they are lower on the list. Evaluation Edition and Developer Edition have the same Notification Services functionality as Enterprise Edition; allowing migration from Standard Edition to these editions allows you to evaluate the features of Enterprise Edition.

SQL Server 2005 Notification Services binaries are upgraded to a higher edition of SQL Server 2005 by running SQL Server 2005 Setup. However, instances of Notification Services are not automatically migrated when you run Setup. You must migrate the instances to the higher edition after running Setup using the following process:

- Install the higher edition of SQL Server 2005 Notification Services.
- Upgrade Notification Services instance metadata.

▣ Migration Matrix

The following table shows possible migration paths. "Yes" indicates that you can migrate from the version in the left column to the version at the top of the column. "No" indicates that the migration fails.

	Evaluation Edition	Developer Edition	Standard Edition	Enterprise Edition
Evaluation Edition	-- ⁴	Yes ²	Yes ³	Yes
Developer Edition	No	-- ⁴	Yes ³	Yes
Standard Edition	Yes ¹	Yes ²	-- ⁴	Yes
Enterprise Edition	No	No	No	-- ⁴

¹ SQL Server 2005 Evaluation Edition has a time limit. When the time limit expires, the instance of Notification Services will not start.

² Note the license restrictions with SQL Server 2005 Developer Edition.

³ Note that not all features are supported in SQL Server 2005 Standard Edition. The instance update, performed at the end of the migration process, will flag any features in the application that are not supported by Standard Edition.

⁴ Migration succeeds, but no changes are made because the source and target version and edition are the same.

▣ **Preparing to Migrate**

Before migrating instances of Notification Services from a lower to a higher edition of SQL Server 2005, perform the following tasks:

- Determine the best time to perform the migration.
Migrate instances during periods of low activity to minimize the impact on event collection, notification generation and distribution, and subscription management. You should also try to avoid migrating during a scheduled vacuuming interval, which often occurs during periods of low activity. The vacuuming schedule is defined in application definition files (ADFs) in Coordinated Universal Time (UTC) format.
- Prepare a backup device for the instance and application databases.
During migration, you are instructed to back up your instance and application databases. Having backup devices defined reduces the time required to migrate instances.

▣ **How to Migrate**

You can migrate instances of Notification Services to another edition of SQL Server 2005 using SQL Server Management Studio or command prompt utilities:

- [How to: Migrate Notification Services Instances to a Higher Edition \(SQL Server Management Studio\)](#)
- [How to: Migrate Notification Services Instances to a Higher Edition \(Command Prompt\)](#)

▣ **See Also**

Concepts

[How to: View Notification Services Version Information](#)

[Migrating Notification Services Instances](#)

[Installing SQL Server Notification Services](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Upgrading Replicated Databases



[See Also](#)



[Upgrading to SQL Server 2005](#) > [Upgrading SQL Server Components](#) >

Updated: **17 July 2006**

SQL Server 2005 supports upgrading replicated databases from previous versions of SQL Server; it is not required to stop activity at other nodes while a node is being upgraded. Ensure that you adhere to the rules regarding which versions are supported in a topology:

- SQL Server version 7.0 service pack 4 (SP4) is the minimum version required to participate in a replication topology with SQL Server 2005. If you use SQL Server 2000, service pack 3 (SP3) or greater is required.
- A Distributor can be any version as long as it is greater than or equal to the Publisher version (in many cases the Distributor is the same instance as the Publisher).
- A Publisher can be any version as long as it less than or equal to the Distributor version.
- Subscriber version depends on the type of publication:
 - A read-only Subscriber to a transactional publication can be any version within two versions of the Publisher version. For example: a SQL Server version 7.0 Publisher running can have SQL Server 2005 Subscribers; and a SQL Server 2005 Publisher can have SQL Server version 7.0 Subscribers.
 - An updatable Subscriber to a SQL Server 2005 transactional publication can be any version equal to or greater than SQL Server 2000 SP3.
 - A Subscriber to a merge publication can be any version less than or equal to the Publisher version.

Note:

This topic is available in the Setup Help documentation and in SQL Server 2005 Books Online. Topic links that appear as bold text in the Setup Help

documentation refer to topics that are only available in Books Online.

▣ **Upgrading to Standard Edition, Workgroup Edition, or Express Edition**

Before upgrading from one edition of SQL Server 2005 to another, verify that the functionality you are currently using is supported in the edition to which you are upgrading. For more information, see the section "SQL Server 2005 Replication Features" in the topic [Features Supported by the Editions of SQL Server 2005](#).

▣ **New Replication Agent Security Model**

In previous versions of SQL Server, agents ran, by default, under the context of the SQL Server Agent service account. SQL Server 2005 allows fine-grained control over each account under which the replication agents run and make Windows Integrated connections to databases and other resources; a different account can be specified for each agent. For more information, see **Security Considerations for Replication** and **Replication Agent Security Model**.

The new security model has the following implications for upgrading and running more than one version of SQL Server in a topology:

- Replication scripts created from SQL Server 2000 or SQL Server 7.0 should be upgraded for SQL Server 2005 in order to take advantage of security enhancements. For more information, see **How to: Upgrade Replication Scripts (Replication Transact-SQL Programming)**.
- A Distributor or Subscriber upgraded from a previous version of SQL Server to SQL Server 2005 continues to run under the SQL Server Agent account and likely has more privileges than it requires. After upgrade we recommend that you specify separate accounts for the agents with the appropriate minimum privileges. To specify separate accounts:
 1. Script out the publication and subscriptions.
 2. Make changes to the scripts. For more information, see **How to: Upgrade Replication Scripts (Replication Transact-SQL Programming)**.
 3. Drop the publication and subscriptions. For more information, see **Publishing Data and Database Objects** and **Subscribing to Publications**.
 4. Recreate them using the modified scripts.

For information on the privileges required by agents, see **Replication Agent Security Model**; for information on managing logins and passwords, see **Managing Logins and Passwords in Replication**. New replication configurations created after an upgrade requires specific account configuration for each replication agent.

☒ **Note:**

Any agents configured to use SQL Server Authentication for local database connections is changed to use Windows Authentication. Local

connections are those connections made by an agent to an instance of SQL Server running on the same computer as the agent. For example, the Merge Agent for a pull subscription runs at the Subscriber, so the connections it makes to the Subscriber are local connections.

- Participants in a replication topology running previous versions of SQL Server retain the previous replication security model unchanged. For example:
 - A pull subscription to a Subscriber running SQL Server 2000 or SQL Server version 7.0 does not use the new security model because the Merge Agent or Distribution is created at the Subscriber.
 - A push subscription from a Distributor running SQL Server 2005 to a Subscriber running SQL Server 2000 or SQL Server version 7.0 uses the new security model because the Merge Agent or Distribution is created at the Distributor.
 - A Publisher running SQL Server 2000 or SQL Server version 7.0 with a Distributor running SQL Server 2005 does not use the new security model (for the Snapshot Agent, Log Reader Agent, or Queue Reader Agent) because the agents are created in the context of the publication database.

▣ **Running Agents After Upgrade for Merge Replication**

After upgrade, run the Snapshot Agent for each merge publication and the Merge Agent for each subscription to update replication metadata. You do not have to apply the new snapshot, because it is not necessary to reinitialize subscriptions. Subscription metadata is updated the first time the Merge Agent is run after upgrade. This means that the subscription database can remain online and active during the Publisher upgrade.

Merge replication stores publication and subscription metadata in a number of system tables in the publication and subscription databases. Running the Snapshot Agent updates publication metadata and running the Merge Agent updates subscription metadata. It is only required to generate a publication snapshot. If a merge publication uses parameterized filters, each partition also has a snapshot. It is not necessary to update these partitioned snapshots. (In SQL Server 2000, *parameterized filters* were referred to as *dynamic filters*, and *partitioned snapshots* were referred to as *dynamic snapshots*).

Run the agents from Microsoft SQL Server Management Studio, Replication Monitor, or from the command line. For more information about running the Snapshot Agent, see the following topics:

- **How to: Create and Apply the Initial Snapshot (SQL Server Management Studio)**
- **How to: Start and Stop a Replication Agent (SQL Server Management Studio)**
- **How to: Create the Initial Snapshot (Replication Transact-SQL Programming)**
- **Programming Replication Agent Executables**

For more information about running the Merge Agent, see the following topics:

- **How to: Synchronize a Pull Subscription (SQL Server Management Studio)**
- **How to: Synchronize a Push Subscription (SQL Server Management Studio)**

- **How to: Synchronize a Pull Subscription (Replication Programming)**
- **How to: Synchronize a Push Subscription (Replication Programming)**

After upgrading SQL Server in a topology that uses merge replication, change the publication compatibility level of any publications if you want to use new features. For more information, see [Using Multiple Versions of SQL Server in a Replication Topology](#).

▣ **Web Synchronization for Merge Replication**

The Web synchronization option for merge replication requires that the SQL Server Replication Listener (replisapi.dll) be copied to the virtual directory on the Internet Information Services (IIS) server used for synchronization. When you configure Web synchronization, the file is copied to the virtual directory by the Configure Web Synchronization Wizard. If you upgrade the SQL Server components installed on the IIS server, you must manually copy replisapi.dll from the COM directory to the virtual directory on the IIS server. For more information about configuring Web synchronization, see **Configuring Web Synchronization**.

▣ **Restoring a Replicated Database from a Previous Version**

To ensure replication settings are retained when restoring a backup of a replicated database from a previous version: restore to a server and database with the same names as the server and database at which the backup was taken.

▣ Change History

Release	History
17 July 2006	Changed content: <ul style="list-style-type: none"><li data-bbox="438 499 1372 583">• Added a statement that subscription databases may remain online and active during a Publisher upgrade.

▣ **See Also**

Reference

[Version and Edition Upgrades](#)

Concepts

[Replication Backward Compatibility](#)

[Using Multiple Versions of SQL Server in a Replication Topology](#)

Other Resources

[Administering Replication](#)

[Replication Enhancements](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Upgrading SQL Server Components How-to Topics



[See Also](#)



[Upgrading to SQL Server 2005](#) > [Upgrading SQL Server Components](#) >

The topics in this section provide information about upgrading or migrating existing Microsoft SQL Server components to Microsoft SQL Server 2005.

▣ In This Section

- [Upgrading the Database Engine How-to Topics](#)
- [Upgrading or Migrating Analysis Services How-to Topics](#)
- [Migrating Notification Services How-to Topics](#)
- [Upgrading or Migrating Data Transformation Services How-to Topics](#)

▣ **See Also**

Tasks

[How to: Install SQL Server 2005 from the Command Prompt](#)

Concepts

[Preparing to Install SQL Server 2005](#)

[SQL Server Surface Area Configuration](#)

Other Resources

[Installing SQL Server Components How-to Topics](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Upgrading the Database Engine How-to Topics



[See Also](#)



[Upgrading to SQL Server 2005](#) > [Upgrading SQL Server Components](#) > [Upgrading SQL Server Components How-to Topics](#) >

The topics in this section describe how to upgrade a Microsoft SQL Server stand-alone or clustered instance to SQL Server 2005.

▣ In This Section

- [How to: Upgrade to SQL Server 2005 \(Setup\)](#)
- [How to: Install SQL Server 2005 from the Command Prompt](#)
- [How to: Upgrade to a SQL Server 2005 Failover Cluster \(Setup\)](#)

▣ **See Also**

Reference

[Version and Edition Upgrades](#)

Other Resources

[Upgrading SQL Server Components](#)

[Backward Compatibility](#)

[Installing SQL Server 2005 Components](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Upgrade to SQL Server 2005 (Setup)



[See Also](#)



[Upgrading SQL Server Components](#) > [Upgrading SQL Server Components How-to Topics](#) > [Upgrading the Database Engine How-to Topics](#) >

Updated: **5 December 2005**

The Microsoft SQL Server 2005 Installation Wizard is Windows Installer-based, and provides a single feature tree for upgrade of SQL Server components. You can also install SQL Server 2005 side by side with SQL Server 2000 or SQL Server 7.0, or migrate existing databases and configuration settings from a previous SQL Server version, and apply them to an instance of SQL Server 2005.

For more information about upgrade and migration options, see [Upgrading SQL Server Components](#). For more information about support for side-by-side configurations, see [Working with Multiple Versions of SQL Server](#).

The Database Engine

To upgrade an instance of the SQL Server 2000 or SQL Server 7.0 Database Engine, see the procedure below.

To upgrade a failover cluster, see [How to: Upgrade to a SQL Server 2005 Failover Cluster \(Setup\)](#).

To migrate user databases to SQL Server 2005, see:

- **Using the Copy Database Wizard**
- **How to: Upgrade to SQL Server 2005 with the Copy Database Wizard**

Analysis Services

For information about upgrading Analysis Services, see [Upgrading or Migrating Analysis Services](#).

Reporting Services

Use SQL Server Setup (Installation Wizard) and the Reporting Services Configuration tools to upgrade Reporting Services. For more information, see [Upgrading Reporting Services](#).

Notification Services

To upgrade Notification Services, upgrade the Database Engine and install SQL Server 2005 Notification Services, and then migrate the instances of Notification Services. For more information, see [Migrating Notification Services Instances](#).

Data Transformation Services

For information about the use of DTS with SQL Server 2005, see [Upgrading or Migrating Data Transformation Services](#).

Replication

SQL Server 2005 supports upgrading replicated databases from previous versions of SQL Server. For more information, see [Upgrading Replicated Databases](#).

▣ Prerequisites

For local installations, you must run Setup as an administrator. If you install SQL Server from a remote share, you must use a domain account that has read and execute permissions on the remote share.

Before upgrading the Database Engine to SQL Server 2005, review the following topics:

- [Hardware and Software Requirements for Installing SQL Server 2005](#)
- [Check Parameters for the System Configuration Checker](#)
- [Security Considerations for a SQL Server Installation](#)
- [SQL Server 2005 Database Engine Backward Compatibility](#)
- [Upgrading the Database Engine](#)

□ Procedure

□ To upgrade the SQL Server Database Engine to SQL Server 2005

1.

Insert the SQL Server 2005 product CD or DVD into your computer's CD or DVD drive. The SQL Server Installation Wizard should start automatically. If the SQL Server Installation Wizard does not start automatically, double-click **Splash.hta** in the root folder of the CD or DVD.

☑ Note:

To run Setup from a network drive, navigate to the installation location on the shared drive, and then double-click **Splash.hta**.

- To begin the installation process, click **Install SQL Server**.
- On the **End User License Agreement** page, read the license agreement, and then select the check box to accept the licensing terms and conditions. Accepting the license agreement activates the **Next** button. To continue, click **Next**. To end Setup, click **Cancel**.
- On the **SQL Server Component Update** page, Setup installs software required for SQL Server 2005. For more information about component requirements, click the **Help** button at the bottom of the page. To begin the component update process, click **Install**. To continue after the update completes, click **Finish**. For more information, see [Installing Prerequisites](#).
- On the **Welcome** page of the SQL Server Installation Wizard, click **Next** to continue.
- On the **System Configuration Check (SCC)** page, the installation computer is scanned for conditions that may block Setup. For information about

configuration check items, click **Help** at the bottom of the page or see [Check Parameters for the System Configuration Checker](#). To interrupt the scan, click **Stop**. To display a list of check items grouped by result, click **Filter** and then select a category from the drop-down list. To view a report of SCC results, click **Report** and then select an option from the drop-down list. Options include viewing the report, saving the report to a file, copying the report to the Clipboard, and sending the report as e-mail. To proceed with Setup after the SCC scan completes, click **Continue**.

- On the **Registration Information** page, enter information in the **Name** and **Company** text boxes. For this release, the product key is entered automatically. To continue, click **Next**. For more information, see [Registration Information](#).
- On the **Components to Install** page, select the components for your installation. A description for each component group appears in the **Components to be Installed** pane when you select it. You can select any combination of check boxes. When you select SQL Server or Analysis Services, if Setup detects that you are installing to a virtual server, the **Install as a Failover Cluster** check box is enabled. You must select this option to install a failover cluster. For more information, see [Components to Install](#).

To install individual components, click **Advanced**. Otherwise, click **Next** to continue.

- If you clicked **Advanced** on the previous page, the **Feature Selection** page is displayed. On the **Feature Selection** page, select the program features to install using the drop-down boxes. To install components to a custom directory, select the feature and then click **Browse**. For more information about the functionality of this page, see [Feature Selection](#). To continue, click **Next** when your feature selections are complete.
- On the **Instance Name** page, select a default or named instance to upgrade. If a default or named instance is already installed, and you select an existing instance for your installation, Setup upgrades it and provides the option to install additional components.

To upgrade a SQL Server default instance already installed on your computer, click **Default Instance**, and then click **Next** to continue.

To upgrade a SQL Server named instance already installed on your computer,

click **Named Instance** and then type the instance name in the space provided, or click **Installed Instances** on the **Instance Name** page, select an instance from the **Installed Instances** list and click **OK** to populate the instance name field. After you have selected the instance to upgrade, click **Next** to continue.

 **Note:**

To upgrade a default instance, there must be a default instance already on the computer. To upgrade a named instance, there must be a named instance already on the computer.

For more information about instance naming rules, see [Instance Name](#).

- On the **Existing Components** page, Setup lists the SQL Server components installed on your computer. Components that can be upgraded to SQL Server 2005 have their check boxes enabled. If a component has a check box that is unavailable, the component does not qualify for upgrade to SQL Server 2005. To view a report of available options and alternatives, click **Details**. To upgrade a component listed on the **Existing Components** page, select its check box and click **Next**. For more information, see [Existing Components](#).
- If you are adding new components, on the **Service Account** page, specify the user name, password, and domain name for the non-SQL Server accounts. For this release, SQL Server 2005 Setup uses the service account information of the existing SQL Server service being upgraded. You can use the same account for all of the services.

You can optionally specify an individual account for each service. Select the **Customize for each service account** check box, select a service name from the list box, and then provide login credentials for each of the services in the list. To proceed, click **Next**. For more information, see [Service Account](#).

 **Note:**

The domain name cannot be a full Domain Name Service (DNS) name. For example, if your DNS name is my-domain-name.com, enter "my-domain-name" in the **Domain** text box. SQL Server Setup does not accept "my-domain-name.com" in the **Domain** text box.

- The **Upgrade Logon Information** page is displayed if the SQL Server

instance to be upgraded is configured to use Mixed Mode (Windows Authentication or SQL Server Authentication). Credentials supplied on this page are used to connect to the existing SQL Server instance so that upgrade scripts can be run. If the existing SQL Server instance is configured to use Windows Authentication, this page is not displayed. To proceed, click **Next**. For more information, see [Upgrade Logon Information](#).

 **Security Note:**

When possible, use Windows Authentication.

 **Security Note:**

For this release, SQL Server Setup may not enforce the strong password requirement on some default configurations of Microsoft Windows Server 2003 where the computer is not a member of a domain. Setting strong passwords is essential to the security of your system. Always use strong passwords. For strong password guidelines, see [Authentication Mode](#) in SQL Server 2005 Books Online.

- If you selected Analysis Services as a feature to install, on the **Collation Settings** page, specify the default collation. You can use the same collation for SQL Server and Analysis Services, or you can specify collations for individual components.

To set separate collation settings for SQL Server and Analysis Services, select the **Customize for each service account** check box. After you select the check box, a drop-down selection box appears. Select a service from the drop-down selection box and then assign its collation. Repeat for each service. To proceed, click **Next**. For more information, see [Collation Settings](#).

- If you selected Reporting Services as a feature to install, the **Report Server Virtual Directories** page is displayed. Accept the default directory names, or specify a location for the directories. For more information about this page, click **Help** at the bottom of the page. To proceed, click **Next**. For more information, see [Report Server Installation Options](#).
- If you selected Reporting Services as a feature to install, the **Report Server Database Setup** page is displayed. Specify a host for the report server database. If you choose to host the report server database in an existing SQL Server

instance, select an instance from the drop-down list, and then select the authentication mode to access that instance. For more information, see [Report Server Installation Details](#).

If you select **SQL Server Authentication**, provide a user name and password with administrative privileges for that instance. **Confirm the Report Server Database Name**. To continue, click **Next**.

- If you selected Reporting Services as a feature to install, the **Report Server Delivery Settings** page is displayed. Specify an SMTP server address and an e-mail address to be used as the sender of e-mail messages from Report Server. To continue, click **Next**.
- On the **Error and Usage Report Settings** page, optionally clear the check box to disable error reporting. To proceed, click **Next**. For more information, see [Error and Usage Report Settings](#).
- On the **Ready to Install** page, review the summary of features and components for your SQL Server installation. For an upgrade scenario, all components and features of the existing instance are selected for the upgrade. To proceed, click **Install**. For more information, see [Ready to Install](#).
- On the **Installation Progress** page, you can monitor installation progress as Setup proceeds. To view the log file for a component during installation, click the product or status name on the **Installation Progress** page.
- On the **Completing the Microsoft SQL Server Installation Wizard** page, you can view the Setup summary log by clicking the link provided on this page. To exit the SQL Server Installation Wizard, click **Finish**.
- If you are instructed to restart the computer, do so now. It is important to read the message from the Setup program when you are done with installation. Failure to restart the computer may cause failures when you run the Setup program in the future.

▣ Next Steps

After upgrading the Database Engine to SQL Server 2005, complete the following tasks:

- **Register your servers** - Upgrade removes registry settings for the previous SQL Server instance. After upgrading, you must reregister your servers.
- **Repopulate full-text catalogs** - The upgrade process marks your databases as full-text disabled. Catalogs must be repopulated, but this operation is not run automatically by Setup because it can be time-consuming. As this operation enhances the performance of your SQL Server 2005 installation, administrators should plan to repopulate full-text catalogs at a convenient time.
- **Update statistics** - To help optimize query performance, we recommend that you update statistics on all databases following upgrade. Use the **sp_updatestats** stored procedure to update statistics in user-defined tables in SQL Server 2005 databases.
- **Update usage counters** - In earlier versions of SQL Server, the values for the table and index row counts and page counts can become incorrect. To correct any invalid row or page counts, we recommend that you run DBCC UPDATEUSAGE on all databases following upgrade.
- **Configure your new SQL Server installation** - To reduce the attackable surface area of a system, SQL Server 2005 selectively installs and activates key services and features. For more information on how to activate SQL Server 2005 features, see [SQL Server Surface Area Configuration](#).

▣ **See Also**

Tasks

[How to: Change Components in an Existing Instance of SQL Server 2005 \(Setup\)](#)

Concepts

[How to: View SQL Server 2005 Setup Log Files](#)

Other Resources

[Upgrading to SQL Server 2005 Backward Compatibility](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Upgrading or Migrating Analysis Services How-to Topics



[See Also](#)



[Upgrading to SQL Server 2005](#) > [Upgrading SQL Server Components](#) > [Upgrading SQL Server Components How-to Topics](#) >

This section contains the following common procedures for upgrading and migrating databases to Microsoft SQL Server 2005 Analysis Services (SSAS).

For common procedures for installing Analysis Services, see [Analysis Services Installation How-to Topics](#).

Important:

The Analysis Services upgrade and migration how-to topics in this section focus only on upgrade and migration options that are specific to Analysis Services. However, these topics provide links to other topics for more information about installation of SQL Server.

▣ **In This Section**

- [How to: Upgrade an Instance of Analysis Services](#)
- [How to: Migrate Analysis Services Databases](#)

▣ **See Also**

Concepts

[Upgrading or Migrating Analysis Services](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Upgrade an Instance of Analysis Services



[See Also](#)



[Upgrading SQL Server Components](#) > [Upgrading SQL Server Components How-to Topics](#) > [Upgrading or Migrating Analysis Services How-to Topics](#) >

You can upgrade an existing instance of Microsoft SQL Server 2000 Analysis Services to Microsoft SQL Server 2005 Analysis Services (SSAS) and, as part of the upgrade process, migrate existing databases from the old instance to the new instance. To upgrade an existing instance of SQL Server 2000 Analysis Services, run Setup and specify the name of the existing instance as the name of the new instance.

Important:

This topic and the accompanying Analysis Services upgrade and migration how-to topics focus only on upgrade and migration options that are specific to Analysis Services. However, these topics provide links to other topics for more information about installation of SQL Server.

To upgrade an instance of Analysis Services

1. Run Setup either by using the SQL Server Installation Wizard or from the command prompt. For more information about how to run Setup by using the SQL Server Installation Wizard, see [How to: Install Analysis Services by Using Setup](#). For more information about how to run Setup from the command prompt, see [How to: Install Analysis Services from the Command Prompt](#).

Important:

When upgrading from a 64-bit edition of Microsoft SQL Server 2000 to a 64-bit edition of Microsoft SQL Server 2005, you must upgrade Analysis Services before upgrading the Database Engine.

2. When you use the SQL Server Installation Wizard, select **Default Instance** on the **Instance Name** page. To confirm the presence of the SQL Server 2000 Analysis Services instance, you can click the **Installed Instances** button. When you run Setup from the command prompt, use the **instancename** option to specify the name of an existing instance of SQL Server 2000 Analysis Services. Because SQL Server 2000 Analysis Services does not support named instances, the name of the existing instance is usually the name of the server.

▣ **See Also**

Tasks

[How to: Migrate Analysis Services Databases](#)

Concepts

[Upgrading or Migrating Analysis Services](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Migrate Analysis Services Databases



[See Also](#)



[Upgrading SQL Server Components](#) > [Upgrading SQL Server Components How-to Topics](#) > [Upgrading or Migrating Analysis Services How-to Topics](#) >

You can migrate existing Analysis Services databases from an instance of Microsoft SQL Server 2000 Analysis Services to an instance of Microsoft SQL Server 2005 Analysis Services (SSAS) either during Setup, by upgrading an existing instance of Analysis Services, or after Setup, by running the Migration Wizard.

This topic describes the process of migrating existing Analysis Services databases by using the Migration Wizard. When you upgrade an instance of SQL Server 2000 Analysis Services during Setup by using the SQL Server Installation Wizard, you do not have to specify any additional options related to database migration.

Important:

This topic and the accompanying Analysis Services upgrade and migration how-to topics focus only on upgrade and migration options that are specific to Analysis Services. However, these topics provide links to other topics for more information about installation of SQL Server.

To migrate existing Analysis Services databases by using the Migration Wizard

1. Start the Migration Wizard from an Analysis Services server node in the Object Browser in SQL Server Management Studio. You can also start the wizard at the command prompt, by running the program **MigrationWizard.exe**.
2. The wizard starts and displays the **Welcome to the Analysis Services**

Migration Wizard page. Read the introductory message and then click **Next** to continue.

3. On the **Specify Source and Destination** page, identify the source SQL Server 2000 Analysis Services server and instance name, and then identify the destination SQL Server 2005 Analysis Services server and instance name. Instead of designating a destination server, you can also choose to save the database schema to a script file and complete the migration later—for example, by using an Analysis Services Execute DDL task in a SQL Server 2005 Integration Services (SSIS) package. For more information about the Analysis Services Execute DDL task, see **Analysis Services Execute DDL Task**.
4. On the **Select Databases to Migrate** page, select the check boxes next to the databases that you want to migrate. You can specify different names for the destination databases if you want.
5. On the **Validating Databases** page, the wizard analyzes the databases to be migrated and reports any issues that it discovers.
6. On the **Migrating Databases** page, the wizard reports its progress as it performs the database migration.
7. On the **Completing the Wizard** page, the wizard reports the results of migration. Click **Finish** to complete the wizard.
8. After you migrate a database, you must process the database from the original data source before you can query the database. For more information, see **Processing Analysis Services Objects**.
9. After migration, you may want to review [Migration Considerations \(SSAS\)](#) to understand the differences between Analysis Services database versions.

▣ **See Also**

Concepts

[Migrating Existing Analysis Services Databases](#)

[Migration Considerations \(SSAS\)](#)

[Upgrading or Migrating Analysis Services](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Upgrading or Migrating Data Transformation Services How-to Topics

 [See Also](#)



[Upgrading to SQL Server 2005](#) > [Upgrading SQL Server Components](#) > [Upgrading SQL Server Components How-to Topics](#) >

This section contains the following common procedures used for upgrading and migrating packages to Microsoft SQL Server 2005 Integration Services (SSIS).

For common procedures used for installing Integration Services, see [Integration Services Installation How-to Topics](#).

▣ In This Section

- [How to: Upgrade from Data Transformation Services to Integration Services](#)
- [How to: Migrate Data Transformation Services Packages to Integration Services](#)

▣ **See Also**

Concepts

[Upgrading or Migrating Data Transformation Services](#)

[SQL Server 2005 Integration Services Backward Compatibility](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Upgrade from Data Transformation Services to Integration Services

 [See Also](#)



[Upgrading SQL Server Components](#) > [Upgrading SQL Server Components How-to Topics](#) > [Upgrading or Migrating Data Transformation Services How-to Topics](#) >

SQL Server 2000 Data Transformation Services (DTS) packages can continue to run alongside SQL Server 2005 Integration Services (SSIS) packages.

For more information about the upgrade and migration options available to users of Data Transformation Services, see [Upgrading or Migrating Data Transformation Services](#).

To upgrade from Data Transformation Services to Integration Services

1. Install SQL Server 2005, including Integration Services, by using the preferred method. For more information, see [How to: Install Integration Services by Using Setup](#) or [How to: Install Integration Services from the Command Prompt](#).
2. Integration Services also installs support for SQL Server 2000 Data Transformation Services (DTS), including the ability to enumerate and run DTS packages. For more information, see [How to: Ensure Support for Data Transformation Services Packages](#).

Note:

Although support for SQL Server 2000 Data Transformation Services (DTS) is installed automatically when you select Integration Services for installation, some of the registry and environment settings required by DTS are removed if you upgrade the last instance of SQL Server 2000 on the computer to SQL Server 2005 during or after setup. To restore these settings, locate "Microsoft SQL Server 2005 Backward

compatibility" in the **Add or Remove Programs** list in the Control Panel, click the **Change** button, and select the **Repair** option on the **Program Maintenance** page of the wizard.

3. Optionally, you can migrate DTS packages to the Integration Services format while leaving the original packages intact. For more information, see [Migrating Data Transformation Services Packages](#). The Package Migration Wizard is installed when Integration Services is installed.

▣ **See Also**

Tasks

[How to: Ensure Support for Data Transformation Services Packages](#)

Concepts

[SQL Server 2005 Integration Services Backward Compatibility](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Migrate Data Transformation Services Packages to Integration Services

 [See Also](#)



[Upgrading SQL Server Components](#) > [Upgrading SQL Server Components How-to Topics](#) > [Upgrading or Migrating Data Transformation Services How-to Topics](#) >

You are not required to migrate SQL Server 2000 Data Transformation Services (DTS) packages to the SQL Server 2005 Integration Services (SSIS) package format. DTS packages can continue to run alongside Integration Services packages.

For more information about the upgrade and migration options available to users of Data Transformation Services, see [Upgrading or Migrating Data Transformation Services](#).

If you want to migrate your DTS packages, SQL Server 2005 provides the Package Migration Wizard to help in this process. Because the wizard leaves your original packages intact, you can always continue to use the original packages if you experience migration difficulties.

The Package Migration Wizard is installed when Integration Services is installed. For more information, see [How to: Ensure Support for Data Transformation Services Packages](#).

 **Note:**

The Package Migration Wizard is available in the Standard, Enterprise, and Developer Editions of SQL Server 2005.

For more information about migrating DTS packages, see [Migrating Data Transformation Services Packages](#).

 **To migrate Data Transformation Services packages to**

Integration Services

1. Optionally, run the Upgrade Advisor to analyze your existing packages and obtain a report of potential migration issues. For more information, see [Known Package Migration Issues](#).
2. Run the Package Migration Wizard to migrate your packages.
3. Reconfigure the package items that must be manually reconfigured as reported by the Upgrade Advisor or the wizard.
4. Test and troubleshoot the migrated packages as needed.

▣ **See Also**

Tasks

[How to: Ensure Support for Data Transformation Services Packages](#)

Concepts

[SQL Server 2005 Integration Services Backward Compatibility](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Migrating Notification Services How-to Topics



[See Also](#)



[Upgrading to SQL Server 2005](#) > [Upgrading SQL Server Components](#) > [Upgrading SQL Server Components How-to Topics](#) >

The topics in this section provide instructions for upgrading and migrating Notification Services instances to Microsoft SQL Server 2005.

▣ In This Section

- [How to: Migrate Notification Services 2.0 Instances to SQL Server 2005 \(Command Prompt\)](#)
- [How to: Migrate Notification Services 2.0 Instances to SQL Server 2005 \(SQL Server Management Studio\)](#)
- [How to: Migrate Notification Services Instances to a Higher Edition \(Command Prompt\)](#)
- [How to: Migrate Notification Services Instances to a Higher Edition \(SQL Server Management Studio\)](#)
- [How to: View Notification Services Version Information](#)

▣ **See Also**

Tasks

[How to: Register the Core Notification Services Assembly for COM Interop](#)

Concepts

[Migrating Notification Services Instances](#)

[Installing SQL Server Notification Services](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Migrate Notification Services 2.0 Instances to SQL Server 2005 (SQL Server Management Studio)

 [See Also](#)



[Upgrading SQL Server Components](#) > [Upgrading SQL Server Components How-to Topics](#) > [Migrating Notification Services How-to Topics](#) >

Updated: **14 April 2006**

If you have an existing instance of Notification Services 2.0 running on Microsoft SQL Server 2000, you can upgrade the databases and Notification Services and then migrate the Notification Services instance to Microsoft SQL Server 2005.

The following procedures show how to upgrade an instance using SQL Server Management Studio.

Important:

Instances of Notification Services can be deployed in many different configurations. Use the following procedure as a guideline for developing and testing your own migration procedures.

To prepare servers for the upgrade

1. Disable the instance of Notification Services using the Notification Services 2.0 version of the **nscontrol** command-prompt utility:
 1. On the **Start** menu, point to **Programs\Microsoft SQL Server Notification Services**, and then click **Notification Services Command Prompt**.
 2. Type the following command to disable the instance:
nscontrol disable -name *instance_name*

2. On each server that runs the NS\$*instanceName* Windows service, stop the service:
 1. On the **Start** menu, point to **Programs\Microsoft SQL Server Notification Services**, and then click **Notification Services Command Prompt**.
 2. Type the following command to stop the instance:
net stop NS\$ *instanceName*
3. On each server that has an event provider, generator, distributor, or subscription management interface, unregister the instance.
 1. Type the following in the open Notification Services Command Prompt window:
nscontrol unregister -name *instance_name*
 2. Close the Command Prompt window.

▣ **To upgrade the Database Engine and install Notification Services**

1. Using SQL Server 2005 Setup, upgrade the Database Engine from SQL Server 2000 to SQL Server 2005. For more information, see [Upgrading the Database Engine](#).
2. On each server that runs an event provider, generator, distributor, or subscription management interface, install SQL Server 2005 Notification Services.

If Notification Services and the databases are located on the same server you can upgrade the database and install Notification Services at the same time.

▣ **To migrate the instance of Notification Services**

1. Open a SQL Server 2005 Notification Services Command Prompt window and add the instance information to the system databases:
 1. On the **Start** menu, point to **All Programs\Microsoft SQL Server 2005\Configuration Tools**, and then click **Notification Services Command Prompt**.
 2. Type the following command to repair instance metadata:
nscontrol repair -name *instance_name* -database *instanceDatabaseName* -schema *instanceDatabaseSchema* -server *databaseServer*
2. Use SQL Server Management Studio to upgrade instance and application database data:
 1. On the **Start** menu, point to **Programs\Microsoft SQL Server**, and then click **SQL Server Management Studio**.
 2. Connect to the instance of SQL Server that hosts the instance and application databases.
 3. In Object Explorer, expand **Notification Services**.
 4. Right-click the instance, point to **Tasks**, and then select **Upgrade**. This upgrades the database schema, stored procedures, and version numbers in the instance and application databases.
3. Modify notification generation rules in the application definition file (ADF) for each application hosted by the instance.

- Modify all notification generation rules that use the **Notify()** function to use the INSERT INTO syntax. For example, change the following Notification Services 2.0 rule:

 Copy

```
SELECT dbo.FlightNotificationsNotify(S.SubscriberId,
S.DeviceName, S.SubscriberLocale, E.Carrier, E.LeavingFrom,
E.GoingTo, E.Price, E.Conditions)
FROM FlightEvents E, FlightSubscriptions S
WHERE E.LeavingFrom = S.LeavingFrom
AND E.GoingTo = S.GoingTo
AND ( (E.Carrier = S.Carrier) OR (S.Carrier = '*' ) )
AND E.Price < S.Price
```

To the following syntax, which no longer uses the **Notify()** function, but instead selects data and inserts it into a view named

after the notification class (FlightNotifications):

 Copy Co

```
INSERT INTO FlightNotifications(SubscriberId, DeviceName,
SubscriberLocale, Carrier, LeavingFrom, GoingTo, Price,
Conditions)
SELECT S.SubscriberId, S.DeviceName, S.SubscriberLocale,
E.Carrier, E.LeavingFrom, E.GoingTo, E.Price, E.Condition
FROM FlightEvents E, FlightSubscriptions S
WHERE E.LeavingFrom = S.LeavingFrom
AND E.GoingTo = S.GoingTo
AND ( (E.Carrier = S.Carrier) OR (S.Carrier = '*' ) )
AND E.Price < S.Price
```

- Modify version numbers in the ADFs and ICF. (Optional)
- If you have moved the instance, update the **SystemName** values in the ADF and the **SqlServerSystem** value in the ICF. These values might be parameters, which indicates that the values are in a **ParameterDefaults** node or are provided when creating the instance.

4. In Object Explorer, right-click the Notification Services instance, point to **Tasks**, and then select **Update**.

Provide the requested information in the **Update** dialog box. Updating the instance and its applications applies the ADF and ICF changes to the databases.

Make sure to select **Enable instance after it is updated**.

5. On each server where the instance was previously registered, register the instance.

SQL Server Management Studio registers the instance on the local computer only. To use SQL Server Management Studio to register the instance, right-click the instance, point to **Tasks**, and then select **Register**.

If you have to register the instance on servers where SQL Server Management Studio is not installed, use the SQL Server 2005 **nscontrol register** command-line utility to register the instance.

6. Replace any custom components with new components compiled by

using the SQL Server 2005 assemblies and the Microsoft .NET Framework 2.0.

7. If using COM interop, reregister the core Notification Services assembly. For more information, see [How to: Register the Core Notification Services Assembly for COM Interop](#).
8. In Object Explorer, right-click the instance and select **Start**.

▣ **See Also**

Tasks

[Migrating Notification Services 2.0 Instances to SQL Server 2005](#)

[How to: Migrate Notification Services 2.0 Instances to SQL Server 2005 \(Command Prompt\)](#)

Other Resources

Managing Instances and Applications How To (Notification Services)
nscontrol register Command

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Migrate Notification Services 2.0 Instances to SQL Server 2005 (Command Prompt)

 [See Also](#)



[Upgrading SQL Server Components](#) > [Upgrading SQL Server Components How-to Topics](#) > [Migrating Notification Services How-to Topics](#) >

Updated: **14 April 2006**

If you have an existing instance of Notification Services 2.0 running on Microsoft SQL Server 2000, you can upgrade that instance to Microsoft SQL Server 2005. To perform the upgrade, you must upgrade the databases and upgrade the instance of Notification Services. The following procedure shows how to upgrade an instance using command-prompt utilities.

Important:

Instances of Notification Services can be deployed in many different configurations. Use the following procedure as a guideline for developing and testing your own migration procedures.

To prepare servers for the upgrade

1. On one server where the instance is registered, disable the instance of Notification Services by using the Notification Services 2.0 version of the **nscontrol** command-prompt utility:
 1. On the **Start** menu, point to **All Programs**, point to **Microsoft SQL Server 2005**, point to **Configuration Tools**, and then click **Notification Services Command Prompt**.
 2. Type the following command to disable the instance:
nscontrol disable -name *instance_name*

2. On each server that runs the NS\$*instanceName* Windows service, stop the service:
 1. Open a Notification Services 2.0 command prompt window and type the following command to stop the instance:
net stop NS\$ *instanceName*
3. On each server that has an event provider, generator, distributor, or subscription management interface, unregister the instance.
 1. Type the following in the open Notification Services Command Prompt window:
nscontrol unregister -name *instance_name*
 2. Close the Command Prompt window.

▣ **To upgrade the Database Engine and install Notification Services**

1. Using SQL Server 2005 Setup, upgrade the Database Engine from SQL Server 2000 to SQL Server 2005. For more information, see [Upgrading the Database Engine](#).
2. On each server that runs an event provider, generator, distributor, or subscription management interface, install SQL Server 2005 Notification Services.

If Notification Services and the databases are located on the same server you can upgrade the database and install Notification Services at the same time.

Installations of Notification Services 2.0 are not altered during the upgrade process.

▣ **To migrate the instance of Notification Services**

1. On each server where the instance was previously registered, register the instance:

1. On the **Start** menu, point to **All Programs**, point to **Microsoft SQL Server 2005**, point to **Configuration Tools**, and then click **Notification Services Command Prompt**.
2. Use the **nscontrol register** command to register the instance. The following command shows how to register the instance and create a Windows service that connects to the database server using Windows authentication:
nscontrol register -name *instanceName* -server *databaseServer* -service -serviceusername *username* -servicepassword *password*
For more information, type **nscontrol register -?** or see **nscontrol register Command**.

2. On one server, type the following command to repair instance metadata:

```
nscontrol repair -name instance_name -database  
instanceDatabaseName -schema instanceDatabaseSchema
```

3. Type the following command to upgrade instance and application data:

```
nscontrol upgrade -name instanceName
```

4. Modify notification generation rules in the application definition file (ADF) for each application hosted by the instance.

- Modify all notification generation rules that use the **Notify()** function to use the INSERT INTO syntax. For example, change the following Notification Services 2.0 rule:

 [Copy](#)

```
SELECT dbo.FlightNotificationsNotify(S.SubscriberId,  
S.DeviceName, S.SubscriberLocale, E.Carrier, E.LeavingFrom  
E.GoingTo, E.Price, E.Conditions)  
FROM FlightEvents E, FlightSubscriptions S  
WHERE E.LeavingFrom = S.LeavingFrom  
AND E.GoingTo = S.GoingTo  
AND ( (E.Carrier = S.Carrier) OR (S.Carrier = '*' ) )  
AND E.Price < S.Price
```

To the following syntax, which no longer uses the **Notify()** function, but instead selects data and inserts it into a view named

after the notification class (FlightNotifications):

 Copy Co

```
INSERT INTO FlightNotifications(SubscriberId, DeviceName,
SubscriberLocale, Carrier, LeavingFrom, GoingTo, Price,
Conditions)
SELECT S.SubscriberId, S.DeviceName, S.SubscriberLocale,
E.Carrier, E.LeavingFrom, E.GoingTo, E.Price, E.Condition
FROM FlightEvents E, FlightSubscriptions S
WHERE E.LeavingFrom = S.LeavingFrom
AND E.GoingTo = S.GoingTo
AND ( (E.Carrier = S.Carrier) OR (S.Carrier = '*' ) )
AND E.Price < S.Price
```

- Modify version numbers in the ADFs and ICF. (Optional)
- If you have moved the instance, update the **SystemName** values in the ADF and the **SqlServerSystem** value in the ICF. These values might be parameters, which indicates that the values are in a **ParameterDefaults** node or are provided when creating the instance.

5. Type the following command to update the instance of Notification Services:

```
nscontrol update -in ICFPath\ICFName.xml
```

Exact arguments vary depending on the authentication mode.

6. Type the following command to enable the instance of Notification Services:

```
nscontrol enable -name instanceName
```

Exact arguments vary depending on the authentication mode.

7. Replace any custom components with new components compiled by using the SQL Server 2005 assemblies and the Microsoft .NET Framework 2.0.
8. If using COM interop, reregister the core Notification Services assembly. For more information, see [How to: Register the Core Notification Services Assembly for COM Interop](#).

9. When you are ready to start the instance, type the following command on each server that runs the Notification Services engine components:

```
net start NS$ instanceName
```

▣ **See Also**

Tasks

[Migrating Notification Services 2.0 Instances to SQL Server 2005](#)

[How to: Migrate Notification Services 2.0 Instances to SQL Server 2005 \(SQL Server Management Studio\)](#)

Other Resources

nscontrol Utility

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Migrate Notification Services Instances to a Higher Edition (SQL Server Management Studio)

 [_See Also](#)



[Upgrading SQL Server Components](#) > [Upgrading SQL Server Components How-to Topics](#) > [Migrating Notification Services How-to Topics](#) >

If you create an instance of Notification Services using a lower edition of Microsoft SQL Server 2005, such as Standard Edition, and later want to upgrade to a higher edition of SQL Server, such as Enterprise Edition, you must upgrade the instance of Notification Services to the higher edition. The following procedure shows how to migrate a Notification Services instance to a higher edition using SQL Server Management Studio.

Important:

Instances of Notification Services can be deployed in many different configurations. Use the following procedure as a guideline for developing and testing your own migration procedures.

To upgrade to a higher edition of Notification Services

1. In SQL Server Management Studio Object Explorer, expand **Notification Services**.
2. Right-click the instance and select **Disable**.
3. Right-click the instance you are migrating and select **Stop**.
4. Close SQL Server Management Studio and any other open applications.
5. On each server where the instance of Notification Services is installed, use Setup to upgrade Notification Services to the higher edition.

6. In SQL Server Management Studio Object Explorer, expand **Notification Services**.
7. In Object Explorer, right-click the instance, point to **Tasks**, and then select **Upgrade**. This upgrades the edition and version numbers in the instance and application databases.
8. Right-click the instance and select **Update**.

Provide the requested information in the **Update** dialog.

Make sure to select **Enable instance after it is updated**.

9. Right-click the instance and select **Start**.

▣ **See Also**

Tasks

[Migrating Notification Services Instances to a Higher Edition of SQL Server 2005](#)

[How to: Migrate Notification Services Instances to a Higher Edition \(Command Prompt\)](#)

Other Resources

nscontrol Utility

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Migrate Notification Services Instances to a Higher Edition (Command Prompt)

 [See
Also](#)



[Upgrading SQL Server Components](#) > [Upgrading SQL Server Components How-to Topics](#) > [Migrating Notification Services How-to Topics](#) >

If you create an instance of Notification Services using a lower edition of Microsoft SQL Server 2005, such as Standard Edition, and later want to upgrade to a higher edition of SQL Server, such as Enterprise Edition, you must migrate the instance of Notification Services to match the database server. The following procedure shows how to migrate a Notification Services instance to a higher edition using command-prompt utilities.

Important:

Instances of Notification Services can be deployed in many different configurations. Use the following procedure as a guideline for developing and testing your own migration procedures.

To upgrade to a higher edition of Notification Services

1. On one server where the instance is registered, disable the instance:
 1. On the **Start** menu, point to **All Programs\Microsoft SQL Server 2005\Configuration Tools**, and then click **Notification Services Command Prompt**.
 2. Type the following command to disable the instance:
nscontrol disable -name *instanceName*
2. On each server where the instance is running, type the following command to stop the instance:

net stop NS\$ *instanceName*

If you are hosting the engine components in your own application, make sure the instance is not running.

3. On each server where the instance of Notification Services is installed, upgrade Notification Services to the higher edition.

4. On one server, type the following command to upgrade metadata in the databases:

nscontrol upgrade -name *instanceName* -server *databaseServer*

Exact arguments vary depending on the security configuration.

5. On one server, type the following command to update metadata in the databases:

nscontrol update -in *ICFPath\ICFName.xml*

Exact arguments vary depending on the security configuration.

6. On each server that runs an NS\$*instanceName* Windows service, type the following to start the service:

net start NS\$ *instanceName*

If you are hosting the engine components in your own application, start the instance.

7. On one server, type the following command to enable the instance:

nscontrol enable -name *instanceName*

▣ **See Also**

Tasks

[Migrating Notification Services Instances to a Higher Edition of SQL Server 2005](#)

[How to: Migrate Notification Services Instances to a Higher Edition \(SQL Server Management Studio\)](#)

Other Resources

nscontrol Utility

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: View Notification Services Version Information



[See Also](#)



[Upgrading SQL Server Components](#) > [Upgrading SQL Server Components How-to Topics](#) > [Migrating Notification Services How-to Topics](#) >

Notification Services stores the version and edition of Notification Services used to create, update, or upgrade the instance in the instance database, and stores the version used to register the instance in the Microsoft Windows Registry. When you migrate an instance of Notification Services, this information is updated.

If the version in the Windows Registry and the instance database do not match, the instance of Notification Services does not start. If you have trouble starting the instance after migration, verify the version and edition information using the procedure below.

▣ Procedures

To view product and registry version

-

In Object Explorer, right-click Notification Services and select **List Versions**.

The **List Versions** dialog only shows instances that are registered locally. The instance version number is the version in the instance's registry entry.

📌 Note:

On servers where Microsoft SQL Server Management Studio is not installed, run **nscontrol listversions** from the Notification Services Command Prompt.

To view the database version

1. In SQL Server Management Studio, click the **New Query** button and select **Database Engine Query**.
2. In the Database Engine Query window, run the following query, replacing *instanceDatabase* with the name of the instance database and replacing *instanceSchema* with the name of the instance schema.

 [Copy Code](#)

```
USE [instanceDatabase];  
SELECT Top(1) *  
FROM [instanceDatabase].[instanceSchema].[NSVersionInfo]  
ORDER BY [VersionId] DESC;
```

▣ **See Also**

Other Resources

nscontrol listversions Command

Viewing the Status of Instances and Applications

@@VERSION (Transact-SQL)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Backward Compatibility



[See Also](#)



[Upgrading to SQL Server 2005](#) >

The following sections contain backward compatibility information for Microsoft SQL Server components. This content includes information about deprecated features, discontinued features, breaking changes, and behavior changes.

▣ In This Section

Topic	Description
SQL Server 2005 Database Engine Backward Compatibility	Contains topics that describe Database Engine changes in Microsoft SQL Server 2005 that may require you to make changes to your applications.
SQL Server 2005 Analysis Services Backward Compatibility	Contains topics that describe SSAS changes in SQL Server 2005 that may require you to make changes to your applications.
SQL Server 2005 Integration Services Backward Compatibility	Describes SSIS changes in SQL Server 2005 that may require you to make changes to your existing Data Transformation Services applications.
Notification Services Backward Compatibility	Describes Notification Services changes in SQL Server 2005 that may require you to make changes to your exiting Notification Services 2.0 applications.
Reporting Services Backward Compatibility	Contains topics that describe SSRS changes in SQL Server 2005 that may require you to make changes to your existing Reporting Services solutions.
Replication Backward Compatibility	Contains topics that describe Database Engine changes in SQL Server 2005 that may require you to make changes to existing Replication solutions.

▣ **See Also**

Other Resources

[Upgrading SQL Server Components](#)

[Installing SQL Server 2005 Components](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

SQL Server 2005 Database Engine Backward Compatibility



[See Also](#)



[Upgrading to SQL Server 2005](#) > [Backward Compatibility](#) >

Topics in the backward compatibility section describe changes in behavior between versions of Microsoft SQL Server.

See...	For topics on...
Discontinued Database Engine Functionality in SQL Server 2005	Discontinued Database Engine functionality in this release.
Deprecated Database Engine Features in SQL Server 2005	Deprecated Database Engine features in this release.
Breaking Changes to Database Engine Features in SQL Server 2005	Database Engine feature changes that might require changes to applications.
Behavior Changes to Database Engine Features in SQL Server 2005	Other changes in behavior to database features in this release.

▣ **See Also**

Reference

[Version and Edition Upgrades](#)

[Features Supported by the Editions of SQL Server 2005](#)

Other Resources

sp_dbcmptlevel (Transact-SQL)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Deprecated Database Engine Features in SQL Server 2005



[Upgrading to SQL Server 2005](#) > [Backward Compatibility](#) > [SQL Server 2005 Database Engine Backward Compatibility](#) >

Updated: **12 December 2006**

Deprecated features include features that will not be supported in the next version of SQL Server and features that will not be supported in a future version of SQL Server.

▣ Features Not Supported in the Next Version of SQL Server

These Database Engine features will not be supported in the next version of SQL Server. We recommend that, as time allows, you replace these features with the replacement item if possible.

Category	Deprecated feature	Replacement
Backup and restore	DUMP statement	BACKUP
Backup and restore	LOAD statement	RESTORE
Backup and restore	BACKUP LOG WITH NO_LOG	None. The transaction log is automatically truncated when the database is using the simple recovery model. If you need to remove the log backup chain from a database, switch to the simple recovery model.
Backup and restore	BACKUP LOG WITH TRUNCATE_ONLY	None. The transaction log is automatically truncated when the database is using the simple recovery model. If you need to remove the log backup chain from a database, switch to the simple recovery model.
Backup and restore	BACKUP TRANSACTION	BACKUP LOG
Backup and restore	BACKUP { DATABASE LOG } WITH PASSWORD	None.

Backup and restore	BACKUP { DATABASE LOG } WITH MEDIAPASSWORD	None.
Backup and Restore	RESTORE { DATABASE LOG } ... WITH DBO_ONLY	RESTORE { DATABASE LOG } ... WITH RESTRICTED_USER
Backup and restore	RESTORE { DATABASE LOG } WITH PASSWORD	None.
Backup and restore	RESTORE { DATABASE LOG } WITH MEDIAPASSWORD	None.
Compatibility levels	60 and 65 compatibility levels	None.
Compatibility levels	70 compatibility level	None.
DBCC	DBCC CONCURRENCYVIOLATION	None.
Extended stored procedure programming	srv_getuserdata srv_setuserdata	Use CLR Integration instead.
Full-text search	sp_fulltext_service action values clean_up , connect_timeout , and data_timeout return zero.	None.
Instance options	SET REMOTE_PROC_TRANSACTIONS sp_configure 'remote proc trans'	Use linked servers and distributed queries. sp_addlinkedserver
Remote servers	Use of remote servers sp_addserver to create remote servers	Use linked servers. sp_addlinkedserver to create linked servers
Security	sp_addalias sp_dropalias sp_addgroup sp_changegroup	Superseded by roles

	sp_dropgroup	
	sp_helpgroup	
Security	SETUSER	EXECUTE AS
System tables	syssegments	None.

▣ Features Not Supported in a Future Version of SQL Server

These Database Engine features are supported in the next version of SQL Server, but will be removed in a later version. The specific version of SQL Server has not been determined.

Category	Deprecated feature	Replacement
Backup and restore	sp_helpdevice	sys.backup_devices
Collations	Hindi Lithuanian_Classic SQL_AltDiction_CP1253_CS_AS	None. These collations exist in Microsoft SQL Server 2008 but are not visible through fn_helpcollations .
Compatibility level	80 compatibility level	None. For more information about compatibility levels, see sp_dbcmptlevel (Transact-SQL) .
Data types	sp_addtype	CREATE TYPE
Data types	timestamp syntax for rowversion data type	rowversion data type syntax
Database management	sp_attach_db sp_attach_single_file_db	CREATE DATABASE statement with the FOR ATTACH option to rebuild multiple log files, or use the FOR ATTACH_REBUILD_LOG option if one or more have a new log file. To use the FOR ATTACH_REBUILD_LOG option.
Database objects	CREATE DEFAULT DROP DEFAULT	DEFAULT keyword in CREATE/ALTER TABLE

	sp_bindefault	
	sp_unbindefault	
Database objects	CREATE RULE DROP RULE sp_bindrule sp_unbindrule	CHECK keyword in CREATE/ALTER TABLE
Database objects	sp_renamedb	MODIFY NAME in ALTER DATABASE.
Database objects	Ability to return result sets from triggers	None.
Database options	'concat null yields null' of sp_dboption	None.
Database options	sp_dboption sp_resetstatus	ALTER DATABASE SET ONLINE EMERGENCY
Database options	TORN_PAGE_DETECTION option of ALTER DATABASE	PAGE_VERIFY TORN_PAGE_DETECTION option of ALTER DATABASE
DBCC	DBCC DBREINDEX	REBUILD option of ALTER INDEX.
DBCC	DBCC INDEXDEFRAG	REORGANIZE option of ALTER INDEX
DBCC	DBCC SHOWCONTIG	sys.dm_db_index_physical
Extended properties	Level0type = 'type' and Level0type = 'USER' to add extended properties to level-1 or level-2 type objects.	Use Level0type = 'USER' to add an extended property to a user or role. Use Level0type = 'SCHEMA' to add an extended property to level-1 types like TABLE or VIEW and level-2 types like COLUMN or TRIGGER. For more info see sp_addextendedprop
	srv_alloc	

Extended
stored
procedure
programming

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_paramtype
srv_pfield
srv_pfieldex
srv_rpcdb
srv_rpcname
srv_rpcnumber
srv_rpcoptions
srv_rpcowner

Use CLR Integration inste

	srv_rpcparams srv_senddone srv_sendmsg srv_sendrow srv_setcoldata srv_setcollen srv_setutype srv_willconvert srv_wsendmsg	
Extended stored procedure programming	sp_addextendedproc sp_dropextendedproc sp_helpextendedproc	Use CLR Integration inste
Extended stored procedures	xp_LoginConfig	IsIntegratedSecurityOnly argument of SERVERPROPERTY
Full-text search	sp_fulltext_catalog	CREATE/ALTER/DROP FULLTEXT CATALOG
Full-text search	sp_fulltext_table sp_fulltext_column sp_fulltext_database	CREATE/ALTER/DROP FULLTEXT INDEX ALTER FULLTEXT INDEX
Full-text search	sp_help_fulltext_tables[_cursor] sp_help_fulltext_columns[_cursor] sp_help_fulltext_catalogs[_cursor]	sys.fulltext_indexes sys.fulltext_index_columns sys.fulltext_catalogs
Functions	fn_get_sql	sys.dm_exec_sql_text
Index options	sp_indexoption	ALTER INDEX

	fillfactor = 0	fillfactor = 100
Index options	CREATE INDEX <index_option>::= syntax	CREATE INDEX <relational_index_option> syntax
Instance options	Default setting of disallow results from triggers option = 0	Default setting of disallow from triggers option = 1
Locking	sp_lock	sys.syslock_information
Locking	syslockinfo	sys.syslock_information
Metadata	DATABASEPROPERTY	DATABASEPROPERTY
	FILE_ID	FILE_IDEX
	INDEXKEY_PROPERTY	sys.index_columns
Other	DB-Library Embedded SQL for C	Although the SQL Server Database Engine still supports connections from existing applications using the DB-Library and Embedded SQL APIs, Microsoft does not include the files or documentation needed to continue programming work on applications that use these APIs. A future version of the SQL Server Database Engine will not support connections from DB-Library or Embedded SQL applications. Do not use DB-Library or Embedded SQL to develop new applications. Remove any dependencies on either DB-Library or Embedded SQL when modifying existing applications. Instead of the DB-Library APIs, use the SQLClient namespace or an API such as OLE DB or ODBC. SQL Server 2005 does not include the DB-Library DLL required to run

		applications. To run DB-L or Embedded SQL applica you must have available th Library DLL from SQL Se version 6.5, SQL Server 7 SQL Server 2000.
Query hints	FASTFIRSTROW hint	OPTION (FAST n).
Removable databases	sp_certify_removable sp_create_removable	sp_detach_db
Removable databases	sp_dbremove	DROP DATABASE
Security	sp_addapprole sp_dropapprole	CREATE APPLICATION DROP APPLICATION RO
Security	sp_addlogin sp_droplogin	CREATE LOGIN DROP LOGIN
Security	sp_adduser sp_dropuser	CREATE USER DROP USER
Security	sp_grantdbaccess sp_revokedbaccess	CREATE USER DROP USER
Security	sp_addrole sp_droprole	CREATE ROLE DROP ROLE
Security	sp_approlepassword sp_password	ALTER APPLICATION F ALTER LOGIN
Security	sp_changeobjectowner	ALTER SCHEMA or ALT AUTHORIZATION
Security	sp_defaultdb sp_defaultlanguage	ALTER LOGIN
Security	sp_denylogin sp_grantlogin	ALTER LOGIN DISABL CREATE LOGIN

	sp_revokelogin	DROP LOGIN
Security	USER_ID FILE_ID	DATABASE_PRINCIPAL FILE_IDEX
Security	sp_srvrolepermission sp_dbfixedrolepermission	These stored procedures return information that was correct in SQL Server 2000. The output does not reflect changes to the permissions hierarchy implemented in SQL Server 2005. For more information, see "Permissions of Fixed Roles" in SQL Server 2005 Books Online.
SET options	SET ANSI_NULLS SET ANSI_PADDING SET CONCAT_NULL_YIELDS_NULL	None.
SET options	SET OFFSETS	None.
SET options	SET ROWCOUNT for INSERT, UPDATE, and DELETE statements	TOP keyword.
	sysaltfiles syscacheobjects syscolumns syscomments sysconfigures sysconstraints syscurconfigs sysdatabases sysdepends	

System tables

sysdevices
sysfilegroups
sysfiles
sysforeignkeys
sysfulltextcatalogs
sysindexes
sysindexkeys
syslockinfo
syslogins
sysmembers
sysmessages
sysobjects
sysoledbusers
sysopentapes
sysperfinfo
syspermissions
sysprocesses
sysprotects
sysreferences
sysremotelogins
syssservers

Compatibility views. For more information, see **Compatibility Views (Transact-SQL)**.

 **Important:**

The compatibility views expose metadata for features introduced in SQL Server 7.0. It is recommended that you upgrade your application to use the new catalog views. For more information, see **Catalog Views (Transact-SQL)**.

	systypes	
	sysusers	
Table hints	Specifying NOLOCK or READUNCOMMITTED in the FROM clause of an UPDATE or DELETE statement when applied to the target table of the statement.	Remove the NOLOCK or READUNCOMMITTED hints from the FROM clause.
Textpointers	READTEXT, WRITETEXT, UPDATETEXT	None.
Textpointers	'text in row' table option	Use varchar(max) , nvarchar(max) , and varbinary(max) data types. For more information, see sp_tableoption (Transact-SQL).
Textpointers	TEXT, NTEXT and IMAGE data types	Use varchar(max) , nvarchar(max) , and varbinary(max) data types.
Textpointers	TEXTPTR(), TEXTVALID()	None.
Transact-SQL	:: function-calling sequence	Replaced by SELECT col FROM sys.fn_function_name.
Transact-SQL	3-part and 4-part column references in SELECT list	2-part names is the standard compliant behavior.
Transact-SQL	A string enclosed in quotes used as a column alias for an expression in a SELECT list: <i>'string_alias' = expression</i>	<i>expression [AS] column_alias</i> <i>expression [AS] [column_alias]</i> <i>expression [AS] "column_alias"</i> <i>expression [AS] 'column_alias'</i> <i>column_alias = expression</i>
Transact-SQL	Stored procedure numbers sys.numbered_procedures sys.numbered_procedure_parameters	None.
Transact-SQL	<i>table_name.index_name</i> syntax in DROP INDEX	<i>index_name ON table_name</i> syntax in DROP INDEX.

Transact-SQL	UPDATE table1, table2, ... SET syntax	Specifying more than one the UPDATE target is nonstandard and ambiguous
--------------	---------------------------------------	---

▣ Change History

Release	History
12 December 2006	Changed content: <ul style="list-style-type: none">• Removed syslanguages and syscharsets from the table of features not supported in a future version of SQL Server.
17 July 2006	New content: <ul style="list-style-type: none">• Added syssegments to table of features not supported in next version of SQL Server.

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Discontinued Database Engine Functionality in SQL Server 2005



[Upgrading to SQL Server 2005](#) > [Backward Compatibility](#) > [SQL Server 2005 Database Engine Backward Compatibility](#) >

Updated: 5 December 2005

These Database Engine features from earlier versions of SQL Server are not supported in SQL Server 2005.

Category	Discontinued feature	Replacement
Backup and restore	Named pipe backup devices.	Disk or tape devices. For security reasons, SQL Server does not support backup to named pipes.
Command prompt utilities	osql utility	Use the sqlcmd utility.  Note: sqlcmd and osql utilities do not support the passing of stored procedure return values or the exit value. For more information, see sqlcmd Utility .
Configuration options	' allow updates ' option of sp_configure .	Option is present but direct updates to system tables are not supported
Configuration options	' open objects ' option of sp_configure .	Option is present but its functionality has been deactivated. In SQL Server 2005, the number of open database objects is managed dynamically and is only limited by available memory. The 'open objects' option has been left in sp_configure to ensure backward compatibility with existing scripts.
Configuration options	'set working set size' option of sp_configure	Option is present but its functionality has been deactivated.

Database creation	DISK INIT DISK RESIZE	Legacy behavior from SQL Server 6.x
Database creation	FOR LOAD option of CREATE DATABASE	RESTORE operations can create a database
DBCC	DBCC DBREPAIR	Use DROP DATABASE to remove a damaged database. For more information, see DROP DATABASE (Transact-SQL) .
DBCC	DBCC NEWALLOC	DBCC CHECKALLOC For more information, see DBCC (Transact-SQL) .
DBCC	DBCC PINTABLE, DBCC UNPINTABLE	None.
DBCC	DBCC ROWLOCK	Row-level locking is automatic.
DBCC	DBCC TEXTALL DBCC TEXTALLOC	DBCC CHECKDB DBCC CHECKTABLE
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.	None.
Network protocols	The following protocols: NWLink IPX/SPX, AppleTalk, Banyan Vines, Multiprotocol.	Configure your application and the instance of the Database Engine to use one of the supported protocols: TCP/IP sockets, named pipes, VIA, or shared memory. For more information, see Choosing a Network Protocol .
Rebuild master	Rebuildm.exe	Use the REBUILDDATABASE option in Setup.exe. For more information, see How to: Install S

		Server 2005 from the Command Prompt.
Sample databases	Northwind and pubs	Use AdventureWorks ; however, Northwind and pubs are available as downloads, or can be copied from a previous installation of SQL Server. For more information, see Samples and Sample Databases .
Setup.exe	Remote Setup - the TARGETCOMPUTER parameter - is not supported.	Use a remote connection to run the SQL Server Setup program in user interface mode or from the command prompt.
APIs	SQL-DMO based WMI provider	Managed code: Microsoft.SqlServer.Management.Smo.V Non-managed code: WMI Provider for Configuration Management
APIs	SQL Namespace API (SQL-NS)	None
Transact-SQL	*= and =* outer join operators	Use the JOIN syntax of the FROM clause. For more information, see FROM (Transact-SQL) .
Virtual tables	syslocks	sys.dm_tran_locks For more information, see sys.dm_tran_lo

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Breaking Changes to Database Engine Features in SQL Server 2005



[See Also](#)



[Upgrading to SQL Server 2005](#) > [Backward Compatibility](#) > [SQL Server 2005 Database Engine Backward Compatibility](#) >

Updated: **14 April 2006**

This topic describes the changes made to the Database Engine in Microsoft SQL Server 2005 that could cause applications based on earlier versions of SQL Server to break.

□ Client/Server Connectivity

Feature	Description
Banyan VINES Sequenced Packet Protocol (SPP), Multiprotocol, AppleTalk, or NWLink IPX/SPX network protocols	<p>SQL Server 2005 does not support the Banyan VINES Sequenced Packet Protocol (SPP), Multiprotocol, AppleTalk, or NWLink IPX/SPX network protocols. To connect to SQL Server 2005, client applications must use a supported protocol. If an alias is set up that uses one of the unsupported protocols, the alias must be modified to use one of the supported protocols.</p> <p>If an application connection string specifically uses or loads one of the unsupported protocols, by either specifying NETWORK=DBMSRPCN for RPC, NETWORK=DBMSADSN for Appletalk, or NETWORK=DBMSVINN for Banyan VINES property, or by using an explicit prefix such as spx:server\instance for SPX, bv:server for Banyan VINES, adsp:server for AppleTalk, or rpc:server for multiprotocol, then you must modify your application to use one of the supported protocols.</p> <p>For more information, see Choosing a Network Protocol.</p>
MDAC	<p>Versions of MDAC earlier than MDAC 2.6 do not support named instances. To allow application connections to named instances, upgrade to the current version of MDAC.</p>
Winsock proxy	<p>The Winsock proxy cannot be configured by using SQL Server tools. For information about how to configure the Winsock proxy, see the proxy server documentation.</p>

▣ Configuration Options

Feature	Description
<p>AUTO_UPDATE_STATISTICS</p>	<p>Set AUTO_UPDATE_STATISTICS to ON before you upgrade any databases. Otherwise, database statistics are not updated as part of the upgrade to SQL Server 2005. Relying on statistics from an earlier version of SQL Server may cause suboptimal query plans. By setting AUTO_UPDATE_STATISTICS to ON, all statistics are updated when they are first referenced. Updating statistics increases the possibility that more optimal query plans are selected when you execute queries.</p> <p> Note: In some cases, after you set AUTO_UPDATE_STATISTICS to ON, the process of updating statistics may affect server performance of queries when statistics are first referenced.</p> <p>To set the AUTO_UPDATE_STATISTICS database SET option to ON, use the ALTER DATABASE statement; or, to update statistics in the database, run sp_updatestats.</p>
<p>max server memory option</p>	<p>In SQL Server 2000, the SQL Server buffer pool can exceed the limit specified by the max server memory option if the system physical memory is available. In SQL Server 2005, the buffer pool cannot exceed the value of max server memory. When this limit is reached, the query fails with an "insufficient system memory" error.</p> <p>If you receive this error and the max server</p>

	<p>memory option is set, increase the value of the option or reset the value to the default value of 2147483647. For more information, see Server Memory Options.</p>
<p>query governor cost limit option</p>	<p>Applying SET GOVERNOR_QUERY_COST_LIMIT or the query governor cost limit option of sp_configure may cause queries that run in an earlier version of SQL Server not to run in SQL Server 2005. This behavior occurs because of changes in query cost modeling.</p> <p>Update the query governor cost-limit settings of the connection or server instance to an appropriate value, or set the value to 0 to specify no limit on the time period in which a query can run.</p>

▣ Databases, Data and Log Files

Feature	Description
Compressed drives	<p>SQL Server 2005 cannot create or upgrade databases on compressed drives. When you install SQL Server 2005, select an uncompressed drive for system databases and verify that databases to be upgraded are not on compressed drives. However, note that after the database has been upgraded, you can put read-only databases and read-only secondary filegroups on an NTFS compressed file system.</p>
Data files	<p>Additional disk space is required for data files to handle the following changes:</p> <ul style="list-style-type: none"> • Additional system metadata is created and maintained in the PRIMARY filegroup of each user database for database objects and user permissions. For example, in earlier versions of SQL Server, the permissions associated with a grantor or grantee is stored in a single row as a bitmap. In SQL Server 2005, the bitmap is expanded into multiple rows. • Large object (LOB) columns defined as text, ntext, or image data types require an additional 40 bytes of disk space per column. This one-time space increase occurs during the first update of each LOB column. • The full-text document ID (DOCID) map is stored in the data file instead of in the full-text catalog. <p>To make sure that resources can handle size increases during upgrade and subsequent production operations, we recommend setting autogrow to ON for all user data files before you upgrade to SQL Server 2005. After you have upgraded and tested your workloads, you may want to set autogrow OFF or adjust the FILEGROWTH increment accordingly. For more information, see ALTER DATABASE (Transact-SQL).</p>
Database compatibility	<p>When a database is upgraded to SQL Server 2005 from any earlier version of SQL Server, the database retains its existing compatibility level. If you change the compatibility mode to 90 after you upgrade, the differences in compatibility mode may</p>

mode	affect your applications. For more information about these differences, see sp_dbcmptlevel (Transact-SQL) .
Database ID 32767	In SQL Server 2005, this database ID is reserved. Detach the database before you upgrade.
Filegroups	All databases in the instance of SQL Server must have filegroups set to READ_WRITE before you upgrade to SQL Server 2005. To set the filegroup to READ_WRITE, use ALTER DATABASE .
Log files	<p>In SQL Server 2005, additional disk space is required for transaction log files. During the undo phase of a crash recovery, SQL Server 2005 lets users access the database. This is possible because the transactions that were uncommitted when the crash occurred reacquire any locks they held before the crash. When transactions are being rolled back, their locks help protect them from interference by users. This additional locking information must be maintained in the transaction log.</p> <p>To make sure resources can handle size increases during upgrade and subsequent production operations, we recommend setting autogrow to ON for all user log files before upgrading to SQL Server 2005. After you have upgraded and tested your workloads, you may want to set autogrow OFF or adjust the FILEGROWTH increment accordingly. For more information, see ALTER DATABASE (Transact-SQL).</p>
model database	<p>In SQL Server 2005, the model database contains the following changes:</p> <ul style="list-style-type: none"> • Larger minimum size. • Compatibility level is set to 90. • PAGE_VERIFY database option is set to CHECKSUM.
tempdb database	Additional disk space is required for the tempdb data and log files in SQL Server 2005. To make sure resources can handle size increases during upgrade and subsequent production operations, we recommend setting autogrow to ON for all tempdb data and log files before upgrading to SQL Server 2005. After you have upgraded and tested your workloads, you may want to set autogrow OFF or adjust the FILEGROWTH increment accordingly.

For more information, see **Troubleshooting Insufficient Disk Space in tempdb**.

▣ Features

Feature	Description
Extended stored procedures	<p>Extended stored procedures that are previously registered without the DLL name may not work after you upgrade to SQL Server 2005. This is because the old BINN directory is not added to the new path during the upgrade. SQL Server may not be able to locate the extended stored procedures.</p> <p>Before you upgrade to SQL Server 2005, follow these steps for each extended stored procedure that is not registered by using a full path name:</p> <ol style="list-style-type: none"> 1. To remove the extended stored procedure, run sp_dropextendedproc. 2. To register the extended stored procedure with the full path name, run sp_addextendedproc.
Log shipping	<p>Log shipping in earlier versions of SQL Server is incompatible with SQL Server 2005 and cannot be upgraded directly. After you upgrade to SQL Server 2005, reconfigure log shipping by using SQL Server Management Studio. For more information, see Upgrading a SQL Server 2000 Configuration.</p>
osql utility	<p>The osql utility does not support the ED and !! commands. Remove the ED and !! commands from your scripts. To use the ED and !! commands, use the sqlcmd utility instead.</p>
SQL-DMO WMI Provider	<p>The SQL-DMO WMI Provider has been discontinued and is not available in SQL Server 2005.</p>
SQL Mail	<p>SQL Server supports SQL Mail upgrade from SQL Server 7.0 or SQL Server 2000, however SQL Server 2005 requires Microsoft Outlook 2002, or a later version of a mail client.</p> <p> Note: This feature will be removed in a future version of Microsoft SQL Server. If you are currently using this feature in new development work, and plan to modify applications, stop using this feature. To send mail from SQL Server 2005, use the sp_sendmail stored procedure.</p>
	<p>When a client that is connected by using SQL Server Authentication</p>

SQL Mail	Mail that contains an attachment, SQL Server cannot set an appropriate context and will return an error. To avoid this problem, use Windows
SQL Namespace API (SQL-NS)	The SQL Namespace API (SQL-NS) has been discontinued and is no
Trace flags	<p>In SQL Server 2000, a trace flag set in session A does not automatically affect an already existing session B. Instead, the trace flag takes effect only the next time any trace flag is set in session B. This behavior is nondeterministic in SQL Server 2000 and is deterministic in SQL Server 2005. In SQL Server 2005, trace flags set in session A are set immediately in other concurrent sessions.</p> <p>Also, in SQL Server 2005, trace flags can be specified as either local or global using an additional argument in the DBCC TRACEON statement. If the local argument is not specified, the default value is local in SQL Server 2005, which is different from SQL Server 2000 in which the default is global.</p> <p>For more information, see Trace Flags (Transact-SQL).</p>
Trace flags	Some SQL Server 2000 trace flags do not exist in SQL Server 2005. Some trace flags have different functionality in SQL Server 2005. You should disable these trace flags before you upgrade to SQL Server 2005. After you upgrade, verify that the functionality of the trace flag has not changed. Also verify that the trace flag is required before you reenabling any trace flags.
Triggers	<p>In SQL Server 2005, data definition language (DDL) statements, such as CREATE INDEX, cannot be performed on the inserted and deleted tables inside a trigger. In earlier versions of SQL Server, some DDL statements can be performed on the inserted and deleted tables. For more information, see Using the inserted and deleted Tables.</p>
	<p>In SQL Server 2005, duplicate table or view index names are not allowed. Before you upgrade, use the following steps to remove duplicate indexes:</p> <ol style="list-style-type: none"> 1. Locate the duplicate indexes by executing the following query: <pre> SELECT DISTINCT OBJECT_NAME(o.id), name FROM sysindexes as o WHERE EXISTS (SELECT name FROM sysindexes as i WHERE i.id = o.id AND i.name = o.name and i.indid < o.indid); </pre>

Duplicate index names	<p>2. Use sp_rename to change one of the index names. Because the the same, you cannot determine which index will be renamed. To differentiate the indexes.</p> <pre>EXEC sp_rename N'table_name.index_name', N'new_index_name'</pre> <p>3. Verify which index was renamed by executing the following query. The following query returns all indexes, including key column name table or view:</p> <pre>SELECT i.name AS IndexName, c.name AS ColumnName, ik.colid AS ColID FROM sysindexes i JOIN sysindexkeys ik ON i.id = ik.id and i.indid = ik.indid JOIN syscolumns c ON c.id = ik.id and ik.colid = c.colid WHERE i.id = OBJECT_ID('table_or_view_name')</pre> <p>4. If you have to, use sp_rename again to correct the index names.</p>
Object names	<p>In SQL Server 2005, you cannot use the 0xFFFF character in object name that contains this Unicode character cannot be accessed when the database compatibility level 90. Rename objects that contain this character.</p>

▣ Indexed Views

Feature	Description
	<p>The following function expressions are considered nondeterministic in SQL Server 2005 and, therefore, may interfere with indexed view creation:</p> <ul style="list-style-type: none"> • References to string literals that are implicitly converted to datetime and smalldatetime. • Implicit conversion of non-Unicode character data between collations. <p>Expressions that involve implicit conversion of character strings to datetime or smalldatetime are considered nondeterministic in SQL Server 2005, unless the compatibility level is set to 80 or earlier. This is because the results depend on the LANGUAGE and DATEFORMAT settings of the server session. For example, the results of the expression CONVERT(datetime, '30 listopad 1996', 113) depend on the LANGUAGE setting, because the string 'listopad' means different months in different languages. Similarly, in the expression DATEADD(mm, 3, '2000-12-01'), SQL Server interprets the string '2000-12-01' based on the DATEFORMAT setting.</p>
Function determinism	<p>Implicit conversion of non-Unicode character data between collations is also considered nondeterministic, unless the compatibility level is set to 80 or earlier.</p> <p>Creating indexes on views that contain these expressions is not allowed in database compatibility level 90. Although existing views that contain these expressions from an upgraded database are maintainable, the query optimizer will not consider them in query plans in either compatibility level 80 or 90. For information about setting the compatibility level, see sp_dbcmptlevel.</p>

	<p>In the definition of indexed views in SQL Server 2005, you should explicitly convert the literal to the preferred date type by using a deterministic date format style. For a list of which date format styles are deterministic, see CAST and CONVERT (Transact-SQL).</p> <p>If you use implicit string-to-date conversions in existing indexed views that have been upgraded to SQL Server 2005, you should be certain that the LANGUAGE and DATEFORMAT settings are consistent in your databases and applications to avoid possible indexed view corruption.</p>
IGNORE_DUP_KEY	<p>When you create a unique clustered index on a view in SQL Server 2005, the IGNORE_DUP_KEY option must be set to OFF. This is the default setting. Setting IGNORE_DUP_KEY to ON could lead to indexed view corruption.</p> <p>Drop the clustered index on the view and re-create it without specifying the IGNORE_DUP_KEY option.</p>
Query hints	<p>Query hints in the definitions of indexed views are ignored in compatibility level 80. This may cause some applications to behave differently between compatibility levels 80 and 90. For more information, see Designing Indexed Views, Creating Indexed Views, and Query Hint (Transact-SQL).</p>

▣ Security

Feature	Description
Login names	<p>The following fixed server role names are reserved in SQL Server 2005 and cannot be used as user-defined login names:</p> <ul style="list-style-type: none"> • sysadmin • serveradmin • setupadmin • securityadmin • processadmin • dbcreator • diskadmin • bulkadmin <p>Before you upgrade to SQL Server 2005, follow these steps:</p> <ol style="list-style-type: none"> 1. Record the security identifiers (SIDs) of the logins by executing the following statement: <pre>SELECT name, sid FROM master.dbo.syslogins WHERE name IN('sysadmin', 'serveradmin', 'setupadmin', 'securityadmin', 'processadmin', 'dbcreator', 'diskadmin', 'bulkadmin')</pre> 2. Drop the logins. 3. To create new logins, use the sp_addlogin system procedure. In the <code>sid</code> parameter for each corresponding login, specify the SID that is returned in step 1.
Login security identifier (SID)	<p>In SQL Server 2005, duplicate security identifiers (SID) are not allowed for one of the logins and associated users before you upgrade.</p>
Remote	<p>In earlier versions of SQL Server, logins that come from remote instances of SQL Server can be marked as trusted by using the sp_remoteoption system procedure. SQL Server 2005 does not support this method of labeling remote logins. After you upgrade to SQL Server 2005, remote logins will no longer be marked as trusted.</p>

login mapping	<p>marked as trusted.</p> <p>For setting up and managing remote logins, use linked servers and linked stored procedures. For more information, see Linking Servers.</p>
SQL Server 6.5 logins	<p>SQL Server 6.5 saves password hashes in a format that is no longer supported. The old password cannot be directly upgraded to SQL Server 2005.</p> <p>To enable this login, you must reset its password. You can reset the password using ALTER LOGIN:</p> <pre>ALTER LOGIN <login name> WITH PASSWORD = '<new password>' MUST_CHANGE</pre> <p>The new password will be validated against the password complexity policy of your system, unless policy checking is disabled. We recommend that you use complex passwords and not disable policy checking. The MUST_CHANGE option forces the user to select a new password. This is not required, but is recommended.</p> <p>You can identify dormant SQL Server 6.5 logins by using the following query:</p> <pre>SELECT * FROM sysxlogins WHERE (xstatus & 2048) = 2048; GO</pre>
	<p>The name sys is reserved in SQL Server 2005 and cannot be used as a user name. Rename the user before you upgrade to SQL Server 2005. If the user is renamed, the database will be in a suspect state after the upgrade process and will be unavailable until the database is brought online.</p> <p>Before-Upgrade Procedure</p> <p>Before you upgrade to SQL Server 2005, in each database that contains the user sys, do the following:</p> <ol style="list-style-type: none"> 1. Create a new user. 2. Use the following statements to display all permissions that are granted to user sys and granted to user sys. <pre>-- Return permissions granted by user sys. SELECT * FROM sysprotects WHERE grantor = USER_ID('sys')</pre>

sys user
name

```
-- Return permissions granted to user sys.  
SELECT * FROM sysprotects WHERE uid = USER_ID('sys')
```

3. To transfer ownership of all objects that are owned by **sys** to the new user use **sp_changeobjectowner**.
4. Drop user **sys**.
5. To restore the original permissions captured in step 2, use the **AS n** clause of the **GRANT** statement.
6. Modify scripts to reference the new user.

After-Upgrade Procedure

If the user **sys** was not renamed before you upgraded, do the following:

1. Execute the statement **ALTER DATABASE db_name SET ONLINE**. The database will be in **SINGLE_USER** mode.
2. Follow all steps in the "Before-Upgrade Procedure" section.
3. Execute the statement **ALTER DATABASE db_name SET MULTI_USER**.

▣ System Objects and Metadata

Feature	Description
INFORMATION_SCHEMA.COLUMNS	<p>In SQL Server 2005, the ORDINAL_POSITION property of the INFORMATION_SCHEMA.COLUMNS view returns the bit pattern that is returned by the ColumnID property of the sys.columns view.</p> <p>To obtain a bit pattern that is comparable to the ColumnID property of the sys.columns view, reference the ColumnID property of the sys.columns view function when you query the INFORMATION_SCHEMA.COLUMNS view, as shown in the following example:</p> <pre>SELECT TABLE_NAME, COLUMN_NAME, COLUMNPROPERTY(OBJECT_ID(TABLE_NAME), COLUMN_NAME, 'ColumnID') AS ColumnID FROM AdventureWorks.INFORMATION_SCHEMA.COLUMNS WHERE TABLE_NAME = 'Contact';</pre>
INFORMATION_SCHEMA.SCHEMATA	<p>In earlier versions of SQL Server, the INFORMATION_SCHEMA.SCHEMATA view returns the names of all schemas in an instance of SQL Server. In SQL Server 2005, this view returns the names of all schemas in a database. This behavior is consistent with the behavior of the SCHEMATA view in earlier versions of SQL Server. For more information, see SCHEMATA.</p>
INFORMATION_SCHEMA column names that match the value '%SCHEMA'	<p>In earlier versions of SQL Server, INFORMATION_SCHEMA column names that match the value '%SCHEMA' are not returned. In SQL Server 2005, these columns are returned. When a database is upgraded to SQL Server 2005, the behavior is the same as the user name and any application that uses schema separation will not fail. However, users that use schema separation features in their application may fail if the expected schema name is not returned.</p> <p>For more information, see User-Schema Separation.</p>
sp_helptrigger	<p>SQL Server 2005 adds trigger_schema to the list of columns returned by the sp_helptrigger system stored procedure. The trigger_schema column of sp_helptrigger in applications.</p>

	<p>applications to handle the additional objects in the sys.triggers catalog view.</p>
<p>syslockinfo and sp_lock</p>	<p>In SQL Server 2000, the rsc_objid column in syslocks and the objid and indid columns in sysindexes and sysindexinfo refer to the object ID and index ID. In SQL Server 2005, the objid and indid columns in syslocks refer to the object ID and index ID. In SQL Server 2005, when lock partitioning is enabled, syslockinfo returns one row for any specific lock resource. In SQL Server 2005, when lock partitioning is enabled, sp_lock returns one row for any specific lock resource running under one transaction. In SQL Server 2005, sp_lock returns $N + 1$ rows returned, where N is the number of locks held on the same resource. In SQL Server 2005, GRANTED and WAITING are the lock status for the same resource; in SQL Server 2000, GRANTED and WAITING are the lock status for the same resource. For more information, see syslocks and sys.syslockinfo (Transact-SQL).</p>
<p>System object name and system type name collation matching</p>	<p>In earlier versions of SQL Server, system object names and system type names are matched against the collation of the current database. In SQL Server 2005, system object names and system type names are matched against the collation of the current database. In SQL Server 2005, system object names and system type names must correspond to the collation of the current database. If the collation of these objects in your script or application does not match the collation of the current database, the script or application may fail. For example, if the current database has a collation of Latin1_General_CI_AS, a script or application may fail if the current database has a collation of Latin1_General_CI_AS_KS_WS.</p>
<p>System object modification</p>	<p>Direct system catalog updates are not supported. If you attempt to do this will generate the following error message:</p> <p>"Server: Msg 259, Level 16, State 1, Line 1 Ad hoc updates to system catalogs are not supported."</p> <p>Modify your SQL scripts to use sp_rename instead of running an UPDATE statement. For example, use ALTER DATABASE instead of running an UPDATE statement on the database catalog table.</p>
	<p>Statements such as DROP TABLE, sp_dropextendedproc cannot be used to remove these objects are deployed in the repository. Remove all statements that try to drop these objects.</p>

System object removal	applications. Modify your application to grant EXECUTE permission on system objects of the Surface Area Configuration to some of these objects. For example, the sa_configure procedure can be disabled or enabled using the Surface Area Configuration tools.
sysperfinfo	In SQL Server 2005, sysperfinfo replaced the sys.dm_os_performance_counters column. Modify applications to ensure that they can handle the bigint data type. In SQL Server 2005, sysperfinfo is replaced by the sys.dm_os_performance_counters view instead.
System tables queried by using ' dbo ' in the search criteria	In earlier versions of SQL Server, system tables reside in the master database. In SQL Server 2005, they are owned by sys and logically appear to reside in the sys database. Applications that query system tables and have search criteria that include dbo will fail.

▣ Transact-SQL

Feature	Description
@@VERSION	SQL Server 2005 returns more detailed information than SQL Server 2000, in the format <i>major.minor.build.incremental-build</i>
CREATE STATISTICS	Specifying WITH ROWS in CREATE STATISTICS statements is not supported in SQL Server 2005. Modify CREATE STATISTICS statements that include WITH ROWS by specifying SAMPLE <i>number</i> between WITH and ROWS, or by specifying other options that comply with the documented syntax.
DISK INIT	The DISK INIT statement, used in earlier versions of SQL Server to create database or transaction log devices, has been removed from SQL Server 2005. Replace all occurrences of this statement with equivalent CREATE DATABASE or ALTER DATABASE statements.
UNION inside an INSERT INTO...SELECT statement	<p>When a UNION operator is inside an INSERT statement, SQL Server 2005 independently casts the data type of each UNION operation according to the rules of data type conversion. Then, the data types of the final result of the UNION operation are cast to the corresponding columns of the table targeted by the INSERT operation. This change in behavior may cause data-type casting errors in your applications.</p> <p>The following example demonstrates a data-type casting error. In compatibility levels 80 and earlier, the integer constant 1 in the first SELECT statement is cast directly to the data type of the destination column ReturnedValue, which is varchar(255). In compatibility level 90, the data type of the UNION result set is determined before casting to the destination column. For the second column in the first SELECT statement, the data type is determined to be int. For the second column in the second SELECT statement, the data type is determined to be varchar(4). Because the int data type has a higher precedence than the varchar(4) data type, when the data types of the UNION result set are then determined, the value test will be cast to the int data</p>

	<p>type causing a data type conversion error.</p> <p style="text-align: right;"> Copy Code</p> <pre>CREATE TABLE #test(ReturnedName varchar(255) NOT NULL ReturnedValue varchar(255) NULL) INSERT INTO #test SELECT 'col1', 1 UNION ALL SELECT 'test', 'test' DROP TABLE #test</pre>
UPDATETEXT	<p>SQL Server 2005 does not support using text pointers in UPDATETEXT statements that read and write to the same binary large objects (BLOB) by using the same text pointer. Copy the BLOB to a temporary table or to a table variable, and then reassign the value to the original column.</p>
WITH keyword when you are using table hints	<p>In SQL Server 2005, with some exceptions, table hints are supported in the FROM clause of a query only when the hints are specified by using the WITH keyword.</p> <p>For more information, see FROM (Transact-SQL) and Table Hint (Transact-SQL).</p>
ORDER BY in a view definition	<p>In SQL Server 2005, the ORDER BY clause in a view definition is used only to determine the rows that are returned by the TOP clause. The ORDER BY clause does not guarantee ordered results when the view is queried, unless ORDER BY is also specified in the query itself.</p>
UPDATE with locking hints	<p>In SQL Server 2000, locking hints are not checked for conflicts in an UPDATE statement when both of the following conditions are true:</p> <ul style="list-style-type: none"> • The table in the FROM clause is aliased. • The same table is referenced as a target of the UPDATE statement without an alias. <p>SQL Server ignores the locking hints supplied in the FROM clause and does not issue an error if the hints conflict. In SQL Server 2005, an error is returned when locking hints conflict under these conditions.</p>

□ XML

Feature	Description
OPENXML	<p>Because of changes to MSXML, OPENXML no longer supports noninteger positional predicates. In SQL Server 2005, MSXML 3.0 is the underlying engine that is used to process XPath expressions that are used within OPENXML queries. MSXML 3.0 has a more conformant XPath 1.0 engine in which the semantics of noninteger values in positional predicates has changed.</p> <p>For example, the following as an XPath expression <code>a[5.1]</code> will now return no element instead of the fifth <code><a></code> element. To correct this, use the rounded value directly. For example, revise the previous example to <code>a[5]</code>.</p>
OPENXML XPath expressions	<p>MSXML 3.0 has a stricter XPath 1.0 engine in which support for the following functions has been removed:</p> <ul style="list-style-type: none"> • format-number() • formatNumber() • current() • element-available() • function-available() • system-property() <p>For format-number() and formatNumber(), you can use Transact-SQL. For the other unsupported functions, there is no direct workaround.</p>
User-defined type 'xml'	<p>In SQL Server 2005, xml is a reserved system type. Use sp_rename to rename the type either before or after you upgrade and modify the application to work with the new type name.</p>

▣ Change History

Release	History
14 April 2006	<p data-bbox="350 426 561 457">New content:</p> <ul data-bbox="461 506 1414 674" style="list-style-type: none"><li data-bbox="461 506 1414 583">• Added an entry under the Transact-SQL section about using ORDER BY in a view definition.<li data-bbox="461 590 1414 674">• Added an entry under the Transact-SQL section about using locking hints with UPDATE.

▣ **See Also**

Reference

[Behavior Changes to Database Engine Features in SQL Server 2005](#)

[Deprecated Database Engine Features in SQL Server 2005](#)

[Discontinued Database Engine Functionality in SQL Server 2005](#)

Other Resources

[SQL Server 2005 Database Engine Backward Compatibility](#)

sp_dbcmptlevel (Transact-SQL)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Behavior Changes to Database Engine Features in SQL Server 2005



[See Also](#)



[Upgrading to SQL Server 2005](#) > [Backward Compatibility](#) > [SQL Server 2005 Database Engine Backward Compatibility](#) >

Updated: **17 July 2006**

This topic describes the changes in behavior of some Database Engine features in Microsoft SQL Server 2005 from their behavior in earlier versions of SQL Server.

▣ **Backup and Recovery**

When you restore an existing database, SQL Server 2005 requires that you back up the tail of the log before you restore the database under the full or bulk-logged recovery model. Trying to restore a database before you back up the tail of the log causes an error, unless the RESTORE statement contains either a WITH REPLACE or WITH STOPAT clause. For more information, see **Tail-Log Backups**.

▣ Cursors

The following table lists the implicit cursor conversions that occur in SQL Server 2000 but do not occur in SQL Server 2005. The workaround is to request the specific type of cursor, instead of depending on implicit conversion.

Condition	In SQL Server 2000, cursor is converted from	To
Query contains inline table-valued function.	Dynamic/keyset	Static
Query references a remote object.	Fast-forward	Keyset
Query contains no tables.	Fast-forward	Static
Query contains table function.	Fast-forward	Static
Query contains derived table.	Fast-forward	Static
Query contains multi-statement table function.	Dynamic	Keyset to Static
	Keyset	Static
Query contains a virtual table.	Dynamic/Keyset	Static
Query contains a view with TOP.	Fast-forward	Static
Query contains updatable partitioned views.	Fast-forward	Static
API cursor query contains an indexed view with a unique access path, but not at the base-table level.	Keyset	Static
API cursor query contains a table with no clustered index or unique key.	Keyset	Static
API cursor result set contains a text , ntext , or image column.	Fast-forward	Dynamic

SQL Server 2005 does not support generating keyset-driven or static Transact-SQL cursors asynchronously. Transact-SQL cursor operations, such as OPEN or FETCH, are ordinarily batched; therefore, there is no need for the asynchronous generation of Transact-SQL cursors. SQL Server 2005 continues to support asynchronous keyset-driven or static API server cursors where low-latency OPEN is a concern because of client roundtrips for each cursor operation.

In SQL Server 2005, when a dynamic cursor is declared over a table with no unique indexes, and a row is deleted outside the cursor, a subsequent refresh of the cursor retrieves a placeholder for the row with NULL data. In earlier versions of SQL Server, the cursor does not return the affected row.

In SQL Server 2005, if different cursor declarations are possible through conditional logic within a batch or stored procedure, INSERT, UPDATE, and DELETE statements executed through the cursor cause a recompile. In SQL Server 2000, these statements do not cause a recompile.

In the following example, the UPDATE statement will cause a recompile of the module.

 [Copy Code](#)

```
IF(@some_condition=1)
    DECLARE c CURSOR FOR
        SELECT region FROM db.dbo.mytable ORDER BY lastname
ELSE
    DECLARE c CURSOR FOR
        SELECT postalcode FROM db.dbo.mytable ORDER BY firstname
...
FETCH NEXT FROM c
...
UPDATE db.dbo.mytable
    SET  firstname='a' WHERE CURRENT OF c
```

To avoid recompilation, rewrite the code in the following way:

 [Copy Code](#)

```
IF (@some_condition=1)
BEGIN
    DECLARE c CURSOR FOR
        SELECT region FROM db.dbo.mytable ORDER BY lastname
    FETCH NEXT FROM c
    UPDATE db.dbo.mytable
        SET  firstname='a' WHERE CURRENT OF c
END
ELSE
BEGIN
    DECLARE c CURSOR FOR
        SELECT postalcode FROM db.dbo.mytable ORDER BY firstname
    FETCH NEXT FROM c
    UPDATE db.dbo.mytable
```

```
    SET  firstname='a' WHERE CURRENT OF c
END
```

In SQL Server 2005, cursor scroll locks acquire a lock upgrade from S (Shared) to U (Update), unless a higher lock hint is specified in the query. SQL Server 2000 transparently adds a U lock hint for scroll lock cursors. The SQL Server 2005 behavior allows for better concurrency but opens up the chance of deadlocks for concurrent cursors. Use the UPDLOCK hint to achieve the behavior you want. For more information, see **Table Hint (Transact-SQL)**.

In SQL Server 2000, when a column or row value that appears more than one time in a cursor fetch buffer is updated, the other occurrences might not reflect the update because of the size of the fetch buffer or the position of the rows within the buffer. In SQL Server 2005, the rows in the fetch buffer are updated so that the values are always consistent, regardless of the fetch buffer size or the position of the rows.

In SQL Server 2000, cursors that involve a UNION ALL statement incorrectly return updatability information based on the first set in the UNION. Therefore, some cursors are reported as updatable when they are not. SQL Server 2005 reports UNION ALL results as computed and, therefore, not updatable. If all cursor rows come from a single table or view, you can define a cursor that is updatable by changing the query to use IN or OR clauses, instead of UNION.

In SQL Server 2000, when the CURSOR_CLOSE_ON_COMMIT option is set to ON, all cursors in a connection are closed when the transaction is committed. In SQL Server 2005, only cursors that are opened within the current transaction are closed at transaction commit time. Cursors that are opened before the start of the transaction remain open.

SQL Server 2000 allows some cursor declarations that are not valid and converts them to other types of cursors. SQL Server 2005 does not allow declarations that are not valid. For example, a cursor that requests both updatability and insensitivity is not valid in SQL Server 2005 and fails with an error.

▣ Databases, Data and Log Files

SQL Server 2000 behavior	SQL Server 2005 behavior		
<p>The AUTO_CLOSE database option is a synchronous process that can decrease performance when the database is accessed by an application that repeatedly makes and breaks connections to the Database Engine.</p>	<p>The AUTO_CLOSE process is asynchronous. Repeatedly opening and closing the database no longer impairs performance.</p>		
<p>Files on raw partitions do not autogrow; therefore, the MAXSIZE and FILEGROWTH parameters are not required when <i>os_file_name</i> specifies a raw partition.</p>	<p>Files on raw partitions can autogrow. The MAXSIZE and FILEGROWTH parameters can be specified.</p>		
<p>Permissions are not set on the data and log files of each database.</p>	<p>Permissions are set on the data and log files whenever the following operations are applied to the database:</p>		
	<table border="1"> <tr> <td data-bbox="855 1148 1003 1186">Created</td> <td data-bbox="1003 1148 1425 1186">Modified to add a new file</td> </tr> </table>	Created	Modified to add a new file
	Created	Modified to add a new file	
	<table border="1"> <tr> <td data-bbox="855 1194 1003 1232">Attached</td> <td data-bbox="1003 1194 1425 1232">Backed up</td> </tr> </table>	Attached	Backed up
Attached	Backed up		
<table border="1"> <tr> <td data-bbox="855 1241 1003 1283">Detached</td> <td data-bbox="1003 1241 1425 1283">Restored</td> </tr> </table>	Detached	Restored	
Detached	Restored		
<p>The permissions prevent the files from being accidentally tampered with if they reside in a directory that has open permissions. For more information, see Securing Data and Log Files.</p> <p> Note: Microsoft SQL Server 2005 Express Edition does not set data and log file permissions.</p>			

▣ Linked Servers and Distributed Queries

A linked server that was defined by using 'SQLOLEDB' as the provider name will be modified to 'SQLNCLI' (**SQL Native Client OLE DB Provider**) upon upgrade. The **sys.sysservers** compatibility view shows linked servers that use 'SQLNCLI' as 'SQLOLEDB'. The SQL Server 2005 catalog view **sys.servers** shows linked servers that use 'SQLNCLI' as 'SQLNCLI'.

Heterogeneous queries and the use of OLE DB providers are not supported when SQL Server is running in fiber mode. Fiber mode is enabled when the **lightweight pooling** advanced configuration option is set to 1.

The SQL Native Client OLE DB Provider cannot be instantiated out-of-process.

Warning messages from a linked server in SQL Server 2005 are not propagated to the client. The most significant classes of such warnings are the following:

- ANSI-compliant warnings that NULL values have been eliminated in an aggregate calculation
- Arithmetic overflow warnings

For example, the following Transact-SQL statement might generate a warning message if there are NULL values in col1:

 [Copy Code](#)

```
SELECT SUM(col1)
FROM <Table>
GROUP BY col2
```

In SQL Server 2000 and earlier, linked servers propagate the warning message to the client. In SQL Server 2005, linked servers do not.

In SQL Server 2000, when a remote stored procedure does not run successfully because of compile-time errors such as an incorrect parameter binding, the return value/status is set to 0. When this scenario occurs in SQL Server 2005, the return value/status is set to NULL.

▣ Query Processor Architecture

Using parameter values during recompilation is different between SQL Server 2000 and SQL Server 2005 for batches submitted in the following forms:

- Stored procedures
- Using **sp_executesql**
- Prepared statements

When SQL Server 2000 recompiles these batches, it uses the parameter values with which the batches are called as part of recompilation. When SQL Server 2005 recompiles these queries, it uses the parameter values as they exist just before the statement that causes recompilation. These values may differ from those that were originally passed into the batch. For more information, see **Parameters and Execution Plan Reuse**.

▣ Security

Feature	SQL Server 2000 behavior	SQL Server 2005 behavior
GRANT ALL	Grants all applicable permissions.	<p>Granting ALL permission on objects and statements has been deprecated. When GRANT ALL is executed, the following occurs:</p> <ul style="list-style-type: none"> • The command succeeds, but only permissions grantable in SQL Server 2000 are granted to the user. • You receive the following message: "The ALL permission is deprecated and maintained only for compatibility. It DOES NOT imply ALL permissions defined on the entity." <p>SQL Server 2005 provides additional permissions at various scopes that can be used to manage user permissions. For example, the CONTROL permission can be used to grant ownership-like permissions on an object.</p>
	SQL Server 2000 maintains two versions of each SQL Server login password. One is the actual password supplied by the user, and the other is the password converted	Only the actual password is stored. A password entered by a user must match the password stored in the server.

<p>Password comparisons</p>	<p>by SQL Server to all uppercase letters. This enables case-insensitive validation of passwords. Although this behavior is convenient for many users, it makes password-guessing attacks easier by reducing the number of possible passwords.</p>	<p>If a password does not match the password stored in SQL Server, the login fails. If the precise case of the password characters is forgotten, the password must be reset.</p>
<p>Changing the default language for the sa account</p>	<p>The default language for the SQL Server sa account is the language that is selected during setup or upgrade.</p> <p>In earlier releases of Microsoft SQL Server, running sp_configure to change the default language of the server also updates the default language for the sa account.</p>	<p>To change the default language for the sa account in SQL Server 2005, you must run the sp_defaultlanguage stored procedure, execute the DBCC FREESYSTEMCACHE command, and then start a new session. Running sp_configure to change the default language for the server does not update the default language for the sa account.</p>

▣ System Stored Procedures

The following tables list the changes to parameters in the Database Engine system stored procedures.

Stored procedure	Parameter	Change description
sp_bindefault	@objname	Size changed from nvarchar(517) to nvarchar(776) .
sp_bindrule	@objname	Size changed from nvarchar(517) to nvarchar(776) .
sp_changeobjectowner	@objname	Size changed from nvarchar(517) to nvarchar(776) .
sp_detach_db	@keepfulltextindexfile	Size changed from nvarchar(517) to nvarchar(776) .
sp_fulltext_service	@action	Size changed from varchar(20) to nvarchar(100) .
sp_fulltext_service	@value	Data type changed from int to sql_variant .
sp_getapplock	@DbPrincipal	Parameter added.
sp_releaseapplock	@DbPrincipal	Parameter added.
sp_setapprole	@fCreateCookie	Parameter added.
sp_setapprole	@cookie	Parameter added.
sp_settriggerorder	@stmttype	Size changed from varchar(10) to varchar(50) .
sp_settriggerorder	@namespace	Parameter added.
sp_sproc_columns	@fUsePattern	Parameter added.
sp_stored_procedures	@fUsePattern	Parameter added.
sp_table_privileges	@fUsePattern	Parameter added.
sp_table_privileges_ex	@fUsePattern	Parameter added.
sp_tables	@fUsePattern	Parameter added.
sp_tables_ex	@fUsePattern	Parameter added.

▣ System Tables and Views

The following table lists the changes to the columns in system tables and views.

System table or view	Column	Change description
COLUMNS	ORDINAL_POSITION	Data type changed from smallint to int .
PARAMETERS	ORDINAL_POSITION	Data type changed from smallint to int .
REFERENTIAL_CONSTRAINTS	MATCH_OPTION	Size changed from varchar(4) to varchar(7) . Default value changed from 'NONE' to 'SIMPLE'.
REFERENTIAL_CONSTRAINTS	UPDATE_RULE	Size changed from varchar(9) to varchar(11) .
REFERENTIAL_CONSTRAINTS	DELETE_RULE	Size changed from varchar(9) to varchar(11) .
ROUTINE_COLUMNS	ORDINAL_POSITION	Data type changed from smallint to int .
sysaltfiles	name	Data type changed from nchar(128) to sysname .
sysaltfiles	filename	Data type changed from nchar(260) to

		nvarchar(260).
sysconfigures	config	Data type changed from smallint to int .
syscursorcolumns	data_type_sql	Data type changed from smallint to int .
sysfiles	name	Data type changed from nchar(128) to sysname .
sysfiles	filename	Data type changed from nchar(260) to nvarchar(260) .
sysmessages	severity	Data type changed from smallint to tinyint .
sysperfinfo	cntr_value	Data type changed from int to bigint .
sysprocesses	waittime	Data type changed from int to bigint .
sysprocesses	hostprocess	Size changed from nchar(8) to nchar(10) .
sysprocesses	request_id	Column added.
sysprotects	columns	Size changed from varbinary(4000) to varbinary(8000) .
sysservers	srvcollation	Data type changed from int to sysname .
syservers	nonsqlsub	Column added.

sysoledbusers	rmtpassword	Returns only NULL.
sysindexes	keys	Returns only NULL.
sysindexes	statblob	Returns only NULL.
syscomments	compressed	Returns only 0.
sysdevices	size	Returns only 0.
sysobjects	schema_ver	Returns only 0.
sysremotelogins	status	Returns only 0.
syssservers	topologyx	Returns only 0.
syssservers	topologyy	Returns only 0.

Transact-SQL

Feature	SQL Server 2000 behavior	SQL Server
<p>bcp utility</p>	<p>A user that has INSERT and SELECT permissions on a table can use the bcp utility to bulk load data into that table by using the following command:</p> <p style="text-align: right;"> Copy Code</p> <pre>bcp <target table> in <datafile> -c -T</pre> <p>By default, this command disables CHECK constraints and triggers on target tables.</p>	<p>To run the bcp utility ALTER permission and SELECT permission CHECK constraints during the bulk copy upgrade to SQL Server in applications may fail insufficient permissions handled in one of the</p> <ul style="list-style-type: none"> • Grant ALTER TABLE permission to user on all tables during bcp process. • Modify the bcp command to not enforce CHECK constraints and triggers, as shown in the following command: <pre>bcp <target table> in <datafile> "CHECK_CONSTRAINTS FIRE_TRIGGERS"</pre>
	<p>Each reference to built-in</p>	<p>An outer query can reference columns of views. However, if those columns are calling functions such as RAND, these multiple references to the function to be evaluated each actual call inside the table.</p> <p>Multiple references to a subquery do not cause</p>

<p>Built-in system functions</p>	<p>functions, such as NEWID and RAND, produces a different result because it is evaluated one time for each outer query reference.</p>	<p>evaluated multiple times, the value produced by the subquery.</p> <p>For example, in SQL Server 2005, the following query returns the same result. In SQL Server 2005,</p> <pre> SELECT Column1, Column2 FROM (SELECT Column1, Column2 FROM Table1 UNION SELECT Column1, Column2 FROM Table2) t </pre>
<p>BULK INSERT</p>	<p>BULK INSERT supports string-to-decimal type conversion for strings representing numeric values that use scientific notation.</p>	<p>The string-to-decimal conversion in BULK INSERT for the Transact-SQL CONVERT function rejects string values that use scientific notation. BULK INSERT treats these values as not valid and returns errors. For more information, see BULK INSERT (Transact-SQL).</p>
<p>datetime conversions</p>	<p>String-to-datetime conversions are marked as deterministic. However, this is not true for the styles listed in the following table. For these styles, the conversions depend on the language settings.</p> <p>The following table lists the styles for which the string-to-datetime conversion is nondeterministic.</p>	<p>SQL Server 2005 marks these conversions as nondeterministic.</p>

	All styles below 100 ¹	106	
		107	109
		113	130
	1 With the exception of styles 20 and 21		
DBCC CHECKFILEGROUP	If a nonclustered index in the specified filegroup is associated with a table in another filegroup, the index and the base table in the other filegroup are checked.		If a nonclustered index in the specified filegroup is associated with a table in another filegroup, the index and the base table in the other filegroup are checked.
DBCC SHOW_STATISTICS	The rowset returned by DBCC SHOW_STATISTICS does not contain a Name column.		The first rowset returned by DBCC SHOW_STATISTICS contains a column that is titled Name . This column appears as the first column in the rowset. If you have application programs that use the columns returned by DBCC SHOW_STATISTICS, you may need to modify them to access the Name column.
DROP LOGIN	When you execute DROP LOGIN, the login is not dropped if database users are mapped to the login.		When you execute DROP LOGIN, the login is dropped even if database users are mapped to the login.
Expressions in computed columns, CHECK constraints and DEFAULT constraints	The original text of an expression, including white spaces, is preserved in the catalog metadata. For example, a computed column expression entered as c1 + c2 + 1 will appear exactly as entered in the text column in the syscomments system table.		The original text of an expression, including white spaces, is preserved in the catalog metadata. For example, a computed column expression entered as c1 + c2 + 1 will appear exactly as entered in the text column in the syscomments system table. The semantics of the expression are preserved. For example, the expression (([c1]+[c2])+(1)) is stored in the sys.computed_columns catalog view.
			SQL Server 2005 sorts expressions in queries in the order they are evaluated in SQL.

<p>Expressions in queries</p>	<p>Unsafe expressions in queries do not always generate a runtime exception.</p>	<p>behavior provides the benefits:</p> <ul style="list-style-type: none"> • The ability to m computed column query that are th column expressi • The prevention computation of <p>However, depending query and the data in exceptions may occur the query contains an expression. These run include the following</p> <ul style="list-style-type: none"> • Arithmetic exce overflow, and u • Conversion fail precision and ar nonnumeric stri • Aggregation ov not all guarante <p>These exceptions ma Server 2000 in a spec specific data. Howev changed because of c potentially lead to an Server 2000. These r be prevented by mod include conditional e NULLIF or CASE. For see Troubleshooting on Query Expressio</p>
		<p>Permission change: f requires the user to h STATE permission o</p>

fn_servershareddrives	Data returned by the system function fn_servershareddrives can be viewed by members of the public role.	 Important: This SQL Server 2008 is included for backward compatibility. We recommend that you use sys.dm_io_cluster_
fn_virtualfilestats	Data returned by the system function fn_virtualfilestats can be viewed by members of the public role.	Permission change: fn_virtualfilestats requires the user to have the STATE permission on the server.
fn_virtualservernodes	Data returned by the system function fn_virtualservernodes can be viewed by members of the public role.	Permission change: fn_virtualservernodes requires the user to have the STATE permission on the server.  Important: This SQL Server 2008 is included for backward compatibility. We recommend that you use sys.dm_os_cluster_
HOST_ID	HOST_ID returns a char(8) value.	HOST_ID returns a
		The following indexes will be dropped during the upgrade process and need to be rebuilt because of changes in SQL Server 2005: <ul style="list-style-type: none"> • Indexes on columns with CHECKSUM(s) will be disabled because CHECKSUM has changed for CHECKSUM functions with timestamp columns. • Indexes that contain 0x3390, 0x33ca or nchar columns with a non-Unicode based collation.

Indexes

The indexes described in the "SQL Server 2005 behavior" column are allowed.

- because the sort collations has cl
- Indexes on com in which the vie expression cont conversion from **smalldatetime**, nondeterministi to **datetime** or s disabled.

For indexes that requ of the changes descri items described earli procedure.

1. Examine the SC warning messag
2. Run DBCC CH underlying table problem.
3. If the results of indicate a probl index by using o methods:
 - ALTER IN REBUILD
 - CREATE I DROP_EX
 - DBCC DB
4. Identify disable constraints by u statement:

```
SELECT * FROM  
WHERE is_disa
```

5. Enable any FOE by using the AL CONSTRAINT

		<p>PRIMARY KEY constraints are e associated index rebuilt before yo KEY constraints PRIMARY KEY</p> <p>For indexes disabled described in the third use the following pro</p> <ol style="list-style-type: none"> 1. Identify the inde columns that ha using the follow <pre> SELECT object_ object_name, FROM sys.inde JOIN sys.inde ON i.index_id AND i.objec JOIN sys.comp cc ON ic.object_ AND ic.colu WHERE i.is_di </pre> <ol style="list-style-type: none"> 2. For a computed index, and then column definitio CONVERT fun deterministic da 3. For an indexed then redefine it CONVERT fun deterministic da 4. Re-create the in computed colun
		<p>Parallel index operat rebuild indexes are a Server 2005 Develop</p>

<p>Indexes</p>	<p>Parallel index operations are supported in the SQL Server 2000 Developer, SQL Server 2000 Standard, and SQL Server 2000 Enterprise editions.</p>	<p>Enterprise editions.</p> <p>Users who upgrading Standard Edition to SQL Server 2005 Standard Edition should be aware that create, drop, or rebuild operations are not executed serially in SQL Server 2005 Standard Edition and finish.</p> <p>SELECT, INSERT, UPDATE, and DELETE statements are not executed in parallel in SQL Server 2005 Standard Edition.</p> <p>After you upgrade to SQL Server 2005 Standard Edition, you may have to adjust maintenance activities and maintenance time for these operations.</p> <p>To execute parallel in SQL Server 2005 Enterprise Edition.</p>
<p>ORDER BY clause</p>	<p>Column names in the ORDER BY clause are resolved to columns listed in the select list, regardless if they are qualified.</p> <p>For example, the following query executes without error:</p> <pre> USE pubs SELECT au_fname AS 'FName', au_lname AS 'LName' FROM authors a ORDER BY a.LName </pre> <p>Copy Code</p>	<p>Qualified column names are resolved to columns in the FROM clause. If <i>order by</i> is not qualified, it must be qualified with columns listed in the FROM clause.</p> <p>For example, the following query returns an error:</p> <pre> USE AdventureWorks SELECT FirstName, LastName AS 'LName' FROM Person.Contact ORDER BY p.LName </pre>

	SQL Server ignores the qualifier a in the ORDER BY clause and resolves the column name LName to the select list.	SQL Server does not the ORDER BY clause, name LName to tables clause. But the FROM recognize that column alias of table p.
SERVERPROPERTY function	The return type of the ProductVersion property in the SERVERPROPERTY function is varchar .	The return type of the property in the SERV function is nvarchar
sp_addtype	Any user can execute sp_addtype .	<p>Users must be a member of db_ddladmin or db_owner to execute sp_addtype.</p> <p>To enable users to create types, grant the following permissions to the user:</p> <ul style="list-style-type: none"> • To use sp_addtype, the user must be a member of the db_ddladmin or db_owner role. • To create an alias, the user must be granted the CREATE TYPE permission to the schema and the CONTROL permission for the schema.
sp_altermessage	sp_altermessage can be used to specify whether or not a system message (a message with Message ID < 50000) is to be written to the Windows Application log.	sp_altermessage can be used to change the logging behavior of system messages (messages with Message ID < 50000). The User Error Message (Message ID 50000) is not a system message. For more information, see Trace .
	Only members of the sysadmin fixed server role, the db_owner fixed database role, or a	A user that can execute sp_altermessage must have a server-level membership in the db_securityadmin fixed database role. The CONTROL permission is not securable. Executing sp_altermessage without CONTROL permission will fail.

sp_changedbowner	member of both the db_ddladmin and db_securityadmin fixed database roles can execute sp_changeobjectowner .	object will cause the following error message: "Msg 15247, Level 16, State 1, User does not have permission to perform this action."
sp_help	sp_help returns one result set for functions.	sp_help returns two result sets. The additional result set contains the additional result parameters.
sysindexes		sys.sysindexes is incompatible with SQL Server 2005. However, SQL Server 2005 prevents fully backward compatible versions of SQL Server.
System metadata	Members of the public role can query system tables and display metadata from the catalog.	Users querying the catalog of metadata for objects must have some permission on the object. For more information, see Metadata Visibility Troubleshooting .
System tables	System tables have a type of 'S'.	System tables are available as views and have a type of 'V'. Query system tables with the criteria type = 'S' will return system tables. Modify the statement to: "SELECT * FROM sys.tables WHERE type = 'S';" or "SELECT * FROM sys.views WHERE type = 'V';" Migrate to the new catalog management views. For more information, see Mapping SQL Server System Tables to SQL Server Views .

<p>TABLOCK when bulk importing using bcp, BULK INSERT, or OPENROWSET(BULK...)</p>	<p>For bulk importing into a table with a non-empty clustered index, the TABLOCK hint is ignored.</p>	<p>For bulk importing into an empty clustered index, the TABLOCK hint acquires an X lock on the table. This behavior prevents bulk import from running in parallel. If you want to bulk import in this case, use TABLOCK. For more information, see Optimizing Bulk Import.</p>
<p>Triggers</p>	<p>Direct recursion of triggers occurs only when a trigger fires and performs an action that causes the same trigger to fire again.</p>	<p>Direct recursion of triggers occurs in the following circumstances:</p> <ul style="list-style-type: none"> • A trigger fires and performs an action that causes the same trigger to fire again. • The same trigger fires again after a trigger of type AFTER or INSTEAD OF fires. <p>Indirect recursion occurs when a trigger fires and performs an action that causes a trigger of the same type (AFTER or INSTEAD OF) to fire. This trigger then performs an action that causes the original trigger to fire again.</p> <p>For more information, see the Triggers section under Triggers.</p>
<p>Triggers</p>	<p>When an UPDATE or DELETE statement is issued against a partitioned view, either local or distributed, any UPDATE or DELETE triggers that are defined on the base tables of the view fire. These include triggers on tables that are not affected by the update or delete operation.</p>	<p>When an UPDATE or DELETE statement is issued against a partitioned view, any UPDATE or DELETE triggers that are defined on the base table on which the view is based fire. For more information, see Execution.</p>

UPDATE() function	The UPDATE() function does not detect changes to timestamp columns. For these columns, an IF UPDATE() clause inside a trigger body returns FALSE, regardless of whether the columns are updated.	The UPDATE() function does not detect changes to timestamp columns. For these columns, an IF UPDATE() clause inside a trigger body returns TRUE if the columns are updated.
User-defined functions	User-defined functions cannot include nondeterministic built-in system functions.	Transact-SQL user-defined functions cannot include most nondeterministic built-in system functions. For a complete list of built-in functions, see Defined Functions (Transact-SQL) .
varchar , nvarchar , and varbinary data types	A zero-length string or binary value that is used as the definition of a computed table column creates a column of type varchar(0) , nvarchar(0) , or varbinary(0) .	A zero-length string or binary value that is used as the definition of a computed table column creates a column of type varchar(1) , nvarchar(1) , or varbinary(1) . This behavior affects only computed column values. Modify applications that use computed columns with a minimal length of 1 for varbinary columns, nvarchar columns, or varchar columns.
Virtual table access	Virtual tables can be accessed by guests or members of the public role.	VIEW SERVER STATE permission is required to access a virtual table, such as sys.dm_exec_query_plan.
WITH CHECK OPTION on views	Insert and update operations are allowed on views that specify the WITH CHECK OPTION clause and that are created over remote tables, even when those operations fall outside the	SQL Server 2005 added support for the WITH CHECK OPTION clause. Insert and update operations are allowed on views created over remote tables if the views specify the WITH CHECK OPTION clause. If insert and update operations are performed on views created over remote tables and the views do not specify the WITH CHECK OPTION clause, the operations fall outside the scope of the view's security.

	bounds of the SELECT statement of the view.	OPTION clause. For more information (Transact-SQL), ALTER SQL), and Modifying View.
xp_cmdshell	When an error occurs within the execution of xp_cmdshell , an error message is raised but execution is not terminated.	When an error occurs xp_cmdshell , an error execution is terminated.

Change History

Release	History
17 July 2006	<p>New content:</p> <ul style="list-style-type: none">• In the table that lists the Transact-SQL behavior changes, added the entry "Expressions in computed columns, CHECK constraints and DEFAULT constraints."• In the table that lists the Transact-SQL behavior changes under the "SQL Server 2005 behavior" column, changed the statement for the entry "varchar, nvarchar, and varbinary data types." The data type of a computed table column was changed from nvarchar(2) to nvarchar(1).
5 December 2005	<p>New content:</p> <ul style="list-style-type: none">• Added information about cursor declarations that are not valid.• Added "Changing the default language for the sa account" content.• Added "DBCC CHECKFILEGROUP" content.• Added "TABLOCK when bulk importing" content.• Added "Triggers" content. <p>Changed content:</p> <ul style="list-style-type: none">• Corrected the content in "varchar, nvarchar, and varbinary data types."

▣ **See Also**

Reference

[Breaking Changes to Database Engine Features in SQL Server 2005](#)

[Deprecated Database Engine Features in SQL Server 2005](#)

[Discontinued Database Engine Functionality in SQL Server 2005](#)

Other Resources

[SQL Server 2005 Database Engine Backward Compatibility](#)

sp_dbcmptlevel (Transact-SQL)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

SQL Server 2005 and SQL Server 2000



[Upgrading to SQL Server 2005](#) > [Backward Compatibility](#) > [SQL Server 2005 Database Engine Backward Compatibility](#) >

Topics in this section describe SQL Server 2005 support for the following legacy components and features:

See...	For information on...
Distributed Management Objects	Support for Distributed Management Objects in this SQL Server release.
English Query	Support for English Query in this SQL Server release.
Meta Data Services	Support for Meta Data Services in this SQL Server release.

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Distributed Management Objects



[See Also](#)



[Backward Compatibility](#) > [SQL Server 2005 Database Engine Backward Compatibility](#) > [SQL Server 2005 and SQL Server 2000](#) >

The Distributed Management Objects feature is installed by Microsoft SQL Server 2005 as a legacy component. To install the **Legacy Components** feature, click **Advanced** on the **Components to Install** page during SQL Server Setup. Expand the **Client Components** feature and then select **Legacy Components**.

▣ **See Also**

Reference

[Editions and Components of SQL Server 2005](#)

[Features Supported by the Editions of SQL Server 2005](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

English Query



[See Also](#)



[Backward Compatibility](#) > [SQL Server 2005 Database Engine Backward Compatibility](#) > [SQL Server 2005 and SQL Server 2000](#) >

English Query has been discontinued in this release. It cannot be installed or upgraded. If you upgrade a Microsoft SQL Server 2000 installation that includes English Query, English Query is not affected.

▣ **See Also**

Concepts

[SQL Server 2005 and SQL Server 2000](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Meta Data Services



[See Also](#)



[Backward Compatibility](#) > [SQL Server 2005 Database Engine Backward Compatibility](#) > [SQL Server 2005 and SQL Server 2000](#) >

Meta Data Services 3.0 is a component of Microsoft SQL Server 2000 that is no longer available in Microsoft SQL Server 2005. Upgrading does not delete the existing repository tables in the **msdb** database. However, the repository engine that reads and updates the tables is not available after you upgrade, and the repository tables are no longer accessible. To remove the unused tables, you can manually delete them. Repository tables can be identified by the **Rtbl** prefix.

Note:

Before you delete the tables, be sure that they do not contain DTS packages or SSAS metadata that you want to keep.

▣ **See Also**

Concepts

[SQL Server 2005 and SQL Server 2000](#)

Other Resources

Deleting a Table

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

SQL Server 2005 Analysis Services Backward Compatibility



[See Also](#)



[Upgrading to SQL Server 2005](#) > [Backward Compatibility](#) >

The topics in this section describe the changes in behavior between versions of Microsoft SQL Server Analysis Services. These changes include the following:

- Features that existed in Microsoft SQL Server 2000 Analysis Services but that have been removed in Microsoft SQL Server 2005 Analysis Services (SSAS).
- Features that have been retained in SQL Server 2005 Analysis Services for backward compatibility, but which will be removed in a future version of SQL Server.
- Features that have changed in SQL Server 2005 Analysis Services.

▣ In This Section

Topics	Description
Discontinued Analysis Services Functionality in SQL Server 2005	Describes features that existed in SQL Server 2000 Analysis Services but that have been removed in SQL Server 2005 Analysis Services.
Deprecated Analysis Services Functionality in SQL Server 2005	Describes features that have been retained in SQL Server 2005 Analysis Services for backward compatibility, but which will be removed in a future version of SQL Server.
Behavior Changes to Analysis Services Features in SQL Server 2005	Describes features that have changed in SQL Server 2005 Analysis Services.
Known Upgrade Issues for SQL Server 2005 Analysis Services	Describes issues that you may encounter when you upgrade from SQL Server 2000 Analysis Services to SQL Server 2005 Analysis Services.

▣ **See Also**

Concepts

[Migrating Existing Analysis Services Databases](#)

Other Resources

[SQL Server 2005 Analysis Services Backward Compatibility](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Deprecated Analysis Services Functionality in SQL Server 2005



[See Also](#)



[Upgrading to SQL Server 2005](#) > [Backward Compatibility](#) > [SQL Server 2005 Analysis Services Backward Compatibility](#) >

The following Analysis Services features have been retained in Microsoft SQL Server 2005 Analysis Services (SSAS) for backward compatibility, but will be removed in a future version of SQL Server. Deprecated features should not be used in new applications.

▣ **Deprecated Features**

Category	Deprecated Feature	Comments
Connection string properties	InsertInto	Original connection string syntax for populating local cubes.
Connection string properties	CreateCube	Original connection string syntax for populating local cubes. Not supported in ADOMD.NET.
Features	SQL Server 2000 PMML	This feature has been replaced with standard Predictive Model Markup Language (PMML).
Multidimensional Expressions (MDX)	Create Action	Extended syntax is supported, but English statements cannot create new types of actions. DDL should be used instead.
MDX	CalculationPassValue	
MDX	CalculationCurrentPass	
Data Mining Algorithms	Decision Trees: parameter Score_Method	
Other	CELL_EVALUATION_LIST intrinsic cell property	Blank in SQL Server 2005 Analysis Services.

▣ **See Also**

Concepts

[Migrating Existing Analysis Services Databases](#)

Other Resources

[SQL Server 2005 Analysis Services Backward Compatibility](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Discontinued Analysis Services Functionality in SQL Server 2005



[See Also](#)



[Upgrading to SQL Server 2005](#) > [Backward Compatibility](#) > [SQL Server 2005 Analysis Services Backward Compatibility](#) >

The following features that existed in Microsoft SQL Server 2000 Analysis Services have been removed from Microsoft SQL Server 2005 Analysis Services (SSAS).

□ Discontinued Features

Category	Discontinued Feature	Comments
Connection string properties	Mining Execution Location	Accepted for backward compatibility, but is ignored. Execution is always on the server.
Connection string properties	Mining Location	
Connection string properties	Log File	The Log File feature is replaced by the Trace feature.
Connection string properties	Execution Location	
Connection string properties	Distinct Measures by Key	
Connection string properties	Large Level Threshold	
Features	Aggregated Providers	This feature is replaced by plug-in algorithms.
Features	Linked Cubes	This feature is replaced by linked measure groups.
Features	Custom Level Formulas	This feature is replaced by Multidimensional Expressions (MDX) scripts.
Features	Cube and Database Role Commands	No longer supported.
MDX	CreateVirtualDimension	This MDX function is no longer supported.
MDX	CreatePropertySet	This MDX function is no longer supported.
MDX	Ignore	In SQL Server 2000 Analysis Services,

		this function was reserved for future use.
MDX	With Cache Create Cache	Accepted but is not supported.
Other	Active Directory Registration	
Other	Skipped levels in parent-child hierarchies	The SkippedLevelsColumn property, which indicates how many levels to skip when creating a member under a parent in a parent-child hierarchy, is no longer supported.

▣ **See Also**

Concepts

[Migrating Existing Analysis Services Databases](#)

Other Resources

[SQL Server 2005 Analysis Services Backward Compatibility](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Known Upgrade Issues for SQL Server 2005 Analysis Services



[See Also](#)



[Upgrading to SQL Server 2005](#) > [Backward Compatibility](#) > [SQL Server 2005 Analysis Services Backward Compatibility](#) >

The following table lists the issues that you may encounter when upgrading a SQL Server 2000 Analysis Services database to SQL Server 2005 Analysis Services (SSAS). The list below presents the upgrade issues and a brief description of each exactly as they appear in the report generated by the SQL Server 2005 Upgrade Advisor.

Issue Type	Issue Description
An object depending on a linked object is not migrated	Linked cubes and linked dimensions are not migrated by Upgrade Advisor in Microsoft SQL Server 2005 Analysis Services. Therefore, objects that refer to a linked cube or a linked dimension cannot be migrated because the linked object on which the object is based cannot be migrated.
Autoexist can produce different query results when multiple hierarchies are migrated into the same dimension	When multiple hierarchies or virtual dimensions are migrated into the same Microsoft SQL Server 2005 Analysis Services dimension, querying the migrated hierarchies contained in the dimension may produce different results than querying the same hierarchies when they were in SQL Server 2000 Analysis Services.
Browsing experience is different when disabled levels	Microsoft SQL Server 2005 Analysis Services no longer supports hidden or disabled levels in hierarchies. Hidden or disabled levels are migrated as visible and enabled levels. Calculations that involve hierarchies containing such levels

are used	might return unexpected results.
Bucketing may be different for grouping levels	Automatic grouping in Microsoft SQL Server 2005 Analysis Services might return a different set of member groups. Calculations that rely on these member groups might return unexpected results.
Conversion from neutral language to specific language may produce unexpected results	Neutral language identifiers used in previous versions of Analysis Services are migrated to specific language identifiers in Microsoft SQL Server 2005 Analysis Services. Migrating from neutral to specific language identifiers can change the expected translation and collation behavior, producing unexpected results.
Cube role commands are not supported	Microsoft SQL Server 2005 Analysis Services does not support command objects on cube roles and will not migrate commands from previous versions.
Custom level formulas aggregate differently	If a cube contains a dimension with custom level formulas, and also contains dimensions both before and after it with custom member formulas and/or unary operators, then the cube may return different results than previous versions of Analysis Services. This occurs because calculation precedence rules have changed.
Custom member formulas and custom rollup formulas are migrated into MDX script	Custom rollup formulas, custom member formulas, All member formulas, and custom level formulas are no longer supported as properties on dimensions or levels in Microsoft SQL Server 2005 Analysis Services.

<p>Custom aggregations are not migrated</p>	<p>Aggregations that were manually generated in previous versions of Analysis Services are not migrated in Microsoft SQL Server 2005 Analysis Services. Only aggregations generated by the Storage Design Wizard are migrated.</p>
<p>Data members always exist in parent-child dimensions</p>	<p>Previous versions of Analysis Services gave you the option of not including data members in parent-child dimensions. Data members can no longer be excluded from parent-child dimensions.</p>
<p>Database role commands are not supported</p>	<p>Microsoft SQL Server 2005 Analysis Services does not support command objects on database roles and will not migrate commands from previous versions of Analysis Services.</p>
<p>DefaultMember is migrated into MDX script</p>	<p>In previous versions of Analysis Services, the default member of a dimension is specified by a Multidimensional Expressions (MDX) expression, contained in the DefaultMember property of the dimension. In Microsoft SQL Server 2005 Analysis Services, this property is replaced by functionality supported in MDX scripts, and the property is upgraded to a MDX script during migration.</p>
<p>Dimension and hierarchy renaming by migration may cause different query results</p>	<p>Dimension hierarchies in previous versions of Analysis Services are internally represented as separate dimensions, and a naming convention is used to identify them. Migration to Microsoft SQL Server 2005 Analysis Services might create a separate dimension, with a new name, for each dimension hierarchy instead of combining the dimension hierarchies together under the parent dimension because auto-exist results in different security rules than would apply in previous versions of Analysis Services.</p>
<p>Drillthrough</p>	<p>While drillthrough exists in Microsoft SQL Server 2005</p>

settings are not migrated	Analysis Services, drillthrough settings are not migrated from previous versions of Analysis Services.
Hidden levels are visible after migration	Microsoft SQL Server 2005 Analysis Services no longer supports hidden or disabled levels in hierarchies. Hidden or disabled levels are migrated as visible and enabled levels. Calculations that involve hierarchies containing such levels might return unexpected results.
Invalid objects cause migration to fail	The database contains invalid objects. Migration can not be completed when the database contains invalid objects.
Linked cubes are not migrated	Previous versions of Analysis Services supported linked cubes. In Microsoft SQL Server 2005 Analysis Services, this feature has been replaced by linked dimensions and linked measure groups.
Member unique names may change during migration	Microsoft SQL Server 2005 Analysis Services attempts to preserve the unique names of members during migration, but there are certain circumstances in which the unique name for a member is changed. If member unique names change, client applications, Multidimensional Expressions (MDX) expressions, and other properties that depend on member unique names may produce unexpected results.
ODBC data sources are not supported	While previous versions of Analysis Services allowed you to use ODBC data sources, this functionality is no longer supported in Microsoft SQL Server 2005 Analysis Services.
Remote partitions are not migrated	Remote partitions are supported in Microsoft SQL Server 2005 Analysis Services, but they can not be migrated automatically until the remote server has been upgraded to SQL Server 2005.

Some mining model algorithm parameters are not supported	Some mining model algorithm parameters that are available in previous versions of Analysis Services are no longer supported in Microsoft SQL Server 2005 Analysis Services.
The behavior of some MDX functions is changed	Due to enhancements and changes in the Multidimensional Expressions (MDX) language for Microsoft SQL Server 2005 Analysis Services, certain MDX functions are either no longer supported, or behave differently than in previous versions of Analysis Services.
Top Level for dimension security is not supported	In previous versions of Analysis Services, you could specify dimension security so that a user saw a top level different than the top level of the hierarchy. Members that are secured using the Top Level setting will be visible after migration.
UDF source .dlls are not migrated and registered	User Defined Functions (UDFs) registered on previous versions of Analysis Services are not migrated.
Unary operators on non-parent-child hierarchies have limited support	Unary operators on hierarchies other than parent-child hierarchies are not supported in Microsoft SQL Server 2005 Analysis Services unless there is an attribute relationship between the attributes sourcing the levels in the hierarchy.
Virtual cubes are not supported	Previous versions of Analysis Services supported virtual cubes, a combined view of a subset of measures and dimensions from one or more regular or linked cubes. In Microsoft SQL Server 2005 Analysis Services, this feature has

	been replaced by linked measure groups. During upgrading, virtual cubes are migrated as linked measure groups.
Writeback table content is not migrated	The structure for writeback tables constructed by previous versions of Analysis Services has been updated in Microsoft SQL Server 2005 Analysis Services. Data from writeback tables constructed by previous versions of Analysis Services cannot be migrated into the new writeback table structure.

▣ **See Also**

Other Resources

[SQL Server 2005 Analysis Services Backward Compatibility](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Behavior Changes to Analysis Services Features in SQL Server 2005

 [See Also](#)



[Upgrading to SQL Server 2005](#) > [Backward Compatibility](#) > [SQL Server 2005 Analysis Services Backward Compatibility](#) >

In Microsoft SQL Server 2005 Analysis Services (SSAS), the behavior of certain features has changed compared with previous versions of Analysis Services. These changes are listed in the following table.

□ Behavior Changes

Feature	SQL Server 2000 Behavior	SQL Server 2005 Behavior
Browsing	Supports hidden or disabled levels in hierarchies.	Hidden or disabled levels are migrated as visible and enabled levels. Calculations that involve hierarchies that contain such levels might return unexpected results.
Dimension	Supports the option to not include data members in parent-child dimensions.	Data members always exist in parent-child dimensions; data members can no longer be excluded from parent-child dimensions.
Multidimensional Expressions (MDX)	Function: CreateVirtualDimension	Expression will return Error.
MDX	Function: CreatePropertySet	Expression will return Error.
MDX	Statement: With Cache	Accepted syntax, but does nothing.
MDX	Statement: Create Cache	Accepted syntax, but does nothing.
MDX	Calculation precedence rules.	Calculation precedence rules have changed from previous versions of Analysis Services. Because of the above custom rollup, expressions may return different results than with previous versions of Analysis Services.
Security	Supports the option to specify dimension security so that a user can view a top level that is different from the top level of the hierarchy.	Top Level for dimension security is not supported. Members that are secured by using the Top Level setting will be visible after migration.
		Tries to commit a non-existing transaction fail. In SQL Server 2000

Other	AutoCommit and AutoRollback in update.	Analysis Services, the Update statement created its own transaction. In SQL Server 2005 Analysis Services, it does not.
-------	--	---

▣ **See Also**

Concepts

[Migrating Existing Analysis Services Databases](#)

Other Resources

[SQL Server 2005 Analysis Services Backward Compatibility](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Reporting Services Backward Compatibility



[See Also](#)



[Upgrading to SQL Server 2005](#) > [Backward Compatibility](#) >

This section describes changes in behavior between versions of SQL Server Reporting Services. It covers features that are no longer available or are scheduled to be removed in a future release. It also describes fundamental changes to the product that are known to break a custom application that includes Reporting Services functionality.

▣ **In This Section**

- [Deprecated Features in SQL Server 2005 Reporting Services](#)
- [Discontinued Functionality in SQL Server 2005 Reporting Services](#)
- [Breaking Changes in SQL Server 2005 Reporting Services](#)
- [Behavior Changes in SQL Server 2005 Reporting Services](#)

▣ **See Also**

Concepts

[Upgrading Reporting Services](#)

Other Resources

[Backward Compatibility](#)

[Upgrading SQL Server Components](#)

Reporting Services Enhancements

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Deprecated Features in SQL Server 2005 Reporting Services



[See Also](#)



[Upgrading to SQL Server 2005](#) > [Backward Compatibility](#) > [Reporting Services Backward Compatibility](#)
>

Updated: **5 December 2005**

This topic describes the SQL Server Reporting Services features that are scheduled to be removed in the next release. The following features are still available in SQL Server 2005.

▣ **SQL Server 2000 Report Server Web Service Endpoint**

The SQL Server 2000 Report Server Web service endpoint has been deprecated in this release. The previous endpoint has been replaced by two new endpoints: **ReportingService2005** and **ReportExecutionService**.

The new endpoints include all the functionality available in the previous endpoint, plus the new features introduced in SQL Server 2005. For more information, see **Reporting Services Programmability Enhancements** and **Report Server Web Service Endpoints** in SQL Server Books Online.

▣ **HTML OWC Rendering Extension**

The HTML with Office Web Components (OWC) rendering extension has been deprecated in this release. If you upgrade from the previous release, it will continue to be available. New installations do not include the HTML OWC rendering extension as an export option. If you want to use the deprecated extension in a new installation, you can add entries in the `RSReportServer.config` and `RSReportDesigner.config` files to make the extension available. The following example illustrates the entry you must add:

```
<Extension Name="HTMLOWC" Type="Microsoft.ReportingServices.Renderin
```

▣ **Internet Explorer 5.5 Support**

This version of Microsoft Internet Explorer for Windows is deprecated in SQL Server 2005 Service Pack 2. For more information about browser types that are supported, see **Browser Support in Reporting Services** in SQL Server Books Online.

▣ **HTML 3.2 Rendering Extension**

The HTML 3.2 format in the HTML rendering extension has been deprecated in this release. HTML 3.2 cannot be selected as an export option in the report toolbar. To render reports in the HTML 3.2 format, you must specify it on a report URL using the Format parameter. For more information about how to specify a rendering extension, see **Specifying a Rendering Format in a URL** in SQL Server Books Online.

▣ **RS Utility support for the SQL Server 2000 Report Server Web Service Endpoint**

In this release of Reporting Services, the report server script host command line utility (rs.exe) supports the SOAP endpoint of the previous release. In subsequent releases, support for that endpoint will be removed from the product. Whenever possible, use the new SOAP endpoints in all new and revised scripts. For more information, see **Report Server Web Service Endpoints** and **rs Utility** in SQL Server Books Online.

▣ Change History

Release	History
12 December 2006	New content: <ul style="list-style-type: none"><li data-bbox="527 506 1258 541">• Deprecation notice for Internet Explorer 5.5..
5 December 2005	New content: <ul style="list-style-type: none"><li data-bbox="527 674 1356 751">• Deprecation notice for the SQL Server 2000 Report Server Web service endpoint.

▣ **See Also**

Concepts

[Upgrading Reporting Services](#)

Other Resources

[Upgrading SQL Server Components](#)

[Reporting Services Backward Compatibility](#)

[Backward Compatibility](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Discontinued Functionality in SQL Server 2005 Reporting Services



[See Also](#)



[Upgrading to SQL Server 2005](#) > [Backward Compatibility](#) > [Reporting Services Backward Compatibility](#)
>

This topic describes the SQL Server Reporting Services features that are no longer available in SQL Server 2005.

▣ **Rsactivate.exe**

Rsactivate.exe is a command line utility that was used to initialize a single report server instance and multiple report server instances that were part of a single Web farm. Due to changes in how initialization and configuration are now performed, this utility is now obsolete and no longer provides this functionality. To create a scale-out deployment in SQL Server 2005, you use the Reporting Services Configuration tool. For more information about initialization changes, see [Behavior Changes in SQL Server 2005 Reporting Services](#). For more information about configuring multiple report servers to use a single report server database, see [Configuring a Report Server Scale-Out Deployment](#).

▣ **See Also**

Concepts

[Upgrading Reporting Services](#)

Other Resources

[Upgrading SQL Server Components](#)

[Reporting Services Backward Compatibility](#)

[Backward Compatibility](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Breaking Changes in SQL Server 2005 Reporting Services



[See Also](#)



[Upgrading to SQL Server 2005](#) > [Backward Compatibility](#) > [Reporting Services Backward Compatibility](#)
>

Updated: **12 December 2006**

This topic describes changes to Reporting Services functionality that will break custom applications, scripts, or reports.

▣ **No support for drillthrough actions or bookmark links in a header or footer**

In SQL Server 2005 and SQL Server 2005 Service Pack 1, Reporting Services does not support drillthrough actions or bookmark links in a header or footer. This is a breaking change for reports that you created in SQL Server 2000 Reporting Services.

In Service Pack 2, this issue is partially resolved. Drillthrough actions in the page header and footer are supported for reports that run on-demand, but not for reports that run as a snapshot. Snapshots that contain drillthrough or bookmark links in the header or footer must be recreated to obtain the drillthrough functionality:

1. Start Report Manager or Management Studio.
2. Locate the report and open the Execution properties page for that report.
3. Select the **Create a snapshot of the report when this page is saved** check box.
4. Click **Apply**.
5. Repeat these steps for all reports that run as a snapshot and that have drillthrough or bookmark links in the header or footer.

▣ **Report Server WMI Provider from SQL Server 2000 is not supported**

Reporting Services includes a Windows Management Instrumentation (WMI) provider that you can use to programmatically configure the environment in which a report server runs. The SQL Server 2005 release of Reporting Services includes an all-new version of the WMI provider that completely replaces the earlier version. The original version from SQL Server 2000 is not supported in this release of SQL Server.

▣ **New authentication restrictions for data processing extensions**

This release of Reporting Services includes new restrictions on how credentials are handled for some data processing extensions:

- You cannot configure an Analysis Services data source connection to use database credentials. The connection will fail if you are using the ADOMD wrapper.
- You cannot configure an XML data source connection to use database credentials. If you specify no credentials and the unattended execution account is not defined, the request is made through Anonymous access.
- Third-party .NET data processing extensions (for example, DB2.NET, Oracle ODP.NET, and Teradata) explicitly fail data processing if database credentials are passed on the connection string and the unattended execution account is not configured.
- Custom data processing extensions that implement **IDBConnection** (but not **IDBConnectionExtension**) explicitly fail data processing in every case unless the unattended execution account is configured.
- Whenever connection requests are made by the service account (that is, the security context of the service account is used for the network connection to the data source), Reporting Services removes administrator privileges from the service account before it sends the request.

To configure the unattended execution account, use the Reporting Services Configuration tool or **rsconfig** utility. For more information, see **Specifying Connections for Custom Data Processing Extensions** and [Configuring an Account for Unattended Report Processing](#).

▣ Expressions that contain a reference to the object result in report compilation errors

Reports that use expressions to set the **Action** property on an item will no longer compile if the expression includes a **Me.Value** reference. Upgrading reports that include **Me.Value** in **Action** property expressions will result in compiler errors that resemble "The Value expression for the textbox 'textbox1' contains an error: [BC30456] 'Value' is not a member of 'ReportExprHostImpl.textbox1_TextBoxExprHost.DataValue1_DataValueExprHost (rsCompilerErrorInExpression)."

To avoid these errors and upgrade your reports, you must rewrite **Action** expressions, replacing **Me.Value** with the actual value expression. The following example illustrates how to replace **Me.Value** in a textbox-style expression with an actual value expression: `=iif(Fields!Name.Value = 0, "Solid", "None").`

▣ **Support for multivalued parameters might break single-valued parameters in reports**

Multivalued parameter support is a new feature that is introduced in this release. If you are using the old SOAP endpoint and your code gets or sets multivalued parameters, you will now get multivalued parameters back. In the previous release, if you specified multiple parameters, only the first parameter was returned. Be aware that after you upgrade, additional parameters might be returned for parameters that you thought were single-valued. Make sure that you review existing calls to the **Render** method to determine whether single or multivalued parameters are used. You should also test all reports that have parameters to verify that single-valued parameters work as expected.

▣ **Updating parameters through SetReportParameters**

In the previous release, if you wanted to update a parameter property, you had to re-specify all its settings. If the report included query-based or expression-based parameters, updating a single parameter property usually meant you had to republish the report. Failure to republish the report resulted in reports that were missing query-based default values after a parameter property was changed.

In this release, you can update a specific property on a parameter without having to reset all its properties. The default values are no longer set to null if the **DefaultValues** property is unspecified. Be aware that this change might break application code or scripts that use the old SOAP endpoint if the code or script is handling the Null reference.

▣ Adding the **Select All** member for multivalue parameters

In a SQL Server 2005 Reporting Services report, a **Select All** check box was automatically added when you created an available values list for a multivalue report parameter. Checking the **Select All** check box selected all available values. Clearing the **Select All** check box cleared all available values. If you upgraded to SQL Server 2005 Service Pack 1 (SP1), the **Select All** check box was no longer available. In SQL Server 2005 Service Pack 2 (SP2), the **Select All** check box has been restored. When you upgrade your report authoring client to SP2, the **Select All** check box is automatically added to the drop-down list of available values in report preview in Report Designer. After you upgrade your report server to SP2, you can use the report server tools (Report Manager, Report Builder, and SQL Server Management Studio) to view all existing and new reports that use the **Select All** functionality. You do not have to republish existing reports to restore the **Select All** functionality. If you upgrade your tools and you do not see the **Select All** check box, you are probably connected to an SP1 report server.

▣ Referencing extended properties from the Microsoft SQL Server Analysis Services data processing extension

In a SQL Server 2005 Reporting Services report, if you needed to refer to the extended **Key** property for a field from an Analysis Services data source, you needed to use the custom field syntax `Fields!FieldName("MEMBER_KEY")`. After you upgrade to SQL Server 2005 Service Pack 1 (SP1), this syntax no longer returns the **Key** value. This workaround is no longer needed or supported. Instead, you must change references to the extended field syntax for the **Key** property to `Fields!FieldName.Key` or `Fields!FieldName("Key")`.

Change History

Release	History
12 December 2006	<p>New content:</p> <ul style="list-style-type: none">• No support for drillthrough actions. <p>Changed content:</p> <ul style="list-style-type: none">• Select All option restored for SP2.

▣ **See Also**

Concepts

[Upgrading Reporting Services](#)

Other Resources

[Upgrading SQL Server Components](#)

[Reporting Services Backward Compatibility](#)

[Backward Compatibility](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Behavior Changes in SQL Server 2005 Reporting Services



[See Also](#)



[Upgrading to SQL Server 2005](#) > [Backward Compatibility](#) > [Reporting Services Backward Compatibility](#)
>

This topic describes fundamental changes to SQL Server Reporting Services functionality that may affect how you work.

▣ Initialization and Encryption Changes

The Report Server Web service is no longer used to initialize a Reporting Services installation. In this release, only the Report Server Windows service performs initialization. When the Report Server Web service requires encryption or decryption, it calls the Report Server Windows service to get the keys and perform the operation. Before it issues the request, the Web service performs a check to see if it is running as the account specified in the **WebServiceAccount** setting in the RSReportServer.config file. If the account that is running the Web service does not match the specified value, the Web service returns an error indicating an account mismatch.

This change in initialization behavior introduces a stronger dependency on the Report Server Windows service by the Report Server Web service. Running the Report Server Windows service is now a prerequisite for most Web service operations. For more information, see [Initializing a Report Server](#).

▣ **Setup Commands**

In this release, the scope of operations that can be performed through command line Setup has been reduced. Command line Setup can no longer be used to both install and deploy an installation that includes remote computers. Setup no longer deploys a Report Server Web farm installation. If you used Setup in the past to deploy a Reporting Services installation, you must use alternate tools and methods to deploy the software. If you created batch files and setup scripts to deploy Reporting Services components, they will no longer work after you upgrade.

▣ **The DefaultValueQueryBased property is now read-write**

The DefaultValueQueryBased property is a Boolean property that indicates whether the default values of a parameter are based on a query. In the previous release, this property was read-only. This property has been modified to accept write operations, allowing you to programmatically update this value for a published report. If you use this property in your code, be sure to test the application to verify that it behaves as expected.

▣ **Excel rendering extension supports Binary Interchange File Format**

The Excel rendering extension renders reports in the native file format used by Microsoft Excel, eliminating most of the limitations that were part of the previous version.

▣ **Header Row Height Changes**

In the previous release, the first row in a multi-line header determined the row height for all rows in the header (that is, row height was fixed based on the height of the first row). Now, row heights can be variable. If you have reports that contain multi-line headers, you should inspect the reports after you upgrade to verify the row height is correct.

▣ **Leading blank characters in role definition names**

In the previous release, leading blank spaces were allowed in role definition names. This release no longer supports leading blank spaces in new role names that you create. If you already have role definitions that begin with leading blank characters, they will continue to work. This functional change was implemented in response to the behavior of the **DeleteRole** method. If **DeleteRole** detects empty spaces at the beginning of a role definition name, it deletes the role.

▣ **See Also**

Concepts

[Upgrading Reporting Services](#)

Other Resources

[Upgrading SQL Server Components](#)

[Reporting Services Backward Compatibility](#)

[Backward Compatibility](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

SQL Server 2005 Integration Services Backward Compatibility



[See Also](#)



[Upgrading to SQL Server 2005](#) > [Backward Compatibility](#) >

Updated: 17 July 2006

SQL Server 2005 Integration Services (SSIS) provides management, run-time, and design-time support for Data Transformation Services (DTS) packages that were created by using the SQL Server 2000 tools and object model.

[General Compatibility between DTS and SSIS](#)

- [Managing DTS packages](#)
- Run-time support for DTS packages
 - [Installing run-time support for DTS packages](#)
 - [Running DTS packages from the command prompt](#)
 - [Running DTS packages from Integration Services packages](#)
- [Design-time support for DTS packages](#)
- Migrating
 - [Migrating dtsrun commands to dtexec commands](#)
 - [Migrating DTS packages to Integration Services packages](#)
- Limited support
 - [Support for Integration Services in SQL Server 2005 Express Edition](#)
 - [Support for Repository packages](#)
 - [Support for ODBC destinations](#)

▣ **General Compatibility between DTS and SSIS**

Because Integration Services is an entirely new product, and not a version upgrade, you will not encounter software conflicts between DTS and SSIS.

- You can install both DTS and SSIS on the same server.
- You can run both DTS and SSIS packages on the same server when both products are installed. You can run DTS packages, even when the SQL Server 2000 tools are not present, by using an updated version of the DTS runtime that is installed with Integration Services.
- You can save DTS packages in the MSDB database of a SQL Server 2000 or SQL Server 2005 instance. You can save SSIS packages only on a SQL Server 2005 instance.
- You can edit DTS packages, even when the SQL Server 2000 tools are not present, by using an updated version of the DTS Designer that is available for download. You cannot edit DTS packages in Business Intelligence Development Studio. You can edit SSIS packages only in BI Development Studio.

▣ **Managing DTS Packages**

You can manage DTS packages from within SQL Server Management Studio, whether they are stored in an instance of SQL Server 2000 or they have been imported to an instance SQL Server 2005. To view DTS packages in Management Studio, connect to an instance of SQL Server and look in the Object Browser under the Management\Legacy\Data Transformation Services node.

When you select an individual DTS package in Management Studio, you can open the package for editing, migrate it, export it to a structured storage file, or delete it.

When you select the Data Transformation Services node in Management Studio, you can open packages, import them to SQL Server from structured storage files, or launch the Package Migration Wizard.

Note:

To edit DTS packages, a separate download of the DTS designer components is required. See [Installing Design-time Support for SQL Server 2000 DTS Packages](#). DTS packages cannot be opened or edited in BI Development Studio.

▣ **Installing Run-time Support for SQL Server 2000 DTS Packages**

You can run DTS packages and SSIS packages on the same computer.

When you select **Integration Services** for installation, Setup also installs support for DTS packages, including the DTS runtime and DTS package enumeration in SQL Server Management Studio. Support in the runtime is enhanced to enable DTS packages to access SQL Server 2005 data sources.

For more information, see [How to: Ensure Support for Data Transformation Services Packages](#).

Note:

Although support for SQL Server 2000 Data Transformation Services is installed automatically when you select Integration Services for installation, some of the registry and environment settings required by DTS are removed if you upgrade the last instance of SQL Server 2000 on the computer to SQL Server 2005 during or after setup. To restore these settings, locate "Microsoft SQL Server 2005 Backward compatibility" in the **Add or Remove Programs** list in the Control Panel, click the **Change** button, and select the **Repair** option on the **Program Maintenance** page of the wizard.

The backward compatibility files that are installed also include run-time support for the SQL Server 2000 Analysis Services Processing task and its dependency, Decision Support Objects (DSO). However they do not include design-time support for editing DTS packages that contain the SQL Server 2000 Analysis Services Processing task. You can edit packages that contain this task only when you have SQL Server 2000 Analysis Services installed on the same computer.

The backward compatibility files do not include the SQL Server 2000 Data Mining Prediction Query task.

▣ **Running DTS Packages from the Command Prompt**

You can run SQL Server 2000 DTS packages from the command prompt. SQL Server 2005 Integration Services (SSIS) installs the SQL Server 2000 **dtsrun.exe** utility when it installs DTS run-time support.

▣ **Running DTS Packages from Integration Services Packages**

You can also run packages that were created by using SQL Server 2000 tools as part of a SQL Server 2005 data transformation solution. Integration Services provides the **Execute DTS 2000 Package** task for running DTS packages from within SQL Server 2005 Integration Services packages.

For more information, see **Execute DTS 2000 Package Task**.

▣ **Installing Design-time Support for SQL Server 2000 DTS Packages**

SQL Server 2005 Integration Services (SSIS) does not install the DTS designer. An updated version of the DTS package designer is available as a separate Web download. Download the [Microsoft SQL Server 2000 DTS Designer Components](#).

For information on managing certain DTS application properties that affect the DTS design environment, see [Setting DTS Application Properties](#).

▣ **Migrating Command Lines for Executing Packages**

To understand how the command prompt options used with the SQL Server 2000 **dtsrun** utility map to the command prompt options used with the SQL Server 2005 **dtexec** utility, see **dtsrun to dtexec Command Option Mapping**.

▣ Migrating DTS Packages

SQL Server 2005 provides the Package Migration Wizard for migrating SQL Server 2000 DTS packages to the Integration Services package format. Many packages can be fully migrated to SQL Server 2005, especially DTS packages that use tasks, connections, and other objects that have an equivalent in SQL Server 2005. Other packages can be migrated successfully by encapsulating certain tasks or features that cannot be migrated within an Execute DTS 2000 Package task. Before running the Package Migration Wizard, you may want to install and run Upgrade Advisor to analyze your existing packages for anticipated migration issues.

For more information about Upgrade Advisor, see [Using Upgrade Advisor to Prepare for Upgrades](#).

 **Note:**

The Package Migration Wizard is available in the Standard, Enterprise, and Developer Editions of SQL Server 2005.

 **Note:**

Upgrade Advisor cannot analyze, and the Package Migration Wizard cannot migrate, DTS packages that are saved in the **msdb** database of an instance of SQL Server 7.0. However the Upgrade Advisor can analyze, and the Package Migration Wizard can migrate, SQL Server 7.0 DTS packages that are saved as structured storage files, or that are saved in the **msdb** database of an instance of SQL Server 7.0 that has been upgraded to SQL Server 2005.

It is not possible to save or export Integration Services packages to the SQL Server 2000 DTS package format.

For more information, see [Migrating Data Transformation Services Packages](#).

▣ **Support for SQL Server 2000 DTS Packages in SQL Server 2005 Express Edition**

SQL Server 2005 Express Edition does not include Integration Services or support for SQL Server 2000 DTS packages.

- To run existing DTS packages on a SQL Server 2005 Express Edition server, you must leave the SQL Server 2000 client tools or the DTS redistributable files on the server, or reinstall them. SQL Server 2005 Express Edition does not include the DTS runtime.
- To edit existing DTS packages on a SQL Server 2005 Express Edition server, you must use SQL Server 2000, or edit the packages remotely from a server running SQL Server 2005 Workgroup, Standard, Enterprise, or Developer Edition. SQL Server 2005 Express Edition does not include SQL Server Management Studio or BI Development Studio.
- To migrate existing DTS packages to SQL Server 2005 Integration Services, you must use SQL Server 2005 Standard, Enterprise, or Developer Edition. Other SQL Server 2005 editions do not include the Integration Services Package Migration Wizard.

The Import and Export Utility that is included with SQL Server 2005 Express Edition is not the SQL Server Import and Export Wizard and does not use Integration Services.

▣ Support for Meta Data Services (Repository) Packages

SQL Server 2000 Meta Data Services, commonly known as the Repository, is a deprecated component. SQL Server 2005 does not install or use the Repository.

The SQL Server 2000 Data Transformation Services (DTS) designer and the **dtsrun.exe** utility continue to support DTS packages that were saved to Meta Data Services.

SQL Server 2005 Integration Services supports the Repository only in the Upgrade Advisor and the Package Migration Wizard, and only if SQL Server 2000, the SQL Server 2000 tools, or the Repository redistributable files are installed on the local computer. When the Repository files are present, the Upgrade Advisor can scan, and the Package Migration Wizard can migrate, DTS packages that were saved to Meta Data Services. When the Repository files are not present, the Upgrade Advisor can only scan, and the Package Migration Wizard can only migrate, DTS packages that were saved to SQL Server or to structured storage files.

The Integration Services Execute DTS 2000 Package task cannot execute a DTS package that was saved to the Repository. However the SQL Server 2000 DTS runtime can execute Repository packages. As a workaround for this limitation in Integration Services, when the Repository files are present, you can create a new SQL Server 2000 DTS package as a wrapper and save this new package to SQL Server or as a structured storage file. Use an Execute Package task in the new DTS package to execute the Repository package, and use the Execute DTS 2000 Package task in the Integration Services package to execute the new wrapper package.

▣ **Support for ODBC Destinations**

SQL Server 2005 Integration Services has an ODBC source component, for loading data from ODBC data sources, but does not have a corresponding ODBC destination component, for saving data to ODBC destinations. It is possible to create an ad hoc ODBC destination for use within a single package by using the Script component. For more information, see **Creating an ODBC Destination with the Script Component**.

▣ Change History

Release	History
17 July 2006	<p>Changed content:</p> <ul style="list-style-type: none">• Added section about ODBC destinations.• Added section about compatibility of DTS and SSIS packages.• Noted restriction on exporting packages to DTS 2000 format, and on design-time support for 2000 packages.
14 April 2006	<p>Changed content:</p> <ul style="list-style-type: none">• Added list of all the operations that can be performed on DTS packages with SQL Server 2005 tools.• Added section on Managing DTS Packages.• Added section on Running DTS Packages from the Command Prompt.• Added note about Upgrade Advisor to section on migration.• Provided link for download of DTS designer components.
5 December 2005	<p>Changed content:</p> <ul style="list-style-type: none">• Added information about downloading the DTS Designer and configuring its options.• Added note and link about migrating dtsrun.exe command lines.• Added information about support for migration from SQL Server 7.0.

▣ **See Also**

Concepts

[Known Package Migration Issues](#)

Other Resources

dtsrun to dtexec Command Option Mapping

Integration Services Considerations on 64-bit Computers

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Breaking Changes to Integration Services Features in SQL Server 2005



[Upgrading to SQL Server 2005](#) > [Backward Compatibility](#) > [SQL Server 2005 Integration Services Backward Compatibility](#) >

New: 17 July 2006

This topic describes changes made to SQL Server 2005 Integration Services (SSIS) after the initial release of SQL Server 2005 that could cause applications based on earlier versions of Integration Services to break.

▣ Previously Successful Column Conversions May Fail

After you apply SQL Server 2005 Integration Services (SSIS) Service Pack 1 (SP1), column conversions may fail in the following cases:

Column Conversion	Issue Description
Conversions from String to Unsigned Integer	<p>If you have negative numbers in string columns (DT_STR or DT_WSTR) and convert the column data type to an unsigned integer data type (DT_UI1, DT_UI2, DT_UI4 or DT_UI8) the package previously ran successfully, and the negative values were converted to 0. After you apply SP1, the package no longer converts negative values to 0, but instead returns an error and the package may fail.</p> <p>Applies to: Lookup transformation, Fuzzy Lookup transformation, Fuzzy Grouping transformation, and the Flat File source.</p>
Conversions from String to Binary	<p>If you have characters in string columns (DT_STR or DT_WSTR) that are not convertible to a byte array, and you convert the column data type to the DT_BYTES data type, data might be silently corrupted; that is, the value of the column is converted to an array of zeroes. The installation of SP1 adds the UseBinaryFormat property to the Flat File source to help interpret binary data. After you apply SP1, when UseBinaryFormat is set to false (the default value), if a column contains characters that cannot be converted, the conversion generates an error, and packages that previously ran successfully now may fail.</p> <p>Applies to: Flat File source.</p>

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Deprecated Features in SQL Server 2005 Integration Services



[See Also](#)



[Upgrading to SQL Server 2005](#) > [Backward Compatibility](#) > [SQL Server 2005 Integration Services Backward Compatibility](#) >

New: 14 April 2006

This topic describes the SQL Server 2005 Integration Services features that are scheduled to be removed in future releases of the product. The following features are still available in SQL Server 2005.

▣ **SQL Server 2000 Data Transformation Services (DTS) Object Model**

The SQL Server 2000 Data Transformation Services (DTS) object model is deprecated. You should plan to remove dependencies on this object model in anticipation of the next release of SQL Server. The SQL Server 2000 DTS object model itself will continue to be supported in SQL Server 2000 and SQL Server 2005 as part of our ongoing commitment to provide support for those versions of SQL Server; however, the ability to migrate SQL Server 2000 packages may not be supported in the next release.

The DTS functionality that will be deprecated includes the DTS API, the Package Migration Wizard for migrating DTS packages to the next version of SQL Server, support for DTS package maintenance in SQL Server Management Studio, and the Execute DTS 2000 Package task.

▣ **ActiveX Script Task**

The ActiveX Script task in SQL Server 2005 Integration Services (SSIS) is provided only for backward compatibility with DTS and will be removed in the next version of SQL Server. Do not use this feature in the development of new packages. You should modify packages that currently use this feature as soon as possible. You may be able to recreate the functionality of your ActiveX scripts by using the new components and the more flexible precedence constraints available in Integration Services. Or, you can rewrite your scripts by using the Integration Services Script task.

▣ Microsoft Visual Studio for Applications (VSA)

In future versions of Integration Services, Visual Studio for Applications (VSA) will be replaced with Visual Studio Tools for Applications (VSTA) as the engine in the Script task and Script transformation. While this change is not anticipated to incur a dramatic departure from existing syntax, this change may require script conversions when migrating to the next version of Integration Services.

In the next version of Integration Services:

- Microsoft will support the conversion to VSTA of existing Script task and Script transformation scripts authored in VSA.
- Microsoft will support execution of existing non-compiled VSA scripts.

In future versions of Integration Services:

- Microsoft will support the conversion to VSTA of existing Script task and Script transformation scripts authored in VSA
- Microsoft will not support execution of VSA scripts.

Change History

Release	History
17 July 2006	Changed content: <ul style="list-style-type: none">• Added detail about support for ActiveX script.

▣ **See Also**

Other Resources

[Upgrading SQL Server Components](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Setting DTS Application Properties



[See Also](#)



[Upgrading to SQL Server 2005 > Backward Compatibility > SQL Server 2005 Integration Services Backward Compatibility >](#)

New: 5 December 2005

For users of SQL Server 2005 Integration Services (SSIS) who still need to run SQL Server 2000 Data Transformation Services (DTS) packages, Microsoft provides an updated version of the DTS runtime engine as part of SQL Server 2005 Integration Services (SSIS) Setup, and an updated version of the DTS designer as part of a Web download. The SQL Server 2005 tools provide access to almost all DTS features through the familiar designer and run-time engine.

However, the SQL Server 2005 tools no longer provide access to the application-wide settings in the DTS **Package Properties** dialog box, which was accessed in SQL Server 2000 Enterprise Manager by selecting **Properties** from the popup menu on the **Data Transformation Services** node. This topic describes these settings and how to configure them manually, by editing the applicable Registry entries, or programmatically, by calling the corresponding methods in the DTS object model that modify the Registry entries for you.

▣ **DTS Application Properties**

The following DTS application properties are discussed in this topic:

Cache: Turn on cache (check box)

Caches lists of the available tasks, transformations, scripting languages, and OLE DB providers to improve performance.

Cache: Refresh cache (button)

Refreshes the cached lists described above.

Designer: Show multi-phase pump in DTS designer

Enables display of advanced options that allow specific phases of a data transformation, such as "Pre source data" or "On transform failure," to be configured individually.

Debugging: Turn on just-in-time debugging

Enables debugging for developers of custom tasks.

Meta Data Services: Enable Save to Meta Data Services

Allows packages to be saved to Meta Data Services (the Repository).

▣ Note:

Support for the Repository is limited in SQL Server 2005. For more information, see "Meta Data Services (Repository) Packages" in [SQL Server 2005 Integration Services Backward Compatibility](#).

▣ **Setting the DTS Application Properties in the Registry**

To enable or disable the following DTS application properties manually, by modifying the Registry directly, set the following REG_DWORD values to 0 (enabled) or 1 (disabled) under the parent key

HKEY_CURRENT_USER\Software\Microsoft\Microsoft SQL Server\80\DTS\Settings:

Turn on cache

TasksCache

TransformationsCache

ScriptingLanguagesCache

OLEDBProvidersCache

Show multi-phase pump in DTS designer

DesignerSettings

Turn on just-in-time debugging

JITDebug

Note:

It is not possible to refresh the cache by editing Registry values. The cache can only be refreshed programmatically.

To enable or disable the following DTS application property manually, by modifying the Registry directly, set the following REG_DWORD value to 0 (enabled) or 1 (disabled) under the parent key

HKEY_LOCAL_MACHINE\Software\Microsoft\Microsoft SQL Server\80\DTS\Settings:

Enable Save to Meta Data Services

EnableSaveToRepository

Caution:

The following warning was displayed by the **Package Properties** dialog box when the **Enable Save to Meta Data Services** property was enabled:
"Warning: Packages saved to Meta Data Services should not be considered secure. Are you sure you want to enable saving packages to Meta Data Services?"

Sample Registry Script

The following Registry script, when saved in a text file with the **.reg** extension and run, enables all the DTS application properties discussed in this topic. To disable the properties instead, change the new values from 00000001 to 00000000.

Windows Registry Editor Version 5.00

```
[HKEY_CURRENT_USER\Software\Microsoft\Microsoft SQL Server\80\DTS\Se
"TasksCache"=dword:00000001
"TransformationsCache"=dword:00000001
"ScriptingLanguagesCache"=dword:00000001
"OLEDBProvidersCache"=dword:00000001
"DesignerSettings"=dword:00000001
"JITDebug"=dword:00000001
```

```
[HKEY_LOCAL_MACHINE\Software\Microsoft\Microsoft SQL Server\80\DTS\S
"EnableSaveToRepository"=dword:00000001
```

▣ **Setting the DTS Application Properties in the DTS Object Model**

To enable or disable the following DTS application properties programmatically, set the following values of the DTS **Application** object to **True** (enabled) or **False** (disabled).

Turn on cache

.TaskInfos.UseCache

.TransformationInfos.UseCache

.ScriptingLanguageInfos.UseCache

.OLEDBProviderInfos.UseCache

Turn on just-in-time debugging

.JITDebug

To enable or disable the following DTS application option programmatically, set the following values of the DTS **Application** object to the appropriate value from the **DTSDesignerSettings** enumeration:

- Enabled - **DTSDesignerSettings.DTSDesigner_ShowMultiPhaseTransforms**
- Disabled - **DTSDesignerSettings.DTSDesigner_Default**

Show multi-phase pump in DTS designer

.DesignerSettings

The value of the **Enable Save to Meta Data Services** property cannot be set by using the DTS object model. This property can be set by using the **Registry** class in the **Microsoft.Win32** namespace of the .NET Framework Class Library, as demonstrated in the following sample.

Sample Code

The following code sample for a console application, when compiled and run, enables all the DTS application properties discussed in this topic. To disable the options instead, change the new values from **True** to **False**, and change the value of the **DesignerSettings** property to **DTSDesignerSettings.DTSDesigner_Default**.

This application requires a reference to the **Microsoft DTSPackage Object Library (COM)**.

The value of the **Enable Save to Meta Data Services** option can only be set by modifying the Registry directly.

```
Imports Microsoft.Win32
Imports DTS

Module SetDTSProperties

    Sub Main()

        Const SETTINGS_ROOT_KEY As String = "Software\Microsoft\Microsof
        Const METADATASERVICES_VALUE As String = "EnableSaveToRepository

        Dim dtsApp As New DTS.Application
        Dim keySettingsRoot As RegistryKey

        With dtsApp
            .TaskInfos.UseCache = True
            .TransformationInfos.UseCache = True
            .ScriptingLanguageInfos.UseCache = True
            .OLEDBProviderInfos.UseCache = True
            .DesignerSettings = DTSDesignerSettings.DTSDesigner_ShowMultiP
            .JITDebug = True
        End With

        keySettingsRoot = Registry.LocalMachine.OpenSubKey(SETTINGS_ROOT
        If keySettingsRoot Is Nothing Then
            keySettingsRoot = Registry.LocalMachine.CreateSubKey(SETTINGS_
        End If
        With keySettingsRoot
            .SetValue(METADATASERVICES_VALUE, Math.Abs(CType(True, Integer
            .Close()
```

```

        End With

    End Sub

End Module

using Microsoft.Win32;
using DTS;

class SetDTSProperties
{

    public static void Main()
    {

        const string SETTINGS_ROOT_KEY = "Software\\Microsoft\\Microsoft
        const string METADATASERVICES_VALUE = "EnableSaveToRepository";

        DTS.Application dtsApp = new DTS.Application();
        RegistryKey keySettingsRoot;

        {
            dtsApp.TaskInfos.UseCache = true;
            dtsApp.TransformationInfos.UseCache = true;
            dtsApp.ScriptingLanguageInfos.UseCache = true;
            dtsApp.OLEDBProviderInfos.UseCache = true;
            dtsApp.DesignerSettings = DTSDesignerSettings.DTSDesigner_Show
            dtsApp.JITDebug = true;
        }

        keySettingsRoot = Registry.LocalMachine.OpenSubKey(SETTINGS_ROOT
        if (keySettingsRoot==null)
        {
            keySettingsRoot = Registry.LocalMachine.CreateSubKey(SETTINGS_
        }
        {
            keySettingsRoot.SetValue(METADATASERVICES_VALUE, Math.Abs((int
            keySettingsRoot.Close();
        }

    }

}

```

The following subroutine can be added to an application to refresh the cache

when appropriate. Cached values are stored in the Registry under **HKEY_CURRENT_USER\Software\Microsoft\Microsoft SQL Server\80\DTS\Enumeration.**

 [Copy Code](#)

```
Private Sub RefreshCache()  
  
    Me.Cursor = Cursors.WaitCursor  
    With dtsApp  
        .TaskInfos.Refresh()  
        .TransformationInfos.Refresh()  
        .ScriptingLanguageInfos.Refresh()  
        .OLEDBProviderInfos.Refresh()  
    End With  
    Me.Cursor = Cursors.Default  
  
End Sub
```

 [Copy Code](#)

```
private void RefreshCache()  
{  
  
    this.Cursor = Cursors.WaitCursor;  
    {  
        dtsApp.TaskInfos.Refresh();  
        dtsApp.TransformationInfos.Refresh();  
        dtsApp.ScriptingLanguageInfos.Refresh();  
        dtsApp.OLEDBProviderInfos.Refresh();  
    }  
    this.Cursor = Cursors.Default;  
  
}
```

▣ **See Also**

Concepts

[SQL Server 2005 Integration Services Backward Compatibility
Upgrading or Migrating Data Transformation Services](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Notification Services Backward Compatibility



[See Also](#)



[Upgrading to SQL Server 2005](#) > [Backward Compatibility](#) >

The following sections list changes in SQL Server 2005 Notification Services that could affect your existing Notification Services applications.

▣ **Deprecated Features**

The following classes in the Notification Services API, and their associated explicit interface definitions, have been deprecated and replaced with Notification Services Management Object (NMO) classes.

Deprecated class	Replacement NMO class
ApplicationEnumeration	ApplicationCollection
DeliveryChannel	DeliveryChannel
DeliveryChannelEnumeration	DeliveryChannelCollection
EventClass	EventClass
EventClassEnumeration	EventClassCollection
InstanceEnumeration	InstanceCollection
NotificationClass	NotificationClass
NotificationClassEnumeration	NotificationClassCollection
NSInstanceDescription	Instance
Protocol	NotificationClassProtocol
ProtocolEnumeration	NotificationClassProtocolCollection
SubscriptionClass	SubscriptionClass
SubscriptionClassEnumeration	SubscriptionClassCollection

▣ **Discontinued Functionality**

No Notification Services 2.0 functionality has been discontinued in SQL Server 2005.

□ Breaking Changes

SQL Server 2005 Notification Services contains some changes that will cause your existing Notification Services 2.0 applications to stop working when you upgrade to SQL Server 2005. The table below contains the list of known breaking changes.

Breaking change	Description
The Notify() function has been removed.	In Notification Services 2.0, you had to use the Notify() function in any subscription rule that generated notifications. For security and performance reasons, the Notify() has been removed. For more information, see Migrating Notification Services 2.0 Instances to SQL Server 2005 .
Custom components must be recompiled.	In SQL Server 2005, the strong name of the Notification Services assembly has changed. As a result, you must recompile any custom event providers, content formatters, delivery protocols, and custom applications built using the Notification Services API. When you recompile these components or applications, use the SQL Server 2005 assemblies and .NET Framework version 2.0.
Notification Services instances must use the same version of the Database Engine.	Notification Services 2.0 instances must use the SQL Server 2000 Database Engine. SQL Server 2005 Notification Services instances must use the SQL Server 2005 Database Engine. After you upgrade the Database Engine to SQL Server 2005, you must migrate the instance of Notification Services to SQL Server 2005. For more information, see Migrating Notification Services 2.0 Instances to SQL Server 2005 .

▣ **Behavior Changes**

There are no significant behavior changes for SQL Server 2005 Notification Services.

▣ **See Also**

Concepts

[Migrating Notification Services Instances](#)

[Installing SQL Server Notification Services](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Replication Backward Compatibility

 [See Also](#)



[Upgrading to SQL Server 2005](#) > [Backward Compatibility](#) >

Topics in the backward compatibility section describe changes in behavior between versions of Microsoft SQL Server replication. It is important to understand backward compatibility if you are upgrading or if you have more than one version of SQL Server in a replication topology.

Topic	Description
Deprecated Features in SQL Server 2005 Replication	Replication features that have been retained in Microsoft SQL Server 2005 for backward compatibility, but which will be removed in a future version of SQL Server.
Discontinued Functionality in SQL Server 2005 Replication	Replication features that existed in Microsoft SQL Server 2000 but have been removed from SQL Server 2005.
Behavior Changes in SQL Server 2005 Replication	Replication features that have changed in SQL Server 2005.
Breaking Changes in SQL Server 2005 Replication	Replication feature changes that might require changes to applications.
Using Multiple Versions of SQL	Considerations for using more than one version of SQL

[Server in a
Replication Topology](#)

Server in a replication topology.

▣ **See Also**

Concepts

[Upgrading Replicated Databases](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Deprecated Features in SQL Server 2005 Replication



[See Also](#)



[Upgrading to SQL Server 2005](#) > [Backward Compatibility](#) > [Replication Backward Compatibility](#) >

Updated: **14 April 2006**

This topic describes replication features that will be removed in a future release. The features are available in Microsoft SQL Server 2005, but we recommend that you use alternative features when possible.

Note:

This topic is available in the Setup Help documentation and in SQL Server 2005 Books Online. Topic links that appear as bold text in the Setup Help documentation refer to topics that are only available in Books Online.

▣ **Deprecated Features That Affect All Types of Replication**

The following features, which apply to all types of replication, have been deprecated in SQL Server 2005.

Feature	Description
Attachable subscriptions	<p>This feature can be used when deploying a large number of pull subscriptions, which is common in merge replication. In SQL Server 2005, we recommend that you use the following approaches rather than attachable subscriptions:</p> <ul style="list-style-type: none">• For merge publications that are partitioned using parameterized filters, we recommend that you use the new features of partitioned snapshots, which simplify the initialization of a large number of subscriptions. For more information, see Snapshots for Merge Publications with Parameterized Filters.• For publications that are not partitioned, you can initialize a subscription with a backup. For more information, see Initializing a Merge Subscription Without a Snapshot and Initializing a Transactional Subscription Without a Snapshot. <p>In both cases, you can automate the creation of a large number of subscriptions with scripting: create a single subscription; script it; modify the script for each Subscriber; and apply the script at each Subscriber to create a subscription. For more information, see Scripting Replication.</p> <p>For more information about attachable subscriptions, see Attachable Subscriptions.</p>
	<p>Specifying a value of anonymous for the @subscriber_type parameter of sp_addmergepullsubscription (Transact-SQL) or the @subscription_type parameter of</p>

Anonymous subscriptions	<p>sp_addpullsubscription (Transact-SQL) is deprecated.</p> <p>In SQL Server 2005, a value of anonymous is required if you want to create a subscription without registering the subscription at the Publisher. This is necessary for cases in which you cannot establish a SQL Server connection to the Publisher during subscription configuration. In a future release of SQL Server, the anonymous setting will no longer be required for this situation.</p>
Subscriber registration	<p>The stored procedure sp_addsubscriber (Transact-SQL) is deprecated. It is no longer required to explicitly register a Subscriber at the Publisher.</p>
SQL Distributed Management Objects (SQL-DMO)	<p>Existing code will continue to work, but SQL-DMO does not support new features in SQL Server 2005. Use Replication Management Objects (RMO) instead. For more information, see Programming with Replication Management Objects.</p>
Schema changes using sp_repladdcolumn and sp_repldropcolumn	<p>The stored procedures sp_repladdcolumn (Transact-SQL) and sp_repldropcolumn (Transact-SQL) have been deprecated. Use schema change replication instead. For more information, see Schema Changes on Publication Databases.</p> <p>The stored procedures cannot be used for adding or dropping columns with data types introduced in SQL Server 2005: XML, varchar(max), nvarchar(max), varbinary(max), or user defined types (UDT).</p>
Checksum validation	<p>Checksum validation should only be used for Subscribers running Microsoft SQL Server version 7.0. Use binary checksum validation for SQL Server Subscribers running more recent versions. You can also use row count validation for all SQL Server Subscribers, including version 7.0 and those subscribing to publications from Oracle Publishers. For more information, see Validating Replicated Data.</p>
Adding publications	<p>Adding a publication to Active Directory using the @add_to_active_directory parameter of sp_addpublication (Transact-SQL) or</p>

to Active Directory	sp_addmergepublication (Transact-SQL) , has been deprecated. Subscribing to a publication by locating it in Active Directory has been discontinued.
-UseInprocLoader parameter	This parameter of the Distribution Agent and Merge Agent is deprecated because it is not compatible with the XML data type. If you are not replicating XML data, this parameter can be used. For more information, see Replication Distribution Agent and Replication Merge Agent .

▣ **Deprecated Features of Transactional Replication**

The following transactional replication features have been deprecated for SQL Server 2005.

Feature	Description
Subscription expiration for transactional publications	The @retention property of sp_addpublication (Transact-SQL) has been deprecated. Subscriptions are still marked as inactive and must be reinitialized if they have not synchronized within the maximum distribution retention period (the @max_distretention property of sp_adddistributiondb (Transact-SQL)). For more information about retention periods, see Subscription Expiration and Deactivation .
"No sync" subscriptions to transactional publications	A subscription is a "no sync" subscription if a value of none is specified for the @sync_type parameter of sp_addsubscription (Transact-SQL) parameter. If you want to specify that the necessary schema and data are already present in the subscription database, specify a value of replication support only for the parameter instead. For more information, see Initializing a Transactional Subscription Without a Snapshot .
ODBC Subscribers	Use OLE-DB for non-SQL Server Subscribers instead. For information about supported Subscribers, see Non-SQL Server Subscribers .
Transformable subscriptions	This feature is available through the stored procedure interface, but support for this feature in the user interface has been dropped. Using the feature requires installation of Microsoft SQL Server 2000 Data Transformation Services (DTS). For more information, see SQL Server 2005 Integration Services Backward Compatibility . For more information about transformable subscriptions, see Transforming Published Data .
Updatable subscriptions with snapshot publications	Using updatable subscriptions with transactional publications is still supported. For more information, see Updatable Subscriptions for Transactional Replication .
Distribution ActiveX	This control allows you to embed the Distribution Agent in applications. Use RMO instead. For more information, see

control	Synchronizing Subscriptions (RMO Programming).
Replication Distributor Interface	<p>In SQL Server 2000, the Replication Distributor Interface provided an interface to store replicated transactions in the distribution database on the Distributor. This interface could be used to enable publishing from a non-SQL Server database (additional custom programming was required to track changes on the Publisher). Support for this feature has been deprecated, but existing code will continue to work on a server upgraded from SQL Server 2000. For more information, see "Replication Distributor Interface Reference" in SQL Server 2000 Books Online.</p> <p>SQL Server 2005 supports publishing from Oracle databases without custom programming. For more information, see Oracle Publishing Overview.</p>

▣ **Deprecated Features of Merge Replication**

The following merge replication features have been deprecated for SQL Server 2005.

Feature	Description
Alternate synchronization partners	<p>The alternate synchronization partners feature allows you to specify an alternate Publisher with which a Subscriber can synchronize. For more information, see Alternate Synchronization Partners.</p> <p>In SQL Server 2005, we recommend that you use merge replication in conjunction with database mirroring, rather than alternate synchronization partners. For more information, see Replication and Database Mirroring.</p>
Merge ActiveX control	<p>This control allows you to embed the Merge Agent in applications. Use RMO instead. For more information, see Synchronizing Subscriptions (RMO Programming).</p>
Multicolumn UPDATE option	<p>When merge replication performs an update, it updates all changed columns in one UPDATE statement and resets unchanged columns to their original values. Alternatively, it can issue multiple UPDATE statements, with one UPDATE statement for each column that has changed. The multicolumn UPDATE statement is typically more efficient.</p> <p>In previous versions of SQL Server, it was recommended to specify a value of false for the fast_multicol_updateproc article option to address cases in which a multicolumn update (one UPDATE statement) might be less efficient:</p> <ul style="list-style-type: none">• Most updates involve a small number of columns.• Index maintenance on unchanged columns is high because those columns are reset when updates occur. <p>Due to performance improvements in SQL Server, this option is no longer required for these cases.</p>

▣ **See Also**

Concepts

[Replication Backward Compatibility](#)

Other Resources

Replication Enhancements

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Discontinued Functionality in SQL Server 2005 Replication



[See Also](#)



[Upgrading to SQL Server 2005](#) > [Backward Compatibility](#) > [Replication Backward Compatibility](#) >

Updated: **14 April 2006**

This topic describes replication features that have been discontinued in Microsoft SQL Server 2005.

 **Note:**

This topic is available in the Setup Help documentation and in SQL Server 2005 Books Online. Topic links that appear as bold text in the Setup Help documentation refer to topics that are only available in Books Online.

▣ Discontinued Features That Affect All Types of Replication

The following replication features have been discontinued in SQL Server 2005.

Feature	Description
Creating push subscriptions without a connection to the Subscriber in the New Subscription Wizard	Creating push subscriptions in the New Subscription Wizard now requires an active connection to the Subscriber during configuration. For more information about using the wizard, see How to: Create a Push Subscription (SQL Server Management Studio) .
Using file transfer protocol (FTP) to initialize Subscribers running SQL Server version 7.0	This feature is available for Subscribers running SQL Server 2000 or later versions. For more information, see Transferring Snapshots Through FTP .
Creating subscriptions in Windows Synchronization Manager	You can no longer create subscriptions in Synchronization Manager, but you can still synchronize subscriptions. For more information, see Windows Synchronization Manager .
Subscribing to a publication by locating it in Active Directory	Adding a publication to Active Directory using the @add_to_active_directory parameter of sp_addpublication (Transact-SQL) or sp_addmergepublication (Transact-SQL) , has been deprecated. Subscribing to a publication by locating it in Active Directory has been discontinued.
Snapshot ActiveX control	This control allows you to embed the Snapshot Agent in applications. Use the new managed Snapshot Agent instead. For more information, see SnapshotGenerationAgent and How to: Create the Initial Snapshot (RMO Programming) .

Remote agent activation	SQL Server 2000 supported running the Distribution Agent or Merge Agent on another computer and then activating that agent remotely using DCOM. Support has been discontinued for Distributors running SQL Server 2005 and later; it has been deprecated for Distributors running earlier versions of SQL Server. For more information, see "Remote Agent Activation" in SQL Server 2000 Books Online.
Microsoft Access (Jet 4.0) Subscribers	Jet is the underlying database used by Access, and replication supported subscriptions to Jet databases in SQL Server 2000. These subscriptions are no longer supported. We recommend using Microsoft SQL Server 2005 Express Edition instead. Access can use a SQL Server database as a backend, and SQL Server databases are not affected by this issue. For more information, see Replicating Data to SQL Server Express .

▣ Discontinued Transactional Replication Features

The following transactional replication features have been discontinued in SQL Server 2005.

Feature	Description
Microsoft Message Queuing (MSMQ) option for queued updating subscriptions	With queued updating subscriptions, changes from Subscribers are written to a queue; changes are then read from the queue and delivered to the Publisher by the Queue Reader Agent. In SQL Server 2000, subscriptions could use a SQL Server queue or MSMQ to queue changes. The type of queue was specified with the @queue_type parameter of sp_addpublication (Transact-SQL) , which allowed values of sql and msmq . In SQL Server 2005, only a value of sql is allowed. Existing publications that use MSMQ are modified during upgrade to use a SQL Server queue. If you have applications that depend on queued updating using MSMQ, these applications will need to be rewritten to accommodate a SQL Server queue. For more information about queued updating subscriptions, see Updatable Subscriptions for Transactional Replication .

□ Discontinued Merge Replication Features

The following merge replication features have been discontinued in SQL Server 2005.

Feature	Description
Publishing from SQL Server 2005 Express Edition	SQL Server MSDE could serve as a Publisher for merge publications. SQL Server Express, the replacement for MSDE, cannot serve as a Publisher. It can subscribe to merge, transactional, and snapshot publications. Merge replication, and transactional replication with updating subscriptions, both allow changes to be propagated from Subscribers back to the Publisher. For more information about replicating to SQL Server Express, see Replicating Data to SQL Server Express .
IVBCustomResolver interface	This interface allowed you to write COM-based custom conflict resolvers in Visual Basic. In SQL Server 2005, we recommend using Business Logic Handlers rather than COM-based resolvers. For more information, see Executing Business Logic During Merge Synchronization . You can still write COM-based resolvers in C++. For more information, see COM-Based Custom Resolvers .

▣ **See Also**

Concepts

[Replication Backward Compatibility](#)

Other Resources

Replication Enhancements

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Breaking Changes in SQL Server 2005 Replication



[See Also](#)



[Upgrading to SQL Server 2005](#) > [Backward Compatibility](#) > [Replication Backward Compatibility](#) >

Updated: **14 April 2006**

This topic describes replication feature changes that might require changes to applications.

 **Note:**

This topic is available in the Setup Help documentation and in SQL Server 2005 Books Online. Topic links that appear as bold text in the Setup Help documentation refer to topics that are only available in Books Online.

▣ Breaking Changes That Affect All Types of Replication

The following breaking changes apply to all types of replication in Microsoft SQL Server 2005.

Feature	Description
Changes required for replication scripts	<p>The replication agent security model has changed from Microsoft SQL Server 2000. For detailed information about the security model, see Replication Agent Security Model. If you are a member of the sysadmin fixed server role in SQL Server 2005 and you run replication scripts created from SQL Server 2000 or SQL Server 7.0, the scripts execute properly. If you are a member of the dbo fixed database role or another role, the scripts fail and must be upgraded. For information about upgrading scripts, see How to: Upgrade Replication Scripts (Replication Transact-SQL Programming). Although it is not required to upgrade scripts that are executed by members of the sysadmin role, it is recommended in order to take advantage of security enhancements.</p>
Local connections for replication agents	<p>On upgrade to SQL Server 2005, any local connections that use SQL Server Authentication are modified to use Windows Authentication. Local connections are those connections made by an agent to an instance of SQL Server running on the same computer as the agent. For example, the Merge Agent for a pull subscription runs at the Subscriber, so the connections it makes to the Subscriber are local connections.</p> <p>In previous versions of SQL Server, agents ran, by default, under the context of the SQL Server Agent service account. After upgrade, local connections are made under the context of this account. SQL Server 2005 allows fine-grained control over each account under which the replication agents run and make Windows Integrated connections to</p>

	<p>databases and other resources; a different account can be specified for each agent. After upgrade, it is recommended to specify different accounts for each agent. For more information, see Upgrading Replicated Databases and Replication Agent Security Model.</p>
ActiveX controls	<p>All ActiveX controls are marked as unsafe for scripting and initialization.</p> <p>The Snapshot Agent ActiveX control is not available in SQL Server 2005. Use the new managed Snapshot Agent instead. For more information, see SnapshotGenerationAgent and How to: Create the Initial Snapshot (RMO Programming).</p>
Password for the distributor_admin account	<p>Trusted connections between a Publisher and a remote Distributor are no longer supported because they did not require a password (trusted connections were used by default in versions prior to SQL Server 2000 Service Pack 3). If you use a remote Distributor, before upgrading to SQL Server 2005, convert trusted connections to non-trusted connections (this issue does not affect Publishers that use a local Distributor). For more information about the distributor_admin account, see Securing the Distributor.</p> <p>To determine the type of connection being used</p> <ul style="list-style-type: none"> • Execute sp_helpdistpublisher (Transact-SQL) at the Distributor. If the value in the trusted column is 1, you must change to a non-trusted connection. <p>To change to a non-trusted connection</p> <ol style="list-style-type: none"> 1. Execute sp_changedistpublisher (Transact-SQL) at the Distributor, specifying a value of 'trusted' for the parameter @property and a value of 'False' for the parameter @value. <p> Note: Some versions of the SQL Server 2000 Books Online do not list 'trusted' as a valid value for @property. It is valid for all SQL Server 2000 releases.</p>

	<p>2. Execute sp_changedistributor_password (Transact-SQL) at both the Publisher and the Distributor, specifying a strong password for the parameter @password.</p>
<p>SQL Server Express does not include SQL Server Agent</p>	<p>If you are upgrading to SQL Server Express Edition (SQL Server Express), you must reconfigure replication synchronization because SQL Server Express does not include SQL Server Agent.</p> <p>If you want to use pull subscriptions, you must synchronize them using Replication Management Objects (RMO), Windows Synchronization Manager, or by running the replication agent at the command line. For more information, see Replicating Data to SQL Server Express.</p> <p>If you want to continue to use SQL Server Agent to run replication agent jobs, you must use push subscriptions or upgrade to a different version of SQL Server (all versions except SQL Server Express and Microsoft SQL Server Compact Edition include SQL Server Agent). With push subscriptions, the Distribution Agent or Merge Agent runs at the Distributor, so SQL Server Agent is available (SQL Server Express cannot be a Distributor).</p>
<p>Microsoft Access (Jet 4.0) Subscribers</p>	<p>Jet is the underlying database used by Access, and replication supported subscriptions to Jet databases in SQL Server 2000. These subscriptions are no longer supported.</p> <p>It is recommended to use Microsoft SQL Server 2005 Express Edition instead. Access can use a SQL Server database as a backend, and SQL Server databases are not affected by this issue. For more information, see Replicating Data to SQL Server Express.</p>

▣ Breaking Changes for Transactional Replication

The following breaking changes apply to transactional replication in SQL Server 2005.

Feature	Description
Microsoft Message Queuing (MSMQ) option for queued updating subscription	<p>With queued updating subscriptions, changes from Subscribers are written to a queue; changes are then read from the queue and delivered to the Publisher by the Queue Reader Agent. In SQL Server 2000, subscriptions could use a SQL Server queue or MSMQ to queue changes. The type of queue was specified with the @queue_type parameter of sp_addpublication (Transact-SQL), which allowed values of sql and msmq. In SQL Server 2005, only a value of sql is allowed. Existing publications that use MSMQ are modified during upgrade to use a SQL Server queue. If you have applications that depend on queued updating using MSMQ, these applications will need to be rewritten to accommodate a SQL Server queue. For more information on queued updating subscriptions, see Updatable Subscriptions for Transactional Replication.</p> <p>Upgrade will remove the existing MSMQ subscription queues if the MSMQ service is running while SQL Server is being upgraded.</p> <p> Important: In Windows 2000 and Windows XP, the Microsoft Distributed Transaction Coordinator (MSDTC) service must also be running, because MSMQ requires MSDTC on those platforms.</p> <p>If the MSMQ service is not running, remove the queues manually after upgrade is complete. For more information about removing queues, see the Windows documentation.</p>

▣ Breaking Changes for Merge Replication

The following breaking changes apply to merge replication in SQL Server 2005.

Feature	Description
Publishing from SQL Server 2005 Express Edition	SQL Server MSDE could serve as a Publisher for merge publications. SQL Server Express, the replacement for MSDE, cannot server as a Publisher. It can subscribe to merge, transactional, and snapshot publications. Merge replication and transactional replication with updating subscriptions both allow changes to be propagated from Subscribers back to the Publisher. For more information about replicating to SQL Server Express, see Replicating Data to SQL Server Express .
Batching of changes	In previous versions of SQL Server, changes made by the Merge Agent were performed on a row-by-row basis. In SQL Server 2005, changes are batched to improve performance; therefore, more than one row can be inserted, updated, or deleted within a single statement. If any published tables in the publication or subscription databases have triggers, ensure that the triggers can handle multi-row inserts, updates, and deletes. For more information, see Multirow Considerations for DML Triggers .
Re-creation of conflict tables	On upgrade to SQL Server 2005, conflict tables are re-created with DBO as their owner. If any of the tables were owned by other users in SQL Server 2000, your application might need to be modified. Merge replication creates a conflict table for each article in a publication, with a name in the form conflict_PublicationName_ArticleName . All metadata tables are re-created on upgrade, and all conflict tables are created in the DBO schema.
New identity ranges assigned	For tables that use automatic identity range management, replication might assign new identity ranges during upgrade. If any tables have a larger identity range assigned to the Subscriber than to the Publisher, replication assigns a range to the Publisher equal to that of the Subscriber. To determine ranges being used for each article, execute

sp_helpmergearticle (Transact-SQL) in the publication database and check the **pub_identity_range** and **identity_range** columns.

▣ **See Also**

Concepts

[Replication Backward Compatibility](#)

Other Resources

Replication Enhancements

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Behavior Changes in SQL Server 2005 Replication



[See Also](#)



[Upgrading to SQL Server 2005](#) > [Backward Compatibility](#) > [Replication Backward Compatibility](#) >

Updated: **14 April 2006**

This topic describes behavior changes in replication features that existed in Microsoft SQL Server 2000. For information about replication features enhancements, see **Replication Enhancements**.

Note:

This topic is available in the Setup Help documentation and in SQL Server 2005 Books Online. Topic links that appear as bold text in the Setup Help documentation refer to topics that are only available in Books Online.

▣ Behavior Changes That Affect All Types of Replication

The following changes affect all types of replication.

Feature	Description
Replication agent security model	In previous versions of SQL Server, agents ran, by default, under the context of the SQL Server Agent service account. Microsoft SQL Server 2005 allows fine-grained control over each account under which the replication agents run and make Microsoft Windows Integrated connections to databases and other resources; a different account can be specified for each agent. For more information, see Security Considerations for Replication and Replication Agent Security Model . For information about how this change affects upgrade, see the "New Replication Agent Security Model" section of Upgrading Replicated Databases and Breaking Changes in SQL Server 2005 Replication .
Windows Synchronization Manager	In previous versions of SQL Server, the ability to synchronize subscriptions with Synchronization Manager was enabled by default. In SQL Server 2005, you must explicitly enable this option if you want to use Synchronization Manager. For more information, see How to: Synchronize a Subscription Using Windows Synchronization Manager (Windows Synchronization Manager) .
Replication Conflict Viewer	In SQL Server 2000, the Replication Conflict Viewer was packaged for redistribution. In SQL Server 2005, the viewer is not packaged separately. To include the Replication Conflict Viewer in an application, you must install Microsoft .NET Framework 2.0 on the computer on which the application is deployed, and copy a number of files to the computer. For more information, see "Other Replication Upgrade Issues" in Upgrade Advisor Help. For more information about Upgrade Advisor, see Using Upgrade Advisor to Prepare for Upgrades .
	Schema options allow you to specify how attributes and objects associated with tables, such as indexes and constraints, are

Schema option changes	replicated. The behavior of a number of schema options has changed in SQL Server 2005. The next section of this topic provides more information.
-----------------------	--

Schema Options Behavior Changes

The following table summarizes schema option changes in SQL Server 2005.

 **Note:**

If the **0x8000** schema option was set in SQL Server 2000, it is disabled during upgrade to SQL Server 2005. For schema options **0x10** or **0x40**, replication might create a larger number of indexes in SQL Server 2005 than in SQL Server 2000.

Option	Behavior if the option is set in SQL Server 2000	Behavior if the option is set in SQL Server 2005
0x80	Creates a constraint or index. If option 0x8000 is also enabled, the primary key is created as a constraint with an index. If option 0x8000 is not enabled, only the index on the primary key column is created.	Creates a primary key constraint at the Subscriber. Any indexes related to the constraint are also replicated, even if options 0x10 and 0x40 are not enabled (these options control index creation for other cases).
0x4000	Creates a constraint or index. If option 0x8000 is also enabled, the unique constraint is created as a constraint with an index. If option 0x8000 is not enabled, only the index on the column is created.	Creates any unique constraints at the Subscriber. Any indexes related to the constraint are also replicated, even if options 0x10 and 0x40 are not enabled (these options control index creation for other cases).
0x8000	Creates primary key constraints and unique constraints if options 0x80 or 0x4000 are also specified. If neither of these options is specified, option 0x8000 has no effect.	The option has no effect.

▣ Behavior Changes for Transactional Replication

The following changes affect transactional replication.

Feature	Description
Subscriber object ownership	<p>When using the SQL Server 2005 New Publication Wizard to create a publication, the owner of objects created at the Subscriber defaults to the value of the owner of the corresponding object on the Publisher. In previous releases, the owner was not specified during the creation of the object on the Subscriber; it defaulted to the owner associated with the Distribution Agent account used to connect to the Subscriber. The behavior is unchanged for the stored procedure sp_addarticle (Transact-SQL).</p>
Updatable subscriptions security mode	<p>The @security_mode parameter of sp_link_publication governs how the triggers for immediate updating subscriptions execute calls at the Publisher. In SQL Server 2005, the options for this parameter are:</p> <ul style="list-style-type: none">• 0: Uses SQL Server Authentication with the login specified in the stored procedure as login and password.• 1: Uses the security context (SQL Server Authentication or Windows Integrated Authentication) of the user making the change at the Subscriber.• 2: Uses an existing, user-defined linked server login. <p>In previous versions of SQL Server, option 0 was used to specify a dynamic remote procedure call (RPC) from the Subscriber to the Publisher rather than a linked server.</p>

▣ Behavior Changes for Merge Replication

The following changes affect merge replication.

Feature	Description
Publication compatibility level	<p>In previous versions of SQL Server, the compatibility was automatically increased if you enabled a feature required a higher level. In SQL Server 2005, you must manually set the compatibility level to 90RTM before enabling functionality that requires that compatibility. For more information, see the "Compatibility Level for Merge Publications" section of Using Multiple Versions of SQL Server in a Replication Topology.</p>
Compensating actions	<p>In previous versions of SQL Server, compensating actions were taken if errors (such as constraint violations) were encountered during synchronization. In some cases this behavior is desirable, but in some cases it can be problematic; for example, one incorrectly configured Subscriber that generates an error can cause changes undone at the Publisher and all other Subscribers.</p> <p>In SQL Server 2005, the @compensate_for_errors parameter of sp_addmergearticle controls whether compensating actions are taken. When set to False (the default), compensating actions are disabled; however errors are still logged and subsequent merges will continue to attempt to apply the changes. Although data in the affected rows might appear to be out of convergence, soon as you address the error, the change can be applied and data will converge. When set to True, a change that cannot be applied at a node during synchronization leads to compensating actions that undo the change at all other nodes.</p> <p> Note: If the source table for an article is already published in another publication, the value of</p>

	<p>@compensate_for_errors must be the same for both articles. For pull subscriptions at Publishers running SQL Server 2000 version 8.00.858 and earlier (which includes Service Pack 3), compensating actions will occur even with @compensate_for_errors set to False.</p>
<p>Conflict tables</p>	<p>In previous versions of SQL Server, merge replication created a single conflict table for each table article in a publication, with a name in the form conflict_<ArticleName>. In SQL Server 2005, the information is contained in two tables: MSmerge_conflicts_info and a table with a name in the form MSmerge_conflict_<PublicationName>_<ArticleName>.</p>
<p>Retention-based metadata cleanup</p>	<p>SQL Server 2005 uses retention-based metadata cleanup, which was introduced in SQL Server 2000 Service Pack 3. Metadata is periodically deleted from the following tables:</p> <ul style="list-style-type: none"> • MSmerge_contents • MSmerge_tombstone • MSmerge_genhistory • Any before image tables, if present. Before image tables are present if the @keep_partition_changes synchronization optimization option is enabled on the publication (see the next section for more information about this option).
<p>The @keep_partition_changes parameter</p>	<p>The @keep_partition_changes parameter was set to True by default in previous versions of SQL Server, because this results in more data being stored at the Publisher. It is set to True if the publication compatibility level is 90 or higher, and the @use_partition_groups parameter is set to False. For more information about these options, see Parameterized Row Filters.</p>

▣ Change History

Release	History
14 April 2006	New content: <ul style="list-style-type: none">• Added information about behavior changes for schema options.

▣ **See Also**

Concepts

[Replication Backward Compatibility](#)

Other Resources

Replication Enhancements

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Using Multiple Versions of SQL Server in a Replication Topology



[See Also](#)



[Upgrading to SQL Server 2005](#) > [Backward Compatibility](#) > [Replication Backward Compatibility](#) >

Updated: **14 April 2006**

Replication supports replicating data to different versions of Microsoft SQL Server. This topic covers:

- SQL Server versions supported
- Mapping SQL Server 2005 data types for previous versions
- Restoring a replicated database from a previous version
- Compatibility level for merge publications

For information about replicating data to Microsoft SQL Server 2005 Express Edition and Microsoft SQL Server 2005 Compact Edition, see **Replicating Data to SQL Server Express** and **Replicating Data to SQL Server Compact Edition**. For information about the features supported by each edition of SQL Server, see [Features Supported by the Editions of SQL Server 2005](#).

Note:

This topic is available in the Setup Help documentation and in SQL Server 2005 Books Online. Topic links that appear as bold text in the Setup Help documentation refer to topics that are only available in Books Online.

▣ SQL Server Versions Supported

SQL Server version 7.0 Service Pack 4 (SP4) is the earliest version that can participate in a replication topology with SQL Server 2005. If you use SQL Server 2000, SP3 or later is required.

When you replicate between or among different versions of SQL Server, you are often limited to the functionality of the earliest version used. For example, if you upgrade a Distributor to an instance of SQL Server 2005, but you have a Publisher running an instance of SQL Server 2000, and a Subscriber running an instance of SQL Server 7.0, you are limited to the general functionality and replication functionality of SQL Server 7.0.

For all types of replication, the Distributor version must be no earlier than the Publisher version (in many cases the Distributor is the same instance as the Publisher). You can use SQL Server 2005 as a remote Distributor for SQL Server 2005 and SQL Server 2000, but not for SQL Server 7.0. The following tables provide additional information about which versions of SQL Server can participate in the same topology. For more information about replication features supported in the various editions of SQL Server, see [Features Supported by the Editions of SQL Server 2005](#).

Transactional Replication and Snapshot Replication with Read-Only Subscribers

Distributor	SQL Server 7.0	SQL Server 2000	SQL Server 2005
Publisher	SQL Server 7.0	SQL Server 7.0	SQL Server 2000
		SQL Server 2000	SQL Server 2005
Subscribers	SQL Server 7.0	SQL Server 7.0	SQL Server 7.0
	SQL Server 2000	SQL Server 2000	SQL Server 2000
	SQL Server 2005	SQL Server 2005	SQL Server 2005

The above table shows that a read-only Subscriber to a transactional publication can be any version within two versions of the Publisher version. For example, a SQL Server 7.0 Publisher can have SQL Server 2005 Subscribers, and a SQL

Server 2005 Publisher can have SQL Server 7.0 Subscribers.

Transactional Replication with Updating Subscribers

Distributor	SQL Server 7.0	SQL Server 2000	SQL Server 2005
Publisher	SQL Server 7.0 ¹	SQL Server 7.0 ¹	SQL Server 2000 ²
		SQL Server 2000 ²	SQL Server 2005 ³
Subscribers	SQL Server 7.0	SQL Server 7.0	SQL Server 7.0
		SQL Server 2000	SQL Server 2000
		SQL Server 2005	SQL Server 2005

¹ For a SQL Server 7.0 Publisher, only SQL Server 7.0 Subscribers are supported.

² For a SQL Server 2000 Publisher, SQL Server 7.0, SQL Server 2000, and SQL Server 2005 Subscribers are supported.

³ For a SQL Server 2005 Publisher, SQL Server 2000 and SQL Server 2005 Subscribers are supported.

Merge Replication

Distributor	SQL Server 7.0	SQL Server 2000	SQL Server 2005
Publisher	SQL Server 7.0	SQL Server 7.0	SQL Server 2000
		SQL Server 2000	SQL Server 2005
Subscribers	SQL Server 7.0	SQL Server 7.0	SQL Server 7.0
		SQL Server 2000	SQL Server 2000
		SQL Server 2005	SQL Server 2005

The above table shows that a Subscriber to a merge publication can be any version no later than the Publisher version. For more information about compatibility for previous versions, see "Compatibility Level for Merge Publications" later in this topic.

SQL Server 7.0 and SQL Server Management Studio

SQL Server Management Studio can connect to instances running SQL Server 2000 or later. For Subscribers running SQL Server 7.0:

- Subscriptions and publications can be created with SQL Server 7.0 tools, SQL Server 2000 tools, SQL Distributed Management Objects (SQL-DMO), or stored procedures.
- Agents for pull subscriptions cannot be started from Management Studio or Replication Monitor. Agents can be specified to run on a schedule when the subscription is created, or they can be run on-demand from the command prompt.

Subscriptions for Subscribers running SQL Server 7.0 do appear in Management Studio and Replication Monitor after they are created. For information about creating subscriptions and publications, and running agents, see SQL Server 7.0 Books Online.

Using a SQL Server 2005 Distributor with a Publisher Running SQL Server 2000

SQL Server 2005 can be used as a remote Distributor for Publishers running SQL Server 2000. To change agent properties in this scenario, execute the following stored procedures at the Distributor. These procedures allow you to change properties that are new in SQL Server 2005:

- **sp_MSchange_snapshot_agent_properties (Transact-SQL)**
- **sp_MSchange_logreader_agent_properties (Transact-SQL)**
- **sp_MSchange_distribution_agent_properties (Transact-SQL)**
- **sp_MSchange_merge_agent_properties (Transact-SQL)**

If you have a Publisher and Distributor running SQL Server 2000, it is possible to change the credentials under which agents make connections using **sp_changedistpublisher (Transact-SQL)** and **sp_changesubscriber (Transact-SQL)**. However, if you upgrade the Distributor to SQL Server 2005, these procedures cannot be used to change credentials used in existing agent jobs (the procedures do affect agent jobs that are created after the procedure is called). In order to change the credentials for existing agent jobs, call one of the four procedures listed above.

▣ Mapping SQL Server 2005 Data Types for Previous Versions

SQL Server 2005 has introduced a number of new data types. These new data types are mapped to compatible data types at the Subscriber if push subscriptions from a SQL Server 2005 Distributor are used.

SQL Server 2005 data type	SQL Server 2000 or SQL Server 7.0 data type
XML	NTEXT
CLR User Defined Types (UDT)	IMAGE
VARCHAR(max)	TEXT
NVARCHAR(max)	NTEXT
VARBINARY(max)	IMAGE

You must verify that **VARCHAR(max)**, **NVARCHAR(max)**, **VARBINARY(max)**, **XML**, and **CLR UDT** types are mapped appropriately if they are replicated to Subscribers running earlier versions of SQL Server (this is done by default for articles in merge publications). Mapping behavior for these types is controlled by the schema options **0x20**, **0x10000000**, and **0x20000000** for **sp_addarticle (Transact-SQL)** and **sp_addmergearticle (Transact-SQL)**. For more information about setting schema options, see **How to: Specify Schema Options (SQL Server Management Studio)** and **How to: Specify Schema Options (Replication Transact-SQL Programming)**.

SQL Server 2000 introduced two data types that are mapped to compatible data types for SQL Server 7.0. These new data types are mapped to compatible data types at the Subscriber if push subscriptions from a SQL Server 2005 or SQL Server 2000 Distributor are used.

SQL Server 2005 or SQL Server 2000 data type	SQL Server 7.0 data type
SQL_VARIANT	IMAGE
BIGINT	DECIMAL

▣ **Restoring a Replicated Database from a Previous Version**

You can retain replication settings when restoring a backup of a replicated database from a previous version. If you restore the backups to a server and database with the same names as the server and database at which the backup was taken, or if you specify the **KEEP_REPLICATION** option, replication settings are preserved. For more information, see **RESTORE (Transact-SQL)**. After restoring the database, execute **sp_vupgrade_replication (Transact-SQL)** to upgrade schema and system data to support replication at the current product level.

Although preserving replication after restoring from backup from previous version is possible, it is rarely used as an upgrade option. It is more common to upgrade the replicated database as part of a product upgrade or to re-create the database and replication configuration from a set of scripts.

▣ **Compatibility Level for Merge Publications**

Merge replication uses the publication compatibility level to determine which features can be used by publications in a given database. The values range from 70RTM (SQL Server 7.0 with no service packs installed) to 90RTM. The compatibility level is specified:

- With the **@publication_compatibility_level** parameter of **sp_addmergepublication** (Transact-SQL). For more information, see **How to: Set the Publication Compatibility Level (Replication Transact-SQL Programming)**.
- On the **Subscriber Types** page of the New Publication Wizard. For more information about running this wizard, see **How to: Create a Publication and Define Articles (SQL Server Management Studio)**.
- On the **General** page of the **Publication Properties - <Publication>** dialog box. For more information, see **How to: Set the Publication Compatibility Level (SQL Server Management Studio)**.

The following features require a compatibility level of 90RTM or higher:

- Logical records. For more information, see **Grouping Changes to Related Rows with Logical Records**.
- Subscriber upload options. For more information, see **Optimizing Merge Replication Synchronization Performance with Download-Only Articles**.
- Nonoverlapping partitions. For more information, see **Parameterized Row Filters**.
- Business logic handlers. For more information, see **Executing Business Logic During Merge Synchronization**.
- Schema changes using ALTER <OBJECT> statements. For more information, see **Schema Changes on Publication Databases**.

The following features do not depend on the compatibility level, but they do require the Merge Agent that ships with SQL Server 2005; Subscribers running previous versions of SQL Server function as though the feature is not enabled:

- Precomputed partitions. For more information, see **Optimizing Parameterized Filter Performance with Precomputed Partitions**.

- Web synchronization. For more information, see **Web Synchronization for Merge Replication**.
- Parallel phase agents (specifying **-ParallelUploadDownload** for the Merge Agent). For more information, see **Replication Merge Agent**.
- New snapshot features for publications that use parameterized filters, which include the ability:
 - For a Subscriber to request a snapshot if one is not available for its partition.
 - For an administrator to pre-generate and schedule generation of snapshots.
 - To deliver parameterized snapshots using FTP.

For more information, see **Snapshots for Merge Publications with Parameterized Filters** and **Transferring Snapshots Through FTP**.

- Improved history logging and article-level statistics in Replication Monitor. For more information, see **How to: View Information and Perform Tasks for a Subscription (Replication Monitor)**.

Publication Compatibility Level Behavior in SQL Server 2005

It is important to understand the behavior of the publication compatibility level:

- The publication compatibility level is not connected to the database compatibility level.
- If you create a publication with **sp_addmergepublication (Transact-SQL)**, or through Replication Management Objects (RMO), the publication compatibility level is set to 80RTM by default. If you create a publication in the New Publication Wizard, the publication compatibility level is determined based on the options chosen on the **Subscriber Types** page of the wizard.
- In previous versions of SQL Server, the publication compatibility level automatically increased if you enabled a feature that required a higher level. In SQL Server 2005, you must manually set the publication compatibility level to 90RTM before enabling functionality that requires that compatibility level.
If you upgrade a Publisher from SQL Server 7.0, and then select one or more features that require a compatibility level of 80RTM, the compatibility level is automatically increased.
- The publication compatibility level can be decreased only if the Snapshot

- Agent has not been started and there are no subscriptions to the publication.
- All publications in the same database must have the same compatibility level. This requirement has the following consequences:
 - If a database contains a publication with a lower compatibility level (such as 80RTM), and you want to add another publication in the same database with a level of 90RTM, you must manually increase the level of the first publication before the new publication is added.
 - If a database contains two or more publications with lower compatibility levels, and you want to add another publication in the same database with a level of 90RTM, you must drop all but one of the existing publications; increase the level of the remaining publication to 90RTM; re-create the dropped publications with a level of 90RTM; and create the new publication with a level of 90RTM.

▣ Change History

Release	History
14 April 2006	<p data-bbox="321 424 613 462">Changed content:</p> <ul data-bbox="435 504 1409 808" style="list-style-type: none"><li data-bbox="435 504 1409 583">• Added tables to clarify which versions of SQL Server can be used in the same replication topology.<li data-bbox="435 592 1409 672">• Added information about schema options used to map data types.<li data-bbox="435 680 1409 808">• Removed information stating that SQL Server 7.0 can use an instance of SQL Server 2005 as a remote Distributor. This configuration is not supported.

▣ **See Also**

Concepts

[Replication Backward Compatibility](#)

[Upgrading Replicated Databases](#)

Other Resources

Replication Enhancements

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Full-Text Search Backward Compatibility



[Upgrading to SQL Server 2005](#) > [Backward Compatibility](#) >

Topics in the backward compatibility section describe changes in behavior between versions of Full-Text Search in Microsoft SQL Server 2005.

See...	For topics about...
Breaking Changes to Full-Text Search in SQL Server 2005	Full-Text Search changes that might require changes to applications.
Behavior Changes to Full-Text Search in SQL Server 2005	Other changes in behavior to Full-Text Search in this release.

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Breaking Changes to Full-Text Search in SQL Server 2005



[See Also](#)



[Upgrading to SQL Server 2005](#) > [Backward Compatibility](#) > [Full-Text Search Backward Compatibility](#) >

This topic describes Full-Text Search changes that might require changes to applications.

□ Breaking Changes That Affect Applications

The following breaking changes apply to Full-Text Search in Microsoft SQL Server 2005.

Feature	Description
Word breakers and filters	Word breakers and filters used by SQL Server Full-Text Search have been modified significantly to achieve improvements in functionality and reliability. In some specific cases, changes made to the word breakers have the potential to impact how some data is tokenized. This can result in tokens being created in SQL Server 2005 that are different from tokens created in SQL Server 2000. For example, in SQL Server 2005, compound word tokenizing for German has been improved, and now compounds with more than two component words can be tokenized. This could impact query results.
FULLTEXTCATALOG property	In SQL Server 2000, Itemcount returned the aggregate of the number of indexed rows in each full-text indexed table in the full-text catalog, plus one for every table in the catalog. In SQL Server 2005, this has been changed to reflect accurately the aggregate of the number of indexed rows in each full-text indexed table in the full-text catalog.
Length of full-text catalog names	In SQL Server 2005, the length of full-text catalog names is restricted to 120 characters, reduced from 128 in previous releases of SQL Server.
Microsoft Full-Text Engine for SQL Server service and third party filters	Microsoft Full-Text Engine for SQL Server (MSFTESQL) service does not load components that are not signed by Microsoft, by default. A third party filter, such as a PDF filter, that is currently installed on the server is not loaded by the MSFTESQL service by default after upgrade.
	The following Full-Text Search related properties and settings have been removed from SQL Server 2005.

Discontinued Full-Text Search properties	<ul style="list-style-type: none"> • DataTimeout • ConnectTimeout • Clean_up • LogSize
Multiple instances of SQL Server 2005 cannot share Microsoft Full-Text Search Engine for SQL Server (MSFTESQL) service.	On upgrade to SQL Server 2005, each SQL Server instance has a dedicated instance of the Microsoft Full-Text Search Engine for SQL Server (MSFTESQL) service.
Path of Full-Text Search catalogs	In SQL Server 2005, the path in the sysfulltextcatalogs view and the path returned by the sp_help_fulltext_catalogs and sp_help_fulltext_catalogs_cursor system stored procedures has changed. In SQL Server 2000, full-text catalogs existed in the root directory. In SQL Server 2005, they are found in the FTData directory.
Full-text indexes and non-deterministic and imprecise computed columns	SQL Server 2005 does not allow you to create full-text indexes on non-deterministic and imprecise computed columns. Such columns cannot be used as type columns, or as full-text key columns.
sysfulltextnotify	The sysfulltextnotify view is undocumented in SQL Server 2000, but it could still be queried. This view has been removed in SQL Server 2005.
Full-text indexes and system databases	In SQL Server 2000, full-text indexes were supported on the master , tempdb and model databases. SQL Server 2005 does not support this. Any full-text catalogs in these databases are removed during the upgrade to SQL Server 2005.
Global word breakers and filters	In SQL Server 2000, new word breakers and filters could only be added as global operating system level components. SQL Server 2005 allows the instance-level registration of new word breakers and filters. This provides functional and security isolation between instances.
Additional space	Because the document identifier (DOCID) map in SQL Server 2005 is stored in the database, the size

requirement for Full-
Text Search

requirement on the database side has increased.
However, the size requirement on the full-text catalog
side has decreased.

▣ **See Also**

Concepts

[Behavior Changes to Full-Text Search in SQL Server 2005](#)

Other Resources

[Full-Text Search Backward Compatibility](#)

Full-Text Search

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Behavior Changes to Full-Text Search in SQL Server 2005



[See Also](#)



[Upgrading to SQL Server 2005](#) > [Backward Compatibility](#) > [Full-Text Search Backward Compatibility](#) >

This topic describes the behavior changes in Full-Text Search features from previous releases of Microsoft SQL Server.

□ Behavior Changes

The following table contains the list of behavior changes.

Behavior change	Description
Master merge	<p>In SQL Server 2000, a master merge would start at midnight, or when 500,000 documents were full-text indexed.</p> <p>In SQL Server 2005, a master merge starts at the end of full population and also when an internal threshold on the number of full-text index files has been reached.</p> <p>A master merge also occurs when 500,000 documents are full-text indexed, which is the same as in SQL Server 2000.</p> <p>SQL Server 2005 also allows users to start a master merge using data definition language.</p>
MSSearch service	<p>In SQL Server 2000, Full-Text Search used the MSSearch service as the full-text indexing and search engine. In SQL Server 2005, this service has been renamed the MSFTESQL service. The MSFTESQL engine is built on MSSearch technology.</p>
Full-Text Search related stored procedures	<p>Because Full-Text Search related stored procedures have been rewritten in SQL Server 2005 to use the new full-text data definition language (DDL), some errors result in statement level errors.</p> <p>The following errors result in statement level errors. Other failures are batch level failures.</p> <ul style="list-style-type: none">• Permission check failure.• Explicit transaction check failure. (A full-text stored procedure cannot be called within explicit user transactions).• Full-text related action run in master, temp, or

	<p>model databases.</p> <ul style="list-style-type: none"> • Database is not read-only check failed. • Database is not full-text enabled. • Parameter validation: <ul style="list-style-type: none"> ◦ Catalog name not found. ◦ Table name does not exist, or table is not eligible for full-text indexing. ◦ Column name is not valid in the table. ◦ Parameters such as database name and table name are null.
Full-text catalog cleanup functionality	Running sp_fulltext_service cleanup in SQL Server 2005 is not possible.
Noise words	In SQL Server 2005, if a query that uses a Full-Text Search predicate or function contains a noise word and the transform noise words configuration option is not on, a warning is issued and no rows are returned. For more information about this option, see transform noise words Option .
Rows with rank zero returned	In SQL Server 2000, rows with rank zero were not returned. In SQL Server 2005, these rows are included in the result set. This could potentially affect the results of stored procedures and other queries.
Running sp_fulltext_database	<p>In SQL Server 2005, running sp_fulltext_database 'enable' on a user database that is already enabled for full-text indexing does nothing.</p> <p> Note: User databases are enabled for full-text indexing by default.</p>

▣ **See Also**

Concepts

[Breaking Changes to Full-Text Search in SQL Server 2005](#)

Other Resources

[Full-Text Search Backward Compatibility](#)

Full-Text Search

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Installing SQL Server 2005 Components



[See Also](#)



Updated: **5 December 2005**

Microsoft SQL Server 2005 can be installed through use of the Installation Wizard, or from the command prompt. The Installation Wizard provides a graphical user interface that guides you through each install-time decision, and is the preferred method for most users. The Installation Wizard provides guidance for initial setup of SQL Server 2005, including feature selection, instance naming rules, service account configuration, strong password guidelines, and scenarios for setting collations.

Command prompt installations are for advanced scenarios like silent installations; they can be run directly from the command prompt, or from command prompt syntax that references an installation file to specify installation options.

To ensure a successful installation experience, review the following requirements before you install SQL Server 2005:

- [Hardware and Software Requirements for Installing SQL Server 2005](#)
- [Check Parameters for the System Configuration Checker](#)
- [Security Considerations for a SQL Server Installation](#)

To install SQL Server 2005 components using the SQL Server 2005 Installation Wizard, see:

- [How to: Install SQL Server 2005 \(Setup\)](#)
- [How to: Create a New SQL Server 2005 Failover Cluster \(Setup\)](#)

To install SQL Server 2005 components from the command prompt, see:

- [How to: Install SQL Server 2005 from the Command Prompt](#)
- [Remote Setup Information for SQL Server 2005](#)

To upgrade from a previous SQL Server version, see:

- [Version and Edition Upgrades](#)
- [Upgrading to SQL Server 2005](#)

To add or remove SQL Server 2005 components using the SQL Server 2005 Installation Wizard, see:

- [How to: Change Components in an Existing Instance of SQL Server 2005 \(Setup\)](#)

▣ **See Also**

Concepts

[Preparing to Install SQL Server 2005](#)

Other Resources

[Database Engine Installation How-to Topics](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Installing the SQL Server Database Engine



[See Also](#)



[Installing SQL Server 2005 Components](#) >

Updated: 5 December 2005

The SQL Server Database Engine component of Microsoft SQL Server 2005 is the core service for storing, processing, and securing data. The Database Engine provides controlled access and rapid transaction processing to meet the requirements of the most demanding data consuming applications in your enterprise.

The Database Engine supports one or more server instances on a single computer. For more information about selecting features and components to install, see [Editions and Components of SQL Server 2005](#).

To create a typical SQL Server installation, see [How to: Install SQL Server 2005 \(Setup\)](#).



Important:

For local installations, you must run Setup as an administrator. If you install SQL Server from a remote share, you must use a domain account that has read and execute permissions on the remote share.

▣ **Step 1: Preparing to Install the Database Engine**

Before you install the SQL Server 2005 Database Engine, review the following requirements:

- [Hardware and Software Requirements for Installing SQL Server 2005](#)
- [Check Parameters for the System Configuration Checker](#)
- [Security Considerations for a SQL Server Installation](#)
- [Components of the SQL Server Database Engine](#)

▣ **Step 2: Installing the Database Engine**

After you review the installation requirements for SQL Server 2005 and ensure that your computer meets those requirements, you will be ready to install the Database Engine. Depending on your environment, you will install the Database Engine in one of the following ways:

- Install the Database Engine on a clean machine using the Installation Wizard. For more information, see [How to: Install SQL Server 2005 \(Setup\)](#) or [Before Installing Failover Clustering](#).
- Install the Database Engine on a clean computer from the command prompt. For more information, see [Remote Setup Information for SQL Server 2005](#). For example command prompt syntax, see [How to: Install SQL Server 2005 from the Command Prompt](#).
- Migrate or upgrade a previous SQL Server version to SQL Server 2005. For more information, see [Upgrading to SQL Server 2005](#) and [Working with Multiple Versions of SQL Server](#).

▣ **Step 3: Verifying your installation of the Database Engine**

Following installation, verify that the SQL Server 2005 services are running. For more information, see [How to: Verify a Successful Installation of SQL Server 2005 Services](#).

▣ **Step 4: Configuring your installation of the Database Engine**

Some features of SQL Server 2005 are turned off by default. To review configuration options, see [Configuring a SQL Server Installation](#) and [SQL Server Surface Area Configuration](#).

▣ **See Also**

Reference

[Editions and Components of SQL Server 2005](#)
[Collation Options and International Support](#)

Concepts

[Components of the SQL Server Database Engine](#)

Other Resources

[Database Engine Installation How-to Topics](#)
[Configuring High Availability](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Components of the SQL Server Database Engine



[See Also](#)



[Installing SQL Server 2005 Components](#) > [Installing the SQL Server Database Engine](#) >

Microsoft SQL Server 2005 supports up to 50 instances of the Database Engine on a single computer. The following features are installed when you select **SQL Server** Database Engine on the **Components to Install** page of the SQL Server 2005 Installation Wizard:

- Database Engine
- Replication
- Full-Text Search

The following additional features are options for many typical user scenarios:

- Integration Services
- Connectivity Components
- Programming Models
- Management Tools
- SQL Server Management Studio
- **AdventureWorks** sample databases and samples
- Books Online

▣ **See Also**

Reference

[Editions and Components of SQL Server 2005](#)

[Features Supported by the Editions of SQL Server 2005](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Remote Setup Information for SQL Server 2005



[See Also](#)



[Installing SQL Server 2005 Components](#) > [Installing the SQL Server Database Engine](#) >

Updated: 5 December 2005

There are two ways to install Microsoft SQL Server from the command prompt, which are described in the following topics.

 **Note:**

To install SQL Server 2005 to a remote computer, use a remote connection to run SQL Server 2005 Setup in user interface mode or from the command prompt.

▣ In This Section

Topic	Description
Running Setup from the Command Prompt	Lists properties and options for installing SQL Server from the command prompt, including unattended and silent installations. Also provides sample scripts.
Specifying Setup Properties with an .ini File	Provides information on use of the template .ini file on the SQL Server DVD to install SQL Server 2005.

▣ **See Also**

Tasks

[How to: Install SQL Server 2005 from the Command Prompt](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Running Setup from the Command Prompt



[See Also](#)



[Installing SQL Server 2005 Components](#) > [Installing the SQL Server Database Engine](#) > [Remote Setup Information for SQL Server 2005](#) >

Updated: 5 December 2005

The SQL Server 2005 Setup program provides a command prompt interface in addition to the graphical user interface. Use command prompt syntax to customize the way Setup installs SQL Server 2005. Command prompt installations can be specified at the command prompt or in an .ini file.

Note:

Remote Setup is not supported in this release. To install SQL Server 2005 on a remote computer, use a remote connection to run SQL Server Setup in user interface mode or from the command prompt. Record unattended (.iss) Setup is not supported in this release.

For examples that show syntax to install or remove an instance of SQL Server 2005, see [How to: Install SQL Server 2005 from the Command Prompt](#).

▣ **See Also**

Concepts

[How to: Rebuild the Registry for SQL Server 2005](#)

[How to: Rebuild the Master Database for SQL Server 2005](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Specifying Setup Properties with an .ini File



[See Also](#)



[Installing SQL Server 2005 Components](#) > [Installing the SQL Server Database Engine](#) > [Remote Setup Information for SQL Server 2005](#) >

Updated: 5 December 2005

The SQL Server 2005 Setup program provides a command prompt interface in addition to the graphical user interface. Use command prompt syntax to customize the way Setup installs SQL Server 2005. Command prompt installations can be specified at the command prompt or in an .ini file.

Note:

Remote setup is not supported in this release. To install SQL Server 2005 on a remote computer, use a remote connection to run SQL Server Setup in user interface mode or from the command prompt. Record unattended (.iss) setup is not supported in this release.

For examples that show syntax to install or remove an instance of SQL Server 2005, see [How to: Install SQL Server 2005 from the Command Prompt](#).

▣ **See Also**

Concepts

[How to: Rebuild the Registry for SQL Server 2005](#)

[How to: Rebuild the Master Database for SQL Server 2005](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Troubleshooting an Installation of the SQL Server Database Engine



[See Also](#)



[Installing SQL Server 2005 Components](#) > [Installing the SQL Server Database Engine](#) >

Updated: 5 December 2005

The following are some common problems associated with installing the Database Engine and suggested resolution information.

Problem: Applications fail to enlist SQL Server 2005 resources in a distributed transaction

Issue: Because the Microsoft Distributed Transaction Coordinator (MS DTC) is not completely configured in Microsoft Windows, applications may fail to enlist SQL Server 2005 resources in a distributed transaction. This problem can affect linked servers, distributed queries, and remote stored procedures that use distributed transactions.

Resolution: To prevent such problems, you must fully enable MS DTC services on the server where SQL Server 2005 is installed.

To fully enable MS DTC, use the following steps:

1. In **Control Panel**, open **Administrative Tools**, and then open **Computer Management**.
 2. In the left pane of Computer Management, expand **Services and Applications**, and then click **Services**.
 3. In the right pane of Computer Management, right-click **Distributed Transaction Coordinator**, and select **Properties**.
 4. In the Distributed Transaction Coordinator window, click the **General** tab, and then click **Stop** to stop the service.
 5. In the Distributed Transaction Coordinator window, click the **Logon** tab, and set the logon account to **NT AUTHORITY\NetworkService**.
 6. Click **Apply** and **OK** to close the Distributed Transaction Coordinator window. Close Computer Management. Close Administrative Tools.
-

 **Note:**

For installations of SQL Server 2005 on computers participating in a failover cluster, MS DTC must be fully enabled and clustered before you run Setup. If MS DTC is not clustered, Setup will fail. Before running Setup, use the Microsoft Cluster Administrator to ensure that MS DTC has been clustered.

Problem: Upgrade to SQL Server 2005 fails with "SQL Server Setup could not connect to the database service for server configuration." error message.

Issue: If port 1433 is not available during upgrade to SQL Server 2005, SQL Server Setup may return the following error:

SQL Server Setup could not connect to the database service for server configuration.

Resolution: To proceed, terminate the process using port 1433 and continue with SQL Server 2005 Setup:

1. Run **netstat -o** from the command prompt.
2. Identify the application or process using port 1433.
3. Use Task Manager to close the application or terminate the process to free port 1433.
4. Resume SQL Server 2005 Setup.

If you are unable to free port 1433 using the steps above, use the following additional steps:

1. Open **Add or Remove Programs** in Control Panel.
2. Select **Microsoft SQL Server 2005**, and then click **Change**.
3. On the Component Selection page, click the radio button for the SQL Server instance to upgrade, and then click **Next**.
4. On the Feature Maintenance page, click **Database Engine**, and then click **Next**.
5. Click **Resume** to resume the upgrade operation to SQL Server 2005.

Problem: Upgrade may fail if local applications are connected to the

instance being upgraded.

Issue: SQL Server Setup requires single user mode to complete an upgrade to SQL Server 2005. If there are local applications connected to a SQL Server instance during the upgrade operation, Setup will fail with the following error:

SQL Server Setup could not connect to the database service for server configuration. The error was: [Microsoft][SQL Native Client][SQL Server] Login failed for user <DomainName\AdminAccount>. Reason: Server is in single user mode. Only one administrator can connect at this time. Refer to server error logs for more information. For details on how to view Setup logs, see How to: View Setup Log Files" in SQL Server Books Online.

Resolution: This failure is commonly caused when Reporting Services is running during SQL Server upgrade. To work around this issue, stop all applications that connect to the SQL Server instance, then select "Retry" on the Setup error message to allow upgrade to continue.

Issue: SQL Server 2005 Setup fails due to configuration of existing SQL Server components.

Problem: This issue may be caused if an existing SQL Server client installation is configured with the "force encryption" option enabled. The error message for this condition is:

SQL Server Setup could not connect to the database service for server configuration. The error was: [Microsoft]<component name> SSL Provider: The certificate's CN name does not match the passed value. Refer to server error logs and Setup logs for more information. For details on how to view Setup logs, see "How to View Setup Log Files" in SQL Server Books Online.

Resolution: Disable the "force encryption" option on pre-existing SQL Server clients using the SQL Server 2000 client network utility (for MDAC clients in SQL Server 2000) or SQL Configuration Manager (for SQL Native Client clients in SQL Server 2005), and then re-run SQL Server 2005 Setup again.

▣ **See Also**

Tasks

[How to: Verify a Successful Installation of SQL Server 2005 Services](#)

[How to: Read a SQL Server 2005 Setup Log File](#)

Concepts

[Failover Cluster Troubleshooting](#)

[How to: View SQL Server 2005 Setup Log Files](#)

[How to: Rebuild the Registry for SQL Server 2005](#)

[How to: Rebuild the Master Database for SQL Server 2005](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Installing SQL Server Analysis Services



[See Also](#)



[Installing SQL Server 2005 Components](#) >

▣ **Step 1: Preparing to Install SQL Server 2005 Analysis Services**

Before you install Microsoft SQL Server 2005 Analysis Services (SSAS), review the following requirements:

- [Hardware and Software Requirements for Installing SQL Server 2005](#)
- [Check Parameters for the System Configuration Checker](#)
- [Security Considerations for a SQL Server Installation](#)

▣ Step 2: Installing SQL Server 2005 Analysis Services

After you review the installation requirements for SQL Server 2005 and verify that your computer meets those requirements, you are ready to install SQL Server 2005 Analysis Services. Depending on your environment, you install Analysis Services in one of the following ways:

- [Installing SQL Server 2005 Analysis Services on a Clean Computer](#)
- [Installing SQL Server 2005 Analysis Services Side-by-Side with SQL Server 2000 Analysis Services](#)
- [Upgrading to SQL Server 2005 Analysis Services](#)

Installing SQL Server 2005 Analysis Services on a Clean Computer

As with any component in SQL Server 2005, you can install Analysis Services either by running SQL Server Setup through the SQL Server Installation Wizard, or by running the Setup program at the command prompt.

Option 1: Using the SQL Server Installation Wizard

You can install Analysis Services on the local computer or on a failover cluster by using the SQL Server Installation Wizard.

Topic	Description
Installing SQL Server Analysis Services by Using Setup	Setup information that is specific to SQL Server 2005 Analysis Services.
How to: Install SQL Server 2005 (Setup)	Setup information that describes each step in the SQL Server Installation Wizard, and that applies to all components of SQL Server 2005, including Analysis Services.

Option 2: Running the Setup program from the command prompt

You can run SQL Server Setup at the command prompt. You must run the Setup

program at the command prompt when you install Analysis Services on a remote computer.

Topic	Description
Installing SQL Server Analysis Services from the Command Prompt	Command prompt setup information that is specific to Analysis Services.
How to: Install SQL Server 2005 from the Command Prompt	Command prompt setup information that describes each step and option of the command prompt setup process, and that applies to all components of SQL Server 2005, including Analysis Services.

[Back to Top](#)

Installing SQL Server 2005 Analysis Services Side-by-Side with SQL Server 2000 Analysis Services

You can install SQL Server 2005 Analysis Services as a named instance side-by-side with the default instance of Microsoft SQL Server 2000 Analysis Services. Because SQL Server 2000 Analysis Services does not support named instances, the SQL Server 2000 Analysis Services instance must always be the default instance in a side-by-side installation.

You can also migrate existing SQL Server 2000 Analysis Services databases to the SQL Server 2005 Analysis Services database format while leaving the existing databases intact.

Topic	Description
Upgrading or Migrating Analysis Services	General information about side-by-side installation and database migration.
Migrating Existing Analysis Services Databases	Information about the Analysis Services Migration Wizard and the differences between SQL Server 2000 Analysis Services and SQL Server 2005 Analysis Services database objects.
SQL Server 2005	Information about discontinued and deprecated features and

Analysis Services Backward Compatibility	behavior changes from SQL Server 2000 Analysis Services to SQL Server 2005 Analysis Services.
--	---

[Back to Top](#)

Upgrading to SQL Server 2005 Analysis Services

You can upgrade to SQL Server 2005 Analysis Services from an earlier version of Analysis Services by using Setup, or by using several utility programs in addition to SQL Server Setup.

To upgrade during Setup, enter the instance name of an existing instance of SQL Server 2000 Analysis Services. To upgrade after installing a named instance of SQL Server 2005 Analysis Services, use the Analysis Services Migration Wizard to move your existing Analysis Services databases to the new server instance. You can then optionally uninstall SQL Server 2000 Analysis Services and use the Analysis Services Instance Rename tool to make the instance of SQL Server 2005 Analysis Services the default instance.

Topic	Description
Upgrading or Migrating Analysis Services	General information about side-by-side installation and database migration.
Migrating Existing Analysis Services Databases	Information about the Analysis Services Migration Wizard and the differences between SQL Server 2000 Analysis Services and SQL Server 2005 Analysis Services database objects.
SQL Server 2005 Analysis Services Backward Compatibility	Information about discontinued and deprecated features and behavior changes from SQL Server 2000 Analysis Services to SQL Server 2005 Analysis Services.

[Back to Top](#)

▣ **See Also**

Other Resources

[Installing SQL Server 2005 Components](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Installing SQL Server Analysis Services by Using Setup



[See Also](#)



[Installing SQL Server 2005 Components](#) > [Installing SQL Server Analysis Services](#) >

Updated: 5 December 2005

Note:

For detailed information about using the Microsoft SQL Server Installation Wizard to install SQL Server 2005 products components, we recommend that you review [How to: Install SQL Server 2005 \(Setup\)](#).

This topic contains only information specific to the installation of Microsoft SQL Server 2005 Analysis Services (SSAS) by using the SQL Server Installation Wizard.

- [Selecting Components to Install](#)
- [Specifying Analysis Services File Locations](#)
- [Specifying Analysis Services Instances](#)
- [Specifying Analysis Services Service Accounts](#)
- [Specifying Analysis Services Authentication Mode](#)
- [Specifying Analysis Services Collation](#)

▣ Selecting Components to Install

While you can install Analysis Services alone, note that many data warehouse solutions require the installation of additional SQL Server 2005 components to enable the development, deployment, and administration of Analysis Services databases. The following table describes these additional or optional SQL Server 2005 components.

Component	For the Administrator	For the Developer	For Optional Use
Connectivity Components Components for communication between clients and servers, including network libraries for DB-Library, ODBC, and OLE DB.	X	X	
Management Tools Interactive tools for managing SQL Server 2005, including SQL Server Configuration Manager, SQL Server Management Studio, SQL Server Profiler, and Replication Monitor.	X		
Business Intelligence Development Studio The integrated development environment for business intelligence projects.		X	
Software Development Kit Software that assists developers in creating custom solutions for SQL Server 2005.		X	
SQL Server Database Services The SQL Server Database Engine and tools for managing relational and XML			X

data, replication, and full-text search.			
Reporting Services A report server, for managing, executing, rendering, and distributing reports, and the Report Manager, a Web-based user interface that provides access to SQL Server 2005 Reporting Services (SSRS) components, including administrative tools and a user interface for viewing reports.			X
Integration Services SQL Server 2005 Integration Services (SSIS), a set of graphical and programmable objects for moving, copying, and transforming data.			X
Documentation and Samples SQL Server Books Online documentation, programming reference material, and samples for all SQL Server 2005 components.	X	X	X

Installing Selected Components

You can install Analysis Services alone or with additional SQL Server 2005 components, as described in the following procedures.

To install Analysis Services alone

- On the **Components to Install** page of the SQL Server Installation Wizard, select only **Analysis Services**.

 **Note:**

If you choose to install only Analysis Services, then SQL Server 2005 client components, including command-line tools, connectivity components, programming models, management and development tools, and documentation and samples, are not installed by default.

To add or remove installation components by using the Components to Install page

- On the **Components to Install** page of the installation wizard, select or clear the check boxes for those components that you want to add or remove.

 **Note:**

Selecting the **Workstation components, Books Online and development tools** installation option on the **Components to Install** page does not select the sample databases and programming samples by default. You must select these options manually on the **Feature Selection** page.

To add or remove installation components by using the Feature Selection page

1. On the **Components to Install** page, click **Advanced**.
2. On the **Feature Selection** page, select the check boxes for those components that you want to add and clear the check boxes for those components that you want to remove.

 **Note:**

The **Feature Selection** page is a good way to view a detailed list of installation options in the SQL Server Installation Wizard, including the components previously listed in this topic.

▣ Specifying Analysis Services File Locations

SQL Server 2005 Setup puts Analysis Services program files, data files, and shared files in several locations. Each instance of Analysis Services has a separate set of folders in which the program and data files for the instance are stored. The default folder locations for Analysis Services files are listed in the following table.

Analysis Services files	Default location
Instances of SQL Server 2005 and Analysis Services	C:\Program Files\Microsoft SQL Server\MSSQL. <i>n</i> where the number <i>n</i> represents the sequence in which the multiple instances were installed, starting with 1. This location can be changed during setup.
Data and shared files for an instance of Analysis Services	C:\Program Files\Microsoft SQL Server\MSSQL. <i>n</i> \OLAP\Data C:\Program Files\Microsoft SQL Server\MSSQL. <i>n</i> \OLAP\bin
Components that are shared by all instances of the SQL Server 2005 component	C:\Program Files\Microsoft SQL Server\90\Shared This folder cannot be changed during installation.
Tools, documentation, and samples	C:\Program Files\Microsoft SQL Server\90\Tools A single set of tools and documentation is installed for all instances of Analysis Services and other SQL Server components.

To change the default Analysis Services file locations during installation

1. On the **Components to Install** page of the installation wizard, click **Advanced**.
2. On the **Feature Selection** page, click a feature that supports a non-default installation path, and then click **Browse**.
3. In the **Installation path** box of the **Change Folders** page, enter a new file location.

 **Important:**

You can now use the **Feature Selection** page of the installation wizard to install certain SQL Server 2005 components to a location other than the default location of %ProgramFiles%\Microsoft SQL Server\90. This option is available for SQL Server 2005 Integration Services (SSIS), SQL Server 2005 Notification Services, and Workstation Components (which are client tools, documentation, and samples). The corresponding switch to use when you install at the command prompt is **installsqlshareddir**.

▣ **Specifying Analysis Services Instances**

On the **Instance Name** page of the SQL Server Installation Wizard, you can specify whether to install a default instance or a named instance of SQL Server 2005 Analysis Services. You can install only one default instance of SQL Server 2005 Analysis Services. A default instance is not required.

You can specify only one instance name on the **Instance Name** page for all the SQL Server 2005 components that you have selected for installation. Therefore you cannot, for example, install a default instance of the SQL Server Database Engine and a named instance of Analysis Services at the same time. In these circumstances, you must run Setup separately for each instance name that you want to use.

If an instance of SQL Server 2000 Analysis Services is already installed on the server, then you can only install a named instance of SQL Server 2005 Analysis Services, unless you upgrade the existing default instance to SQL Server 2005 Analysis Services. When you enter the name of an existing instance on the **Instance Name** page of the SQL Server Installation Wizard, the **Components to Upgrade** page asks you to confirm your intention to upgrade the existing instance.

When you install SQL Server 2005 Analysis Services side-by-side with SQL Server 2000 Analysis Services, you must install SQL Server 2005 Analysis Services as a named instance.

If you are installing multiple instances of SQL Server 2005 Analysis Services on a single server, you may want to use named instances only, so that you can provide meaningful names to distinguish the instances.

📌 **Note:**

A Microsoft SQL Server 2005 Enterprise Edition license entitles you to install multiple instances of SQL Server 2005 Analysis Services on a single computer. Other editions of SQL Server 2005 require a separate license for each instance. For complete details, refer to the licensing agreement packaged with your product.

For more information about side-by-side SQL Server 2000 Analysis Services and SQL Server 2005 Analysis Services installation scenarios, see [Upgrading or Migrating Analysis Services](#).

You can use the Analysis Services Instance Rename tool to rename an installed instance of SQL Server 2005 Analysis Services, or to make a named instance the default instance when no default instance exists.

For more information, see [How to: Rename an Instance of Analysis Services](#).

▣ Specifying Analysis Services Service Accounts

Analysis Services starts and runs as a Microsoft Windows service. This service appears in the list of installed services in SQL Server Configuration Manager and in Services, the Microsoft Management Console (MMC) snap-in that provides administrative support for Windows services.

Each instance of Analysis Services is supported by its own service, and each instance can use different settings for collation and other server options. The service names for each instance reflect the instance names that you specify during setup. The service name for the default instance is **MSSQLServerOLAPService**. The service name for a named instance is **MSOLAP\$instancename**, where *instancename* is the name that you specify for the instance during setup.

The account used to run the Analysis Services service must be a member of local the Administrators group.

To view an Analysis Services service in the Services snap-in

1. Click **Start**, point to **All Programs**, point to **Administrative Tools**, and then click **Services**.
2. In the list of services, look for one of the following entries:
 - For the default instance of Analysis Services, look for **MSSQLServerOLAPService**.
 - For a named instance of Analysis Services, look for **MSOLAP\$instancename**.

For more information about SQL Server 2005 service accounts, see [Setting Up Windows Service Accounts](#).

▣ **Specifying Analysis Services Authentication Mode**

SQL Server 2005 Analysis Services supports only Windows Authentication.

▣ **Specifying Analysis Services Collation**

Analysis Services uses the same collation designators and sort order settings as the SQL Server Database Engine. If you are installing both Analysis Services and the SQL Server Database Engine under the same instance name, you can choose to use the same settings or different settings for the two servers. If you choose to use different settings, you must configure Analysis Services and the SQL Server Database Engine separately, as described in the following procedure.

To use separate collation settings for Analysis Services and the SQL Server Database Engine

1. On the **Collation Settings** page of the installation wizard, select the **Customize for each service account** check box.
2. Next to **Service**, select **Analysis Services**, and then configure the collation options for Analysis Services.

 **Note:**

The **SQL collations** option does not apply for Analysis Services and is not available when you configure the Analysis Services collation options.

3. Next to **Service**, select **Database Engine** and configure the collation options for the SQL Server Database Engine.

For more information about SQL Server 2005 collation options, see [Collation Options and International Support](#).

▣ **See Also**

Concepts

[Installing SQL Server Analysis Services](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Installing SQL Server Analysis Services from the Command Prompt



[See Also](#)



[Installing SQL Server 2005 Components](#) > [Installing SQL Server Analysis Services](#) >

Updated: 5 December 2005

Note:

We recommend that you review [How to: Install SQL Server 2005 from the Command Prompt](#) for detailed information about installing the Microsoft SQL Server 2005 products and technologies from the command prompt.

This topic contains information specific to installing Microsoft SQL Server 2005 Analysis Services (SSAS) from the command prompt, and does not address other installation methods or the installation of other SQL Server 2005 components.

▣ Analysis Services Command Prompt Installation Parameters

You can install Analysis Services from the command prompt on a local or remote server, or on a failover cluster. When you install from the command prompt, you can specify setup parameters at the command prompt, or you can use the **/settings** switch to load setup parameters from an .ini file. You must use either setup command-line parameters or an .ini file, but you cannot use both.

When you install from the command prompt, you must use the **addlocal="feature1,feature2,..."** switch to specify the SQL Server 2005 features that you want to install. The following table shows the parameters that are required for installing an instance of Analysis Services.

Value	Specifies
<i>Analysis_Server</i>	An instance of Analysis Services
<i>AnalysisDataFiles</i>	The data files for Analysis Services

For example, to install an instance of Analysis Services, together with the product documentation, set the argument **addlocal** as shown in the following example:

 Copy Co

```
setup addlocal="Analysis_Server,AnalysisDataFiles,SQL_BooksOnline
```

For more information about the SQL Server 2005 features that you can specify in the **addlocal** argument, see the template.ini file in the root folder of the SQL Server 2005 installation media, or see [How to: Install SQL Server 2005 from the Command Prompt](#).

▣ Important:

You can now install certain SQL Server 2005 components to a location other than the default location of %ProgramFiles%\Microsoft SQL Server\90. This option is available for SQL Server 2005 Integration Services (SSIS), SQL Server 2005 Notification Services, and Workstation Components (which are client tools, documentation, and samples). Use the **installsqlshareddir** switch for this option when you install from the command prompt.

Other Required Command Prompt Installation Parameters

Additional parameters, such as *PIDKEY*, are required for any SQL Server installation that you perform by using the **setup** command at the command prompt. Other parameters may also be needed for a particular installation of Analysis Services and for other SQL Server 2005 components.

For more information about installing SQL Server 2005 products and technologies from the command prompt, see [How to: Install SQL Server 2005 from the Command Prompt](#).

▣ Additional Analysis Services Command Prompt Installation Parameters

In addition to **addlocal**, the following **/settings** command-line parameters are specific to Analysis Services.

 **Note:**

The arguments listed below cannot be used for failover cluster installations.

Parameter	Description
<i>INSTALLOLAPDIR="OLAP_executable_folder_path"</i>	Specifies the folder in which Analysis Services executables are installed. The default path is C:\Program Files\Microsoft SQL Server\MSSQL.1\OLAP. Install Analysis Services that uses only single-byte characters in the path and
<i>INSTALLOLAPDATADIR="OLAP_data_folder_path"</i>	Specifies the folder in which Analysis Services cubes are installed. The default location is C:\Program Files\Microsoft SQL Server\MSSQL.1\OLAP
<i>OLAPACCOUNT="domain\logon_name"</i>	Specifies the MSSQLServerOLAPService service account domain and name. Use the argument <i>OLAPPASSWORD</i> to specify the password that corresponds to the logon name. If you do not use this argument, the default system account is used. The default is the local system account.
	Specifies whether the MSSQLServerOLAPService

<code>OLAPAUTOSTART={1 0}</code>	<p>service is started automatically when the Microsoft Windows operating system is restarted. Specify 1 to enable the service to automatically start, or specify 0 to disable the service from automatically starting. If you specify this argument, the value 1 will be used.</p>
<code>OLAPPASSWORD="password"</code>	<p>Specifies the password for the MSSQLServerOLAPService service account. If a password is specified by the argument <code>OLAPACCOUNT</code>, this argument is ignored. If you specify this argument, the value will be used as a null password.</p>

For more information about the SQL Server 2005 setup parameters that you can specify when you install from the command prompt, see the `template.ini` file in the root folder of the SQL Server 2005 installation media, or see [How to: Install SQL Server 2005 from the Command Prompt](#).

▣ Specifying Setup Parameters By Using an .Ini File

Optionally, you can specify **setup** command-line parameters by loading them from an .ini file. At the **setup** command line, use the **/settings filename.ini** argument to specify the .ini file.

The **template.ini** file on the SQL Server 2005 installation media provides detailed instructions for creating an .ini file. Note in particular that the first noncommented line in the .ini file must contain the string **[Options]**. The following example illustrates the format of an .ini file:



```
[Options]
USERNAME=MyName
COMPANYNAME=MyCompany
PIDKEY=ABCDE12345FGHIJ67890KLMNO
INSTALLSQLDIR="C:\Program Files\Microsoft SQL Server\"
INSTALLLOLAPDATADIR="C:\Program Files\Microsoft SQL Server\MSSQL\OLAP
ADDLOCAL=Server_Components,Analysis_Server,AnalysisDataFiles,...
INSTANCENAME=MyInstance
OLAPACCOUNT=MyDomain/MyAccount
OLAPPASSWORD=MyPassword
OLAPAUTOSTART=1
...
```

For more information about specifying SQL Server 2005 setup parameters by using an .ini file, see the template.ini file in the root folder of the SQL Server 2005 installation media, or see [How to: Install SQL Server 2005 from the Command Prompt](#).

▣ **See Also**

Concepts

[Installing SQL Server Analysis Services](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Installing SQL Server Reporting Services



[See Also](#)



[Installing SQL Server 2005 Components](#) >

SQL Server Reporting Services (SSRS) is a server-based, end-to-end reporting solution that supports report authoring, distribution, management, and end-user access. Reporting Services includes client and server components. You can install all of the components together on a single computer, or install individual components on separate computers. Examples of installation scenarios involving the use of separate computers include a production report environment that deploys report servers on cluster nodes, a development organization that has multiple Report Designer clients connected to a single report server instance, or a workgroup report server that uses for its internal data storage a remote SQL Server instance available to all employee groups in the organization. You can install Reporting Services with or without other SQL Server components and technologies. You can run multiple instances of a report server on a single server. For more information, see [Installing Multiple Instances of Reporting Services](#).

▣ **Step 1: Preparing to Install Reporting Services**

Before you install Reporting Services, review the following requirements and component descriptions:

- **Planning a Reporting Services Deployment**
- [Hardware and Software Requirements for Installing SQL Server 2005](#)
- [Check Parameters for the System Configuration Checker](#)
- [Security Considerations for a SQL Server Installation](#)
- [Setting Up Windows Service Accounts](#)
- [Selecting Reporting Services Components to Install](#)

▣ **Note:**

An existing report server database file can cause setup to fail if you choose the default installation option. When you choose the default installation option, setup attempts to create a report server database using the default name. If a database with that name already exists, setup will fail and you will have to rollback the installation. To avoid this problem, you can rename the existing database before you run setup or choose the **Install but do not configure server** option so that you can specify custom database settings after setup is finished.

▣ **Step 2: Installing Reporting Services**

After you review the installation requirements for SQL Server 2005 and verify that your computer meets those requirements, you are ready to install SQL Server 2005 Reporting Services. Depending on your environment, you install Reporting Services in one of the following ways:

- Run Setup.exe from a command line to perform a silent or unattended installation. For more information, see [Installing Reporting Services from the Command Prompt](#).
- Run the Installation Wizard to install a default configuration or a files-only configuration. For more information, see [Installing Reporting Services Using Setup](#).

A default configuration installs all components locally on a single computer. This installation option is available only if you are also installing the Database Engine and Reporting Services at the same time.

A files-only installation copies the files and minimally configures the report server. After setup is complete, you use the Reporting Services Configuration tool to bring a report server online. This approach allows you to specify remote computers (for example, choosing a remote Database Engine instance to host the report server databases, or deploy Reporting Services on distributed servers, scale-out deployment, or on Internet-facing Web servers.

For step-by-step instructions, see [Reporting Services Installation How-to Topics](#).

▣ **Step 3: Configuring Reporting Services**

After you install Reporting Services, you can modify the installation to match the deployment model you are implementing. Examples of deployment models that you can effect after Setup is finished include:

- Deploying a report server on an Internet-facing Web server.
- Deploying a report server on a server that hosts other Web applications.
- Deploying one or more report servers in a distributed, multi-server installation.

To learn how to deploy a report server in these configurations, see **Deploying Reporting Services** in SQL Server Books Online after Setup is complete.

If you performed a files-only installation, you must use the Reporting Services Configuration tool to configure a report server so that it is ready to use. For more information, see **Configuring Reporting Services Components** in SQL Server Books Online.

📌 **Important:**

In contrast with previous releases of Reporting Services Setup, you now use the Reporting Services Configuration to install a report server scale-out deployment (previously referred to as "Web farm"). For more information about how changes to setup and scale-out deployment, see **Reporting Services Manageability and Deployment Enhancements** in SQL Server Books Online and [Configuring a Report Server Scale-Out Deployment](#).

▣ **Step 4: Verifying Your Reporting Services Installation**

Following installation and configuration, verify that Reporting Services components are running. For more information, see [How to: Verify a Reporting Services Installation](#).

▣ **See Also**

Tasks

[How to: Install Reporting Services Side-by-Side with an Earlier Version](#)

Concepts

[Installing Multiple Instances of Reporting Services](#)

[Installing Reporting Services on 64-bit Computers](#)

[Upgrading Reporting Services](#)

Other Resources

[Reporting Services Installation How-to Topics](#)

Reporting Services Enhancements

Reporting Services Component Overview

[Installing SQL Server 2005 Components](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Selecting Reporting Services Components to Install



[See Also](#)



[Installing SQL Server 2005 Components](#) > [Installing SQL Server Reporting Services](#) >

SQL Server 2005 Reporting Services (SSRS) consists of server-side and client-side components that can be installed together or separately. Once you know which components you want to use, you can run Setup to install them. For more information about how to install specific components, see [Installing Reporting Services Using Setup](#).

You can run Setup multiple times to install multiple instances of the report server on a single computer. Client components can be installed on any computer, using any edition of Reporting Services. Server components require a SQL Server license. You must have a valid SQL Server license for any computer on which you install a Reporting Services server component.

Note:

The edition you are installing determines the availability of some features. For more information, see [Features Supported by the Editions of SQL Server 2005](#) in SQL Server Books Online.

▣ **Server Components**

The following server components are included with Reporting Services.

Report Server

A report server provides report processing, rendering, and delivery for interactive and scheduled reporting. The report server is implemented as a Microsoft Windows service and as an ASP.NET Web service that runs on Microsoft Internet Information Services (IIS). It is a stateless server that stores metadata and object definitions in a report server database. When the report server component is installed, both the Windows service and Web service are installed on the host computer.

Microsoft recommends that you install a report server on computers that run either Windows Server 2003 or Windows XP. Although you can install and run the report server component on Windows 2000 Server, there are known limitations around impersonation and network access that restrict the use of some report server functionality.

▣ **Important:**

When choosing a computer to host the report server, avoid using a computer that has an underscore in the computer name. Report server does not persist session state information on computers that have an underscore character in the computer name and that have been patched with Internet Explorer Security Patch MS01-055. The security patch prevents cookies from being set on client computers that have an underscore in their names, breaking the session management features of Reporting Services. Recommended solutions are documented in Microsoft Knowledge Base article 316112.

Report Server Database

Reporting Services uses two SQL Server databases to provide internal storage for a report server instance: one for storing data that is used on an ongoing basis, and one for temporary storage. The databases are created together and bound by name. By default, the database names are **reportserver** and

reportservertempdb, respectively. Reporting Services requires a local or remote SQL Server instance to host the report server databases. You can use either a SQL Server 2000 or SQL Server 2005 instance. For more information about edition requirements on the report server database, see [Creating a Report Server Database](#).

The report server databases can be created during or after Setup. Setup can create the report server databases if you are installing a SQL Server 2005 Database Engine instance and Reporting Services concurrently on the same computer. However, if you want to use SQL Server 2000 to host the report server database, or if you want to use any remote SQL Server instance, you must choose the **Install but do not configure server** installation option and use the Reporting Services Configuration tool after Setup completes to create and configure the report server databases. For more information, see [How to: Install a Local Report Server and Remote Report Server Database](#).

Report Manager

Report Manager is a Web-based administration tool used to manage a single report server instance. It is installed with the report server by default. If you are using other tools to manage a report server (for example, SQL Server Management Studio or a custom application), you can omit this component by removing it from the Feature Selection tree.

Report Builder

Report Builder is a ClickOnce application that provides users with a way to create ad hoc reports that are driven off of data models that you create and publish in Model Designer. Report Builder is installed with the report server component. Users can access it through Report Manager or a URL. You can restrict or prevent access to Report Builder by configuring role assignments or setting report server system properties. For more information, see **Securing Report Builder** in SQL Server Books Online.

Reporting Services Configuration

The Reporting Services Configuration tool is installed with a report server instance. You can use it to customize or deploy a report server installation. The

tool is a required component in a report server installation.

▣ Client Components

The following client components are used with Reporting Services. These components can be installed on client computers, and do not require a SQL Server license.

Report Designer

Reporting Services includes a report authoring tool named Report Designer. You can use Report Designer to create, edit, and preview reports, and then deploy them to a report server. Report Designer runs within Visual Studio 2005.

Model Designer

Model Designer is a tool used to create models for ad hoc business reporting in Report Builder. It is used to specify the data relationships that are ultimately used to support ad hoc reporting. Model Designer runs within Visual Studio 2005.

Command Prompt Utilities

Reporting Services includes several command prompt tools that you can use to administer a Reporting Services installation. In most cases, you can use the Reporting Services Configuration tool in place of the command prompt utilities. However, if you prefer a command line approach, you can install the following utilities to perform report server administration tasks:

- **Rsconfig** is used to modify report server database connection settings after installation is complete. It is also used to set encrypted account information used for unattended report processing.
- **Rs** is a script host that can be used to process Microsoft Visual Basic scripts to perform common tasks repeatedly or to run the same tasks on different computers.
- **Rskeymgmt** is used to backup and restore the encryption keys used by the report server.

For more information about each tool, see **rsconfig Utility**, **rs Utility**, and **rskeymgmt Utility** in SQL Server Books Online.

SQL Server Management Studio

You can use Management Studio to administer one or more report server instances. Management Studio provides a comprehensive server management workspace that you can use to manage all of your SQL Server component servers from one place.

SQL Server Configuration Manager

You can use SQL Server Configuration Manager to view and set the properties of the report server Windows service used for scheduled report server operations and report server initialization.

▣ **Choosing a Deployment Model**

You can install all server components on the same computer if you want to conserve SQL Server licenses or if your reporting requirements are not extensive. However, if you are supporting a large number of report users, you will get better performance if you install the report server database on a dedicated server. For more information about deployment models, see **Planning a Reporting Services Deployment** in SQL Server Books Online.

▣ **See Also**

Concepts

[Installing SQL Server Reporting Services](#)

Other Resources

[Reporting Services Component Overview](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Default Configuration for a Report Server Installation



[See Also](#)



[Installing SQL Server 2005 Components](#) > [Installing SQL Server Reporting Services](#) >

When you use the SQL Server Installation Wizard to install the report server component of Reporting Services, Setup provides an option for installing the server in a default configuration. This installation option installs and deploys a report server. When Setup is finished, the report server is running and ready to use. In the Installation Wizard, the option name is **Install the default configuration**. In a command line setup, the default configuration is specified through the **DEFAULT** setting for the **RSCONFIGURATION** property.

▣ Requirements

The default configuration installation option comes with a number of requirements and restrictions, described as follows:

- A local instance of the SQL Server Database Engine must be used to host the report server database. The Database Engine must be installed concurrently with the report server instance.
- The user account used to run Setup must have permission to access and create databases using the local Database Engine instance.
- The default Web site or application pool must be enabled in Microsoft Internet Information Services (IIS), and its Internet Protocol (IP) address must be mapped to **(all unassigned)**. The report server and Report Manager virtual directories will be created in the default Web site.
- The default Web site cannot be configured for Secure Sockets Layer (SSL) connections.
- ASP.NET must be running under the default ASP.NET account. The report server Web service uses the ASP.NET account. On Windows Server 2003, the default ASP.NET account is Network Service. On a Windows 2000 server, the default ASP.NET account is a least-privileged account created specifically for the ASP.NET worker process. If ASP.NET is running under a non-default account, Setup cannot install the default configuration.
- The `IWAM_computername` account must be enabled before you run Setup. Setup uses this account to configure IIS.

□ Default Configuration Settings

The following table compares the settings used in a default configuration with those used in a files-only installation. Except for the Microsoft Windows service account, you cannot specify values in a default configuration during Setup.

Operation	Default Configuration	Files-Only Installation
Configures instance name	Uses the SQL Server instance name	Uses the SQL Server instance name
Installs files	Yes	Yes
Configures Web service account	For Windows XP and Windows 2000, Setup uses the ASP.Net machine account or the Network Service account in machine.config. If the value is not one of these, default setup cannot be used. On Windows Server 2003, Setup uses the account associated with the application pool used with the default Web site.	No
Configures Windows service	Uses the values specified in the Service Accounts page in the Installation Wizard. For Windows XP and Windows 2000, the default is Local System. For Windows Server 2003, the default is Network Service.	Uses the values specified in the Service Accounts page in the Installation Wizard.
Sets folder permissions	Grants permission to the Web service account and Windows service account.	Grants permission to the Windows service account.
Creates report server database	Yes. Creates the reportserver and reportservertempdb databases.	No
Configures	Yes. The Web service account and Windows	

a database connection	service account are granted rights to the database through the rsexecrole.	No
Creates virtual directories	Yes. The virtual directories are created on the default Web site with default names determined by the instance name.	No
Configures e-mail delivery	No	No
Registers the Report Server WMI provider	Yes	Yes
Starts the Report Server Windows service	Yes, if Auto-start service is specified and no setup errors occur.	No
Initializes the report server	Yes	No

▣ **See Also**

Reference

[Report Server Installation Options](#)

Concepts

[Installing SQL Server Reporting Services](#)

[Installing Reporting Services Using Setup](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Installing Reporting Services Using Setup



[See Also](#)



[Installing SQL Server 2005 Components](#) > [Installing SQL Server Reporting Services](#) >

You can use the SQL Server Installation Wizard to install Reporting Services components. You can use the wizard to install some or all of the components.

▣ Choosing Installation Options

When installing the server components, the Installation Wizard provides options that determine how a report server is installed:

- **Install the default configuration** instructs Setup to use default values when installing a report server. This option results in a fully deployed report server that is ready to use once setup is complete. Setup does not prompt you to provide virtual directory names, select a SQL server Database Engine instance, or specify other information to complete the installation. Instead, Setup uses the default values that are described in [Default Configuration for a Report Server Installation](#). The default installation option is available only if you are installing a local Database Engine instance and a Reporting Services instance at the same time. If any of the default values are not available, you cannot choose this installation option.
- **Install but do not configure the server** is used to copy the program files to disk, configure the Report Server Windows service account, and register the report server Windows Management Instrumentation (WMI) provider. This installation option is referred to as a "Files-only" installation. Further configuration is necessary before the report server can be used. The Reporting Services Configuration tool is provided so that you can perform the additional configuration.

To run the Reporting Services Configuration tool, click **Start**, point to **All Programs**, point to **Microsoft SQL Server 2005**, point to **Configuration Tools**, and then click **Reporting Services Configuration**.

Report server installation options are available when you specify **Reporting Services** on the **Components to Install** page or **Report Server** on the **Feature Selection** page. Note that these options apply only to the installation of the server components. If you are installing the client components (such as Report Designer, Model Designer, or the administration tools), these options have no effect.

▣ Choosing Components to Install

You can install server components by selecting **Reporting Services** on the **Components to Install** page. You can install client components by selecting **Workstation components, Books Online and development tools** on the **Components to Install** page.

To pick specific components or to install Reporting Services product documentation or samples, click **Advanced**. Clicking **Advanced** opens the **Feature Selection** page, which provides a complete list of components that can be installed individually.

How to Install Specific Components

Reporting Services includes server components and client components for authoring reports and models, and for administering a report server. You can install SQL Server client components that are useful for managing a server installation. For more information about component descriptions, see [Selecting Reporting Services Components to Install](#).

To install this component	Do this
Report server, Reporting Services Configuration tool, SQL Server Surface Area Configuration tool, SQL Server Configuration Manager, Report Builder	<p>Report server is always installed with tools that are used to enable server functionality. There are no options for excluding these tools from your installation. However, if you select Shared Tools in the Feature Components page, you can specify where the tools are installed on the file system.</p> <ol style="list-style-type: none">1. Select Reporting Services on the Components to Install page.2. Click Advanced.3. Expand Reporting Services.4. (Optional) Remove Report Manager if you do not want to install it.5. (Optional) Select Shared Tools and then click Browse to choose a specific folder location for the

	<p>shared utilities.</p> <p>You cannot remove Report Builder from a server component installation. However, you can disable it after Setup is complete so the tool cannot be run. For more information about how to disable Report Builder, see Securing Report Builder.</p>
Report Manager	<p>Report Manager is selected by default when you choose Reporting Services on the Components to Install page.</p> <p>You can install Reporting Services without Report Manager by removing it from the Feature Components page. You can also deploy Report Manager and the report server on separate computers. For more information about how to install Report Manager and report server on separate computers, see Configuring a Report Server for Internet Access.</p>
Report Designer and Model Designer	<p>Report Designer and Model Designer are available when you install Business Intelligence Development Studio.</p> <ol style="list-style-type: none"> 1. Select Workstation components, Books Online and development tools on the Components to Install page. 2. Click Advanced to verify that Business Intelligence Development Studio is selected or to modify the list of components that will be installed.
Samples reports and applications	<ol style="list-style-type: none"> 1. Select Workstation components, Books Online and development tools on the Components to Install page. 2. Click Advanced. 3. Expand Documentation, Samples, and Sample databases. 4. Select Sample Databases and Sample Code and Applications.

▣ **See Also**

Tasks

[How to: Install a Local Report Server and Remote Report Server Database](#)

[How to: Install Reporting Services on a Single Computer](#)

[How to: Verify a Reporting Services Installation](#)

[How to: Troubleshoot a Reporting Services Installation Problem](#)

Concepts

[Installing SQL Server Reporting Services](#)

[Selecting Reporting Services Components to Install](#)

[Default Configuration for a Report Server Installation](#)

[Installing Reporting Services from the Command Prompt](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Installing Reporting Services from the Command Prompt



[See Also](#)



[Installing SQL Server 2005 Components](#) > [Installing SQL Server Reporting Services](#) >

The Setup.exe program provides a command prompt interface in addition to the graphical Setup program. You can specify setup properties on the command line or in an .ini file to customize the way in which Setup.exe installs Reporting Services components or to suppress the user interface. To run Setup from the command line, you must be a local administrator on the system.

This topic describes the properties used to install Reporting Services components in a command line setup. It also provides examples of common installation scenarios. For more information about syntax and arguments used in a command line setup, see [Running Setup from the Command Prompt](#) and [Specifying Setup Properties with an .ini File](#).

▣ **Command Line Examples for Reporting Services**

The following examples illustrate the command line syntax and settings for common Reporting Services installation scenarios. In order for these examples to work, you must first navigate to the folder that contains the Setup.exe program. Example settings illustrate different ways of specifying a setup .ini file to achieve specific outcomes.

Basic syntax for a silent command line installation

 [Copy Code](#)

```
Setup /settings c:\mysetup.ini /qn
```

Settings for installing Reporting Services components on a local computer

```
[Options]
INSTANCENAME=MSSQLSERVER
ADDLOCAL=SQL_Engine,SQL_Data_Files,RS_Server,RS_Web_Interface,SQL_Wa
SQLAUTOSTART=1
AGAUTOSTART=1
RSAUTOSTART=1
SECURITYMODE=SQL
SAPWD=<password>
PERPROCESSOR=2
RSCONFIGURATION=Default
RSSQLLOCAL=1
```

Settings for installing documentation and samples

```
[Options]
ADDLOCAL=SQL_Documentation,SQL_BooksOnline,SQL_SQLServerBooksOnline,
```

▣ Properties for Reporting Services Components

When you install Reporting Service from the command line, you can specify the following properties. You can specify properties either on the command line or in an .ini file, but not both. If conflicting entries are provided, the entry on the command line is used.

RSACCOUNT="domain\logon_name"

Optional. Default value is the default built-in account for the current operating system. This property applies to the RS_Server feature selection. Specifies the account under which the Report Server Windows service runs. The domain name is limited to 254 characters, and the account name is limited to 20 characters. If a value is not supplied for **RSACCOUNT**, the following table shows the default built-in account selected.

Operating system	Account
Windows 2000	Local System
Windows XP	Local System
Windows Server 2003	Domain User

RSPASSWORD="password"

Optional. Default value is null password. This property applies to the RS_Server feature selection. Specifies the password that corresponds to the user name specified for the **RSACCOUNT** property. The password is limited to 255 characters.

RSUPGRADEDATABASEACCOUNT="username"

Optional. Specifies a SQL Server database login used to connect to the SQL Server instance that hosts the report server database during an upgrade. The database user must have permission to modify the schema. It is useful to set this property if you are upgrading a report server that runs on a remote SQL Server instance. The database user name is limited to 20 characters.

RSUPGRADEPASSWORD="password"

Required if you specified the RSUPGRADEDATABASEACCOUNT.
Specifies the password for the user account. An empty string is a valid value. The password is not stored or saved. The maximum length is 255 characters.

RSCONFIGURATION={ default | filesonly }

Optional. Specifies how a report server instance is installed. The **default** installation requires a local Database Engine instance and a system configuration that uses default values. The Database Engine instance cannot be installed as a virtual server.

The **filesonly** installation installs the program files and minimally configures a report server installation. Further configuration is necessary to make a report server operational. If this property is not specified, **filesonly** is used. For more information, see [Default Configuration for a Report Server Installation](#) and [Report Server Installation Options](#).

RSAUTOSTART={ 1 | 0 }

Optional. Default value is **true (1)**. This property applies to the RS_Server feature selection. Specifies whether the Report Server Windows service is started automatically when Windows is restarted. When the Report Server Windows service is registered with the operating system, the service startup type is set according to this value. Specify true (**1**) to start the service automatically or false (**0**) to set the startup type to manual.

ADDLOCAL = "*feature_selection*"

Specifies the features to be installed either for a new or existing installation. Features are specified as a comma-delimited list. When you install particular features, you must also specify certain properties for the installation to be successful. The following table lists each feature and any properties that are required.

The following table lists the properties available for each component. The feature selection column is the *feature_selection* parameter used with the **ADDLOCAL** or **REMOVE** properties when adding or removing components using the command prompt.

Component	Feature selection value	Available properties
Report Server Web service and Windows service, and Reporting Services Configuration tool	RS_Server	RSACCOUNT, RSPASSWORD, RSAUTOSTART, RSCONFIGURATION
Report Manager	RS_Web_Interface	None
Report Designer and Model Designer	SQL_WarehouseDevWorkbench	None
Reporting Services command line utilities	RS_Tools	None

 **Important:**

When a prerequisite is missing, the associated component cannot be installed. If the affected component is specified on the command line in the **ADDLOCAL** property, setup terminates with an error.

▣ **See Also**

Concepts

[Upgrading Reporting Services](#)

[Selecting Reporting Services Components to Install](#)

[Installing SQL Server Reporting Services](#)

Other Resources

[Reporting Services Backward Compatibility](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Installing the Reporting Services Add-in



[See Also](#)



[Installing SQL Server 2005 Components](#) > [Installing SQL Server Reporting Services](#) >

New: 12 December 2006

The Microsoft SQL Server 2005 Reporting Services Add-in for SharePoint Technologies is a Web download that you can install on a server that has an instance of a SharePoint product or technology. It provides a set of features that enable you to run a report server within a larger deployment of Microsoft Windows SharePoint Services 3.0 or Microsoft Office SharePoint Server 2007.

Features include a Report Viewer Web Part, a proxy server endpoint, and application pages so that you can create, view, and manage reports, report models, and other report server content on a SharePoint site.

You must be a SharePoint farm administrator to install the Reporting Services Add-in.

▣ **Preparing to Install**

Installing the Reporting Services Add-in is one of many steps that are necessary for integrating a report server with an instance of a SharePoint product or technology. For more information about the complete set of steps, see [Configuring Reporting Services for SharePoint 3.0 Integration](#). The order of steps for configuring the servers is important. For best results, be sure that you do them in order.

To prepare to install the Reporting Services Add-in:

1. Verify that you have two server installations - a report server and a SharePoint server - that meet the requirements for integrated operations. For more information, see [Requirements for Running Reporting Services in SharePoint Integrated Mode](#).
2. Download the Reporting Services Add-in to a server that has an instance of the SharePoint product or technology. You can download the Reporting Services Add-in from the [Microsoft Download Center](#).
3. If you are integrating a SharePoint farm with a report server deployment, copy or download the Reporting Services Add-in to each computer in the farm that has a Web server front-end.

▣ **Installing the Reporting Services Add-in on a Standalone Server**

1. Run Setup.exe on a computer that has the Web front-end for a deployment of the SharePoint product or technology that you want to configure for report server integration.

Setup does not provide options for customizing an installation. Setup uses the existing folder structure of the current deployment when adding new files. Setup first verifies whether Windows SharePoint Services 3.0 is installed. Then Setup copies the application files and the proxy server endpoint to the new folder.

A Setup program (.msi) file installs the Reporting Services Add-in in all supported languages. There is no language-specific Setup program.

Application files for all supported languages are installed when you run Setup.

Setup does not configure the server for integration with Reporting Services. You must configure integration after Setup is complete.

2. After Setup is finished, go to Administrator Tools and click **SharePoint 3.0 Central Administration**.
3. Click **Application Management**. You should see a new section named **Reporting Services**. This section has links to pages used to specify integration settings. For instructions on how to specify the settings, see [How to: Set Report Server Integration in SharePoint Central Administration](#).

▣ **Installing the Reporting Services Add-in on Virtual Servers in a SharePoint Farm**

A deployment of Reporting Services integrates with a deployment of a SharePoint product or technology at the farm level. If you have multiple SharePoint Web applications in a farm configuration, you must install the Reporting Services Add-in on each computer that has a Web front-end. After you have installed the add-in on all computers, start Central Administration to configure report server integration for all the servers in the farm.

▣ **Command Line Arguments**

You can append arguments to run Setup in an installation mode.

- Use **SharePointRS.msi -r** to repair an existing installation.
- Use **SharePointRS.msi -u** to uninstall an existing installation of the add-in. Additional steps must be followed to completely uninstall the add-in. For more information, see **How to: Uninstall the Reporting Services Add-in**

▣ **Setup Log Files**

When Setup runs, it logs information to a log file in the %temp% folder for the user who is installed the Reporting Services Add-in. The file name is RS_SP_<number>.log. Each error in the log starts with the string "SSRSCustomActionError".

▣ **See Also**

Tasks

[How to: Uninstall the Reporting Services Add-in](#)

[How to: Install the Windows SharePoint Services Object Model on a Report Server Computer](#)

Concepts

[Deployment Modes for Reporting Services](#)

[Requirements for Running Reporting Services in SharePoint Integrated Mode](#)

[Configuring Reporting Services for SharePoint 3.0 Integration](#)

[Security Overview for Reporting Services in SharePoint Integration Mode](#)

Other Resources

Managing Permissions and Security for Report Server Items on a SharePoint Site

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Installing Reporting Services on 64-bit Computers



[See Also](#)



[Installing SQL Server 2005 Components](#) > [Installing SQL Server Reporting Services](#) >

New: 5 December 2005

The 64-bit editions of Microsoft SQL Server 2005 include Reporting Services. The following table summarizes special considerations for the different types of 64-bit architectures.

Architecture	Special Considerations
x64	<p>SQL Server 2005 Reporting Services is fully supported on x64 operating systems.</p> <p>You can install and run all server and client components of Reporting Services on Microsoft Windows Server 2003 with service pack 1.</p> <p>The operating system, service pack, and the SQL Server edition you are installing must be 64-bit versions.</p>
IA64	<p>This platform is intended for production servers. There is no support for integrated Report Designer or Model Designer in Business Intelligence Development Studio. The design tools are not available for installation on this architecture.</p>
Microsoft Windows-32-on-Windows-64 (WOW64) subsystem	<p>You can run the 32-bit versions of the Reporting Services ASP.NET applications in WOW64. You must specify settings in IIS to enable this configuration. For more information, see How to: Configure IIS to Run 32-bit Reporting Services in WOW64 Mode.</p>

 **Note:**

SQL Server 2000 Reporting Services is not supported on any 64-bit platform. Although SQL Server 2000 Service Pack 4 introduced support for the Microsoft Windows-32-on-Windows-64 (WOW64) subsystem, support for WOW64 does not extend to SQL Server 2000 Reporting Services.

▣ **See Also**

Reference

[Features Supported by the Editions of SQL Server 2005](#)

Concepts

[Installing SQL Server Reporting Services](#)

Other Resources

Integration Services Considerations on 64-bit Computers

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Installing Multiple Instances of Reporting Services



[See Also](#)



[Installing SQL Server 2005 Components](#) > [Installing SQL Server Reporting Services](#) >

New: 14 April 2006

You can install multiple instances of Reporting Services on the same computer. Installing multiple instances is useful if you want to host different content for specific sites. Each instance will have its own report server database, configuration files, and virtual directories. If you want to share a single report server database among multiple instances and you are running SQL Server 2005 Enterprise Edition, you can configure a scale-out deployment. For more information about this deployment scenario, see [Configuring a Report Server Scale-Out Deployment](#).

The following configurations are possible for multi-instance installations:

- Multiple instances of the same version and edition. Each report server instance connects to its own local or remote report server database.
- Multiple instances of different versions and editions. For example, you can run SQL Server 2005 Express, SQL Server 2005 Developer Edition, and SQL Server 2000 Enterprise Edition on the same computer. Multiple instance support is provided through SQL Server 2005. If you are running SQL Server 2000 and 2005 on the same computer, you can have only one SQL Server 2000 Reporting Service instance, and it must run as the default instance (that is, as MSSQLSERVER). Each additional instance must be an edition of SQL Server 2005 and it must be installed as a named instance.

Each instance of SQL Server is isolated from other instances that run on the same computer. You can install different versions and editions as separate instances on the same computer (for example, running the SQL Server 2000 Enterprise Edition and SQL Server 2005 Developer Edition as separate instances).

When you run multiple instances of Reporting Services on the same computer,

you can configure all the instances to use the default Web site in Internet Information Services (IIS). In this case, each report server instance is uniquely identified by its virtual directory. Alternatively, you can configure each instance to use a specific IP Address if you want to use non-default Web sites or route all report server requests through a specific IP address.

▣ **Installing a Report Server Instance**

To install multiple instances of Reporting Services, you must run Setup once for each report server instance you want to install. You cannot install multiple instances at the same time.

Installing a Report Server Default Instance

The first time that you run Setup, you have the option of installing the first report server instance as a default installation if the computer meets the requirements. A default installation is one that uses all default values, resulting in a report server that is ready to use when Setup is complete. You can have only one default instance of Reporting Services on a single SQL Server instance. For more information about requirements for a default installation, see [Default Configuration for a Report Server Installation](#).

If you do not want to use the default values, or if your computer does not satisfy all the requirements for a default installation, you must install Reporting Services as a named instance.

Installing a Report Server as a Named Instance

If you are installing multiple report server instances on the same computer, each additional instance must be installed as a named instance.

To install a named report server instance:

- Specify the instance name in the [Instance Name](#) page of the Installation Wizard.
- Select the **Install but do not configure the server** option on the [Report Server Installation Options](#) page.

On the file system, the program files for each report server instance are kept in separate instance folders (for example, MSSQL.2, MSSQL.3, and so on). Setup creates the folders in the order in which the instance is installed. For more information about how to install multiple instances, see [How to: Install SQL Server 2005 \(Setup\)](#).

▣ **Configuring a Report Server Instance**

To configure different report server instances on the same computer, you can use the Reporting Services Configuration tool to select the instance you want to configure. Reporting Services uses SQL Server instance naming conventions to identify each instance:

- A default instance uses the default instance name of MSSQLServer.
- A default instance of SQL Server Express is SQLExpress (note that SQL Server Express always installs as a named instance).
- Other named instances are identified by the name that you provide at Setup.

For more information about how to connect to a specific report server instance, see [How to: Start Reporting Services Configuration](#).

▣ **Accessing a Report Server Instance in a Multi-instance Deployment**

To access a particular instance of a report server or Report Manager, or to publish reports to a particular instance, you type the URL for the instance you want to use.

The virtual directory for each instance must be unique. If you are creating a virtual directory for a named instance or default SQL Server Express instance, the Reporting Services Configuration tool automatically incorporates instance information to create a unique virtual directory name. For example, suppose you have a server named SERVER01 on which you installed a default report server instance, a named report server instance (identified as TestServer), and a SQL Server Express instance, the default URLs for these instances would be as follows:

- <http://server01/reportserver>
- [http://server01/reportserver\\$SQLExpress](http://server01/reportserver$SQLExpress)
- [http://server01/reportserver\\$TestServer](http://server01/reportserver$TestServer)

▣ **Configuring a Report Server Instance to Use a Custom Web Site or Unique IP Address**

By default, report server instances are configured to the default Web site, where the IP address is mapped to (All Unassigned). You can use the default Web site with its existing configuration to host all the report server instances you install on the same computer. Alternatively, you can run each report server instance under a separate Web site. Using separate Web sites is required if you want to map a specific IP address for each report server instance. For more information about how to configure each report server instance to use a specific Web site or IP address, see [How to: Configure Reporting Services to Use a Non-Default Web Site \(Reporting Services Configuration\)](#).

Change History

Release	History
12 December 2006	Changed content: <ul style="list-style-type: none">• Clarified restrictions around multi-instance installations.•
17 July 2006	Changed content: <ul style="list-style-type: none">• Clarified default installation process.• Moved instructions for how to configure a report server instance to use a specific Web site to a separate topic.

▣ **See Also**

Tasks

[How to: Start Reporting Services Configuration](#)

Reference

[Instance Name](#)

Concepts

[Installing SQL Server Reporting Services](#)

[Configuring Report Server Virtual Directories](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Installing SQL Server Integration Services



[Installing SQL Server 2005 Components](#) >

Updated: **17 July 2006**

▣ **Step 1: Preparing to Install Integration Services**

Before you install Microsoft SQL Server 2005 Integration Services (SSIS), review the following requirements:

- [Hardware and Software Requirements for Installing SQL Server 2005](#)
- [Check Parameters for the System Configuration Checker](#)
- [Security Considerations for a SQL Server Installation](#)

▣ **Note:**

SQL Server 2005 Integration Services does not provide special installation options for a failover cluster. However, after installing Integration Services separately on each node of a cluster, you can manually configure it to operate as a clustered service. For more information, see [How to: Configure Integration Services on a Cluster](#).

▣ **Step 2: Installing Integration Services**

After you review the installation requirements for SQL Server 2005 and ensure that your computer meets those requirements, you are ready to install Integration Services. Depending on your environment, you will install Integration Services in one of the following ways:

- [Installing Integration Services on a Clean Computer](#)
- [Adding Integration Services to an Existing Installation of SQL Server 2005](#)
- [Installing Integration Services Side-by-Side with SQL Server 2000 Data Transformation Services](#)
- [Upgrading to Integration Services](#)

▣ **Note:**

Some SQL Server 2005 components that you may select for installation on the **Components to Install** page or the **Feature Selection** page of the SQL Server Installation Wizard install a partial subset of Integration Services components. These components are useful for specific tasks, but the functionality of Integration Services will be limited. For example, the Database Engine option installs the Integration Services components required for the SQL Server Import and Export Wizard. The Business Intelligence Development Studio option installs the Integration Services components required to design a package, but the Integration Services service is not installed, the Message Queue task is not functional, and you cannot run packages outside of BI Development Studio. To ensure a complete installation of Integration Services, you must select Integration Services on the **Components to Install** page.

▣ **Note:**

For information about SQL Server 2005 features installed on a 64-bit computer, see **Integration Services Considerations on 64-bit Computers**.

Installing Integration Services on a Clean Computer

As with any component of SQL Server 2005, you can install Integration Services either by running SQL Server Setup through the SQL Server Installation Wizard

or by running the Setup program from the command prompt.

Option 1: Using the SQL Server Installation Wizard

You can install Integration Services on the local computer by using the SQL Server Installation Wizard.

Topic	Description
Installing SQL Server Integration Services by Using Setup	Setup information that is specific to Integration Services.
How to: Install SQL Server 2005 (Setup)	Setup information that describes each step of the SQL Server Installation Wizard, and is applicable to all components of SQL Server 2005, including Integration Services.

Option 2: Running the Setup program from the command prompt

You can run SQL Server Setup from the command prompt. You must run the Setup program from the command prompt when you install Integration Services on a remote computer.

Topic	Description
Installing SQL Server Integration Services from the Command Prompt	Command prompt setup information that is specific to Integration Services.
How to: Install SQL Server 2005 from the Command Prompt	Command prompt setup information that describes each step and option of the command prompt setup process, and is applicable to all components of SQL Server 2005, including Integration Services.

Adding Integration Services to an Existing Installation of SQL Server 2005

You can also add Integration Services to an existing installation of SQL Server

2005. For more information, see [How to: Add Integration Services to an Existing Instance of SQL Server 2005](#).

Installing Integration Services Side-by-Side with SQL Server 2000 Data Transformation Services

You can install Integration Services side-by-side with Microsoft SQL Server 2000 Data Transformation Services (DTS). Later, you can optionally migrate existing DTS packages to the Integration Services package format while leaving the existing packages intact.

Topic	Description
Upgrading or Migrating Data Transformation Services	General information about side-by-side installation and package migration.
Migrating Data Transformation Services Packages	Information about the Package Migration Wizard and the differences between SQL Server 2000 Data Transformation Services and Integration Services packages.
SQL Server 2005 Integration Services Backward Compatibility	Information about support for SQL Server 2000 Data Transformation Services in Integration Services.

Upgrading to Integration Services

You can upgrade to Integration Services from Data Transformation Services by using SQL Server Setup to install Integration Services, and then optionally using the Package Migration Wizard to migrate your existing DTS packages to the new Integration Services package format while leaving the existing packages intact.

Topic	Description
Upgrading or Migrating Data Transformation Services	General information about side-by-side installation and package migration.

Migrating Data Transformation Services Packages	Information about the Package Migration Wizard and the differences between SQL Server 2000 Data Transformation Services and Integration Services packages.
SQL Server 2005 Integration Services Backward Compatibility	Information about support for SQL Server 2000 Data Transformation Services in Integration Services.

▣ Change History

Release	History
17 July 2006	Changed content: <ul style="list-style-type: none"><li data-bbox="477 506 1349 590">• Added section about adding Integration Services to an existing installation of SQL Server 2005.
5 December 2005	Changed content: <ul style="list-style-type: none"><li data-bbox="477 716 1414 842">• Added notes directing users to topics on Integration Services cluster configuration and 64-bit features for more information.

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Installing SQL Server Integration Services by Using Setup



[Installing SQL Server 2005 Components](#) > [Installing SQL Server Integration Services](#) >

Updated: 12 December 2006

Note:

For detailed information about how to use the SQL Server Installation Wizard to install Microsoft SQL Server 2005 components, we recommend that you review [How to: Install SQL Server 2005 \(Setup\)](#).

This topic contains only information specific to the installation of Microsoft SQL Server 2005 Integration Services (SSIS) by using the SQL Server Installation Wizard.

- [Selecting Components to Install](#)
- [Specifying Integration Services File Locations](#)
- [Installing the Integration Services Service](#)
- [Installing Support for SQL Server 2000 Data Transformation Services Packages](#)

You can also add Integration Services to an existing installation of SQL Server 2005. For more information, see [How to: Add Integration Services to an Existing Instance of SQL Server 2005](#).

Note:

Some SQL Server 2005 components that you may select for installation on the **Components to Install** page or the **Feature Selection** page of the SQL Server Installation Wizard install a partial subset of Integration Services components. These components are useful for specific tasks, but the functionality of Integration Services will be limited. For example, the Database Engine option installs the Integration Services components that are required for the SQL Server Import and Export Wizard. The Business Intelligence Development Studio option installs the Integration Services

components that are required to design a package, but there are limitations: the Integration Services service is not installed, the Message Queue task is not functional, and you cannot run packages outside BI Development Studio. To ensure a complete installation of Integration Services, you must select Integration Services on the **Components to Install** page.

 **Note:**

For information about SQL Server 2005 features installed on a 64-bit computer, see **Integration Services Considerations on 64-bit Computers**.

▣ Selecting Components to Install

Although you can install Integration Services alone, many data warehouse solutions require the installation of additional SQL Server 2005 components to enable the development, deployment, and administration of Integration Services packages. The following table describes these additional or optional SQL Server 2005 components.

Component	For the Administrator	For the Developer	For Optional Use
Connectivity Components Components for communication between clients and servers, including network libraries for DB-Library, ODBC, and OLE DB.	X	X	
Management Tools Interactive tools for managing SQL Server 2005, including SQL Server Configuration Manager, SQL Server Management Studio, SQL Server Profiler, and Replication Monitor.	X		
Business Intelligence Development Studio The integrated development environment for business intelligence projects.		X	
Software Development Kit Software that assists developers in creating custom solutions for SQL Server 2005.		X	
SQL Server Database Services The SQL Server Database Engine and tools for managing relational and XML			X

data, replication, and full-text search.			
SQL Server Analysis Services SQL Server 2005 Analysis Services (SSAS) and tools used to support online analytical processing (OLAP) and data mining.			X
SQL Server Reporting Services The Report Server, for managing, executing, rendering, and distributing reports, and the Report Manager, a Web-based user interface that provides access to SQL Server 2005 Reporting Services (SSRS) components, including administrative tools and a user interface for viewing reports.			X
Documentation and Samples SQL Server Books Online documentation, programming reference material, and samples for all SQL Server 2005 components.	X	X	X

Installing Selected Components

You can install Integration Services alone or with additional SQL Server 2005 components, as described in the following procedures.

☐ **To install Integration Services only**

- On the **Components to Install** page of the SQL Server Installation Wizard, select only **Integration Services**.

Note:

If you choose to install only Integration Services, then SQL Server 2005 client components, including command-line tools, connectivity

components, programming models, management and development tools, documentation, and samples, are not installed by default.

☐ **To add or remove installation components by using the Components to Install page**

- On the **Components to Install** page, select the check boxes for those components that you want to add and clear the check boxes for those components that you want to remove.

📌 **Note:**

Selecting the **Workstation components, Books Online and development tools** installation option on the **Components to Install** page does not select the sample databases and programming samples. To install sample databases and programming samples, you must select these options manually on the **Feature Selection** page.

☐ **To add or remove installation components by using the Feature Selection page**

1. On the **Components to Install** page, click **Advanced**.
2. On the **Feature Selection** page, select the check boxes for those components that you want to add and clear the check boxes for those components that you want to remove.

📌 **Note:**

The **Feature Selection** page is a good way to view a detailed list of installation options in the SQL Server Installation Wizard, including the components previously listed in this topic.

▣ Specifying Integration Services File Locations

SQL Server 2005 Setup puts Integration Services program files and shared files in several locations. The default folder locations for Integration Services files are listed in the following table.

Integration Services files	Default location
Integration Services core components and the dtexec command prompt utility	C:\Program Files\Microsoft SQL Server\90\DTS\Binn
Integration Services connection managers	C:\Program Files\Microsoft SQL Server\90\DTS\Connections
Integration Services enumerators	C:\Program Files\Microsoft SQL Server\90\DTS\ForEachEnumerators
Files in XML format that map data types from one database version or system to another (for example, from SQL Server 2000 to SQL Server 2005, or from SQL Server 2005 to Oracle). These mapping files are used by the SQL Server Import and Export Wizard.	C:\Program Files\Microsoft SQL Server\90\DTS\MappingFiles
Integration Services data flow components, consisting of sources, transformations, and destinations	C:\Program Files\Microsoft SQL Server\90\DTS\PipelineComponents
Integration Services run-time tasks	C:\Program Files\Microsoft SQL Server\90\DTS\Tasks
Components that are shared by all instances of SQL Server 2005 components	C:\Program Files\Microsoft SQL Server\90\Shared This folder cannot be changed during installation.
Tools, documentation, and samples	C:\Program Files\Microsoft SQL Server\90\Tools A single set of tools and documentation is installed for all instances of Analysis Services and

▣ **To change the default Integration Services file locations during installation**

1. On the **Components to Install** page, click **Advanced**.
2. On the **Feature Selection** page, click a feature that supports a non-default installation path, and then click **Browse**.
3. In the **Installation path** box of the **Change Folders** page, enter a new file location.

▣ **Important:**

You now have the option on the **Feature Selection** page to install certain SQL Server 2005 components to a location other than the default location of %ProgramFiles%\Microsoft SQL Server\90. This option is available for SQL Server 2005 Integration Services, SQL Server 2005 Notification Services, and Workstation Components (client tools, documentation, and samples). The corresponding switch to use when you install from the command prompt is **installsqlshareddir**.

▣ **Installing the Integration Services Service**

Integration Services installs the Integration Services service, a new feature that provides several enhancements and that has no equivalent in SQL Server 2000 Data Transformation Services. The Integration Services service allows you to do the following in SQL Server Management Studio:

- Monitor the execution of all Integration Services packages that are running on a computer.
- Display a hierarchical view of Integration Services packages and folders that are stored in different physical locations.

📌 **Note:**

This service is not required to create, store, and execute SSIS packages.

The Integration Services service is installed and enabled by default when you select the **SQL Server Integration Services** component on the **Components to Install** page. You can only install one instance of the service on a computer.

The service is registered to run under the Network Service account. (On Windows 2000, the service runs under the Local System account.) This setting cannot be modified through the SQL Server Installation Wizard, and is not affected by any selections that you make on the **Service Account** page of the wizard.

The service runs in its own process, and is not dependent upon the SQL Server Agent process.

By default, the Integration Services service stores packages that you store in SQL Server in the local, default instance of the Database Engine. If you want to store packages in a named instance or on another server, you have to edit the configuration file for the Integration Services service. For more information, see **Configuring the Integration Services Service**.

To manage the service, use SQL Server Configuration Manager, or use the **Services** snap-in for Microsoft Management Console (MMC) by opening **Administrative Tools** in **Control Panel**. For more information, see **Managing**

SSIS Service.

▣ **Installing Support for SQL Server 2000 Data Transformation Services Packages**

When you select **Integration Services** for installation, Setup also installs support for SQL Server 2000 Data Transformation Services (DTS) packages, including the DTS runtime and DTS package enumeration in SQL Server Management Studio. Run-time support has been updated to allow DTS packages to access SQL Server 2005 data sources. If you are not installing Integration Services, but you need support for DTS packages, then you need to make sure that **Legacy Components** is selected on the **Feature Selection** page.

The backward compatibility files that are installed also include the SQL Server 2000 Analysis Services Processing task and its dependency, Decision Support Objects (DSO). However they do not include the SQL Server 2000 Data Mining Prediction Query task.

▣ **Note:**

SQL Server 2005 does not install the DTS package designer for editing DTS packages. For more information, see [SQL Server 2005 Integration Services Backward Compatibility](#). DTS packages cannot be opened or edited in BI Development Studio.

When you select Integration Services for installation, the Package Migration Wizard is also installed. You can optionally use the Package Migration Wizard to upgrade SQL Server 2000 DTS packages to the Integration Services format. During migration, the wizard copies the DTS packages and then re-creates them in the Integration Services format. The source packages are left intact and are not modified. For more information, see [Migrating Data Transformation Services Packages](#).

▣ **Note:**

The Package Migration Wizard is available in the Standard, Enterprise, and Developer Editions of SQL Server 2005.

▣ Change History

Release	History
12 December 2006	<p>New content:</p> <ul style="list-style-type: none">• Added link to topic about installing Integration Services on an existing instance of SQL Server.• Added link to topic regarding how to modify the configuration file for the Integration Services service to store packages in a location other than the default.
5 December 2005	<p>New content:</p> <ul style="list-style-type: none">• Added note directing users to the Integration Services 64-bit features topic for more information. <p>Changed content</p> <ul style="list-style-type: none">• Improved the explanation of installation options for legacy components.

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Installing SQL Server Integration Services from the Command Prompt



[Installing SQL Server 2005 Components](#) > [Installing SQL Server Integration Services](#) >

Updated: 17 July 2006

Note:

For detailed information about how to install Microsoft SQL Server 2005 components from the command prompt, we recommend that you review [How to: Install SQL Server 2005 from the Command Prompt](#).

This topic contains only information specific to the installation of Microsoft SQL Server 2005 Integration Services (SSIS) from the command prompt.

Note:

Some SQL Server 2005 components that you may select for installation install a partial subset of Integration Services components. These components are useful for specific tasks, but the functionality of Integration Services will be limited. For example, the Database Engine option installs the Integration Services components required for the SQL Server Import and Export Wizard. The Business Intelligence Development Studio option installs the Integration Services components required to design a package, but there are limitations: the Integration Services service is not installed, the Message Queue task is not functional, and you cannot run packages outside of BI Development Studio. To ensure a complete installation of Integration Services, you must select Integration Services for installation.

Note:

For information about SQL Server 2005 features installed on a 64-bit computer, see **Integration Services Considerations on 64-bit Computers**.

▣ Integration Services Command Prompt Installation Parameters

You can install Microsoft SQL Server 2005 Integration Services (SSIS) from the command prompt on a local or remote server. When you install from the command prompt, you can specify setup parameters at the command prompt, or you can use the **/settings** switch to load setup parameters from an .ini file. You must use either setup command line parameters or an .ini file, but you cannot use both.

When you install from the command prompt, you must use the **addlocal="feature1,feature2,..."** switch to specify the SQL Server 2005 features that you want to install. The following table shows the parameters that you can specify in an installation of Integration Services, including the Integration Services components, management and development tools, the SQL Server Database Engine, and SQL Server Books Online.

Value	Specifies
SQL_DTS	The Integration Services components. Also installs support for Microsoft SQL Server 2000 Data Transformation Services (DTS) packages, including the DTS runtime and DTS package enumeration in SQL Server Management Studio.
SQL_WarehouseDevWorkbench	Business Intelligence Development Studio.
SQL_Tools90	SQL Server Management Studio.
SQL_Engine	The SQL Server Database Engine.
SQL_Data_Files	The data files for the SQL Server Database Engine. This option is required when you specify the SQL_Engine option.
SQL_BooksOnline	SQL Server Books Online. If you do not specify this option, context-sensitive Help will not be available in designers, editors, and wizards.

For example, to install an instance of Integration Services, along with the

product documentation, set the **addlocal** argument as shown in the following example:

 [Copy Code](#)

```
setup addlocal="SQL_DTS,SQL_BooksOnline"
```

For more information about the SQL Server 2005 features that you can specify in the *addlocal* argument, see the *template.ini* file in the root folder of the SQL Server 2005 installation media, or see [How to: Install SQL Server 2005 from the Command Prompt](#).

 **Important:**

You now have the option to install certain SQL Server 2005 components to a location other than the default location of %ProgramFiles%\Microsoft SQL Server\90. This option is available for Integration Services, SQL Server 2005 Notification Services, and Workstation Components (client tools, documentation, and samples). The switch to use for this option when you install from the command prompt is **installsqlshareddir**.

Other Required Command Prompt Installation Parameters

Additional parameters, such as **PIDKEY**, are required for any installation that you perform by using the **setup** command from the command prompt. Other parameters may also be necessary for a particular installation of Integration Services and other SQL Server 2005 components.

For more information about installing the SQL Server 2005 products and technologies from the command prompt, see [How to: Install SQL Server 2005 from the Command Prompt](#).

▣ Specifying Setup Parameters by Using an .Ini File

Optionally, you can specify **setup** command prompt parameters by loading them from an .ini file. On the **setup** command line, use the **/settings filename.ini** argument to specify the .ini file.

The template.ini file on the SQL Server 2005 installation media provides detailed instructions for creating an .ini file. Note in particular that the first noncommented line in the .ini file must contain the string, [Options]. The following example illustrates the format of an .ini file:

 [Copy Code](#)

```
[Options]
USERNAME=MyName
COMPANYNAME=MyCompany
PIDKEY=ABCDE12345FGHIJ67890KLMNO
ADDLOCAL=SQL_DTS, SQL_BooksOnline, ...
...
```

For more information about how to specify SQL Server 2005 setup parameters by using an .ini file, see the template.ini file in the root folder of the SQL Server 2005 installation media, or see [How to: Install SQL Server 2005 from the Command Prompt](#).

▣ Change History

Release	History
5 December 2005	Changed content: <ul style="list-style-type: none">• Added note directing users to Integration Services 64-bit topic for more information.• Clarified the limitations of an incomplete Integration Services installation.

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Installing SQL Server Notification Services



[See Also](#)



[Installing SQL Server 2005 Components](#) >

Microsoft SQL Server Notification Services is a platform for developing and deploying applications that generate and send notifications. You can use Notification Services to send timely, personalized messages to thousands or millions of subscribers, and you can deliver the messages to a wide variety of devices.

This section provides information about installing Notification Services.

▣ **Step 1: Verify Prerequisites**

Before you install Notification Services, review the following requirements:

- [Hardware and Software Requirements for Installing SQL Server 2005](#)
- [Check Parameters for the System Configuration Checker](#)
- [Security Considerations for a SQL Server Installation](#)

▣ **Step 2: Decide Which Components to Install**

Notification Services has two sets of installable components: Engine and Client. Depending on your deployment, you might install all or just one set of components.

- Engine components run Notification Services instances and applications.
- Client components are used to interact with instance and application data from other servers.

Depending on how you use Notification Services, you install its components on one or more servers. You must install components on any server that hosts the following Notification Services functions:

- Any server that runs a hosted event provider, generator, or distributor requires the engine components.
- Any server that runs a non-hosted event provider or subscription management interface requires the client components. The client components are required for registering the instance and using the Notification Services API.
- Any server from which you administer instances of Notification Services requires the client components. The client components are required for registering the instance and interacting with the instance. The client components are included with SQL Server Management Studio.

▣ **Step 3: Install Notification Services**

When you are ready to install Notification Services, use one of the following methods to perform the installation:

- [Installing Notification Services Using Setup](#)
- [Installing Notification Services from the Command Prompt](#)

▣ **Step 4: Configure Notification Services**

After you install Notification Services, you can create Notification Services applications, configure instances of Notification Services to host those applications, and deploy the instances of Notification Services. For more information, see the following sections of SQL Server Books Online:

- **Building Notification Solutions**
- **Deploying Notification Services**

After you deploy instances of Notification Services, you can configure those instances using the SQL Server Management Studio or the **nscontrol** command-prompt utilities. For more information, see **Administering Notification Services**.

▣ **Uninstalling Notification Services**

If you are permanently removing Notification Services, you should delete your instances of Notification Services before uninstalling the product. If you are upgrading instances of Notification Services to a newer version of SQL Server, leave your instances in place and follow the Notification Services migration instructions. For more information, see the following topics:

- **Deleting Instances of Notification Services**
- [Migrating Notification Services Instances](#)
- [How to: Uninstall Notification Services](#)

▣ **See Also**

Other Resources

[Installing SQL Server 2005 Components](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Installing Notification Services Using Setup



[See Also](#)



[Installing SQL Server 2005 Components](#) > [Installing SQL Server Notification Services](#) >

When you install Notification Services, you do not install an instance of Notification Services. Rather, you install the Notification Services binary files. You then configure and deploy instances of Notification Services, which host notification applications.

An instance of Notification Services can run on a variety of system configurations. For example, the Notification Services engine and the databases can be on one server, or the engine and databases can be on separate servers. Additionally, you might have subscription management interfaces or non-hosted event providers located on remote servers.

You must install Notification Services wherever you want to run the Notification Services engine (which runs the hosted event providers, generators, and distributors) and you must install the Notification Services client components wherever you want to run a non-hosted event provider, a subscription management interface, or an administrative console.

▣ **Single-Server Installations**

In a single-server installation, Notification Services and the Microsoft SQL Server Database Engine run on the same server. This is common for development systems and small to medium deployment systems.

When you run Setup, you install an instance of the SQL Server Database Engine and Notification Services. You may also want to install the Workstation Components so that you can use SQL Server Management Studio to create and administer instances of Notification Services using SQL Server Management Studio.

To install Notification Services and the Database Engine

- [How to: Install Notification Services and the Database Engine](#)

▣ **Remote Notification Services Engines**

Notification Services engine components, which run hosted event providers, generators, and distributors, can be on a separate server from the Database Engine.

When you define a Notification Services application, you specify the server where each hosted event provider, generator, and distributor runs. On each specified server, you need to install the Notification Services engine components. You install these components by installing Notification Services.

▣ **Note:**

The Feature Selection dialog in SQL Server Setup allows you to install Engine Components and Client Components. Installing the Engine Components is equivalent to installing all of Notification Services.

To install Notification Services

- [How to: Install the Notification Services Engine Components](#)

▣ **Remote Notification Services Clients**

Most Notification Services applications require one or more client applications. Servers that run client applications, but not the engine components, require the Notification Services client components.

The following types of applications use the client components:

- Subscription management interfaces
- Non-hosted event providers
- Administrative consoles

Installing the Notification Services client components installs all Notification Services files except for the files required to run the Notification Services engine.

 **Note:**

The Notification Services client components are installed when you install SQL Server Management Studio.

To install client components

- [How to: Install the Notification Services Client Components](#)

▣ **See Also**

Concepts

[Installing SQL Server Notification Services](#)

[Installing Notification Services from the Command Prompt](#)

[Migrating Notification Services Instances](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Installing Notification Services from the Command Prompt



[See Also](#)



[Installing SQL Server 2005 Components](#) > [Installing SQL Server Notification Services](#) >

The Microsoft SQL Server 2005 Setup program provides a command prompt interface in addition to the graphical user interface. Use the information in this topic to install Notification Services from the command prompt. For additional information about installing SQL Server 2005 from the command prompt, including the complete syntax and list of parameters, see [How to: Install SQL Server 2005 from the Command Prompt](#).

▣ Notification Services ADDLOCAL Values

Each Notification Services component can be installed from the command prompt using the **ADDLOCAL** argument for Microsoft Windows Installer. To install a component from a command prompt, you must use the correct command-line name. The component name and command-line name for each component are shown in the following table.

Component Name	Command-Line Name
All Notification Services components	Notification_Services
Engine components	NS_Engine
Client components	NS_Client

▣ Examples

The following examples show how to install Notification Services from the command prompt.

Installing all Notification Services components

This example shows how to install all components of Notification Services, displaying progress dialog boxes:

```
start /wait setup.exe /qb ADDLOCAL=Notification_Services
```

Installing Notification Services client components

This example shows how to perform a new installation of the Notification Services client components. No dialog boxes are displayed, but the command prompt does not return until installation is complete:

```
start /wait setup.exe /qn ADDLOCAL=NS_Client
```

Removing Notification Services client components

This example shows how to remove Notification Services client components, displaying progress dialog boxes:

```
start /wait setup.exe REMOVE=NS_Client
```

▣ **See Also**

Concepts

[Installing SQL Server Notification Services](#)

[Installing Notification Services Using Setup](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Installing SQL Server Replication



[See Also](#)



[Installing SQL Server 2005 Components](#) >

Updated: **14 April 2006**

Replication components can be installed using the Microsoft SQL Server 2005 Installation Wizard, at the command prompt, or by using an .ini file. Install replication when installing an instance of SQL Server or when modifying an existing instance.

After replication components are installed, you must configure the server before you can use replication. For more information, see **Configuring and Maintaining Replication** in SQL Server 2005 Books Online.

Important:

If you install replication components when modifying an existing instance of SQL Server, you must stop and restart SQL Server Agent after the install is complete. This action ensures that SQL Server Agent recognizes the replication agent subsystems and can call replication agents from job steps.

▣ **Installing Replication by Using Setup**

To install replication when installing a new instance of SQL Server

- To install replication components, including Replication Management Objects (RMO), select **SQL Server Database Services** on the **Components to Install** page of the Installation Wizard.

To install replication when modifying an existing instance of SQL Server

- To install replication components:
 1. On the **Components to Install** page of the SQL Server Installation Wizard, click **Advanced**.
 2. On the **Feature Selection** page, expand **Database Services**.
 3. Click **Replication**, and then click **Entire feature will be installed on local hard drive**.
- To install Replication Management Objects (RMO):
 1. On the **Components to Install** page of the SQL Server Installation Wizard, click **Advanced**.
 2. On the **Feature Selection** page, expand **Client Components**.
 3. Click **Software Development Kit**, and then click **Entire feature will be installed on local hard drive**.

▣ Installing Replication from the Command Prompt

To install replication when installing a new instance of SQL Server

- To install replication components:
 - Specify a value of **SQL_Engine** for the ADDLOCAL parameter at the command prompt or in an .ini file. For example, the command prompt could be:

```
start /wait setup.exe /qb ADDLOCAL=SQL_Engine SQLACCOUNT=dor
```

- To install Replication Management Objects (RMO):
 - Specify a value of **SDK** for the ADDLOCAL parameter at the command prompt or in an .ini file. For example, the command prompt could be:

```
start /wait setup.exe /qb ADDLOCAL=SQL_Engine,SDK SQLACCOUNT
```

To install replication when modifying an existing instance of SQL Server

- To install replication components:
 - Specify a value of **SQL_Replication** for the ADDLOCAL parameter at the command prompt or in an .ini file. For example, the command prompt could be:

```
start /wait setup.exe /i <installation package> /qb ADDLOCAL
```

- To install Replication Management Objects (RMO):
 - Specify a value of **SDK** for the ADDLOCAL parameter at the command prompt or in an .ini file. For example, the command prompt could be:

```
start /wait setup.exe /i <installation package> /qb ADDLOCAL
```

 [Cop](#)

▣ Change History

Release	History
14 April 2006	<p data-bbox="321 424 537 457">New content:</p> <ul data-bbox="435 506 1424 632" style="list-style-type: none"><li data-bbox="435 506 1424 632">• Added information about the requirement to stop and restart SQL Server Agent if you install replication components when modifying an existing instance of SQL Server.

▣ **See Also**

Concepts

[Installing SQL Server 2005](#)

[Running Setup from the Command Prompt](#)

[Specifying Setup Properties with an .ini File](#)

Other Resources

[Installing SQL Server 2005 Components](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Installing AdventureWorks Sample Databases and Samples



[See Also](#)



[Installing SQL Server 2005 Components](#) >

In Microsoft SQL Server 2005, new Online Transaction Processing (OLTP), Online Analytical Processing (OLAP), and data warehouse sample databases are provided. These databases are used in samples and in code examples included in SQL Server 2005 Books Online. Sample databases used in earlier versions of SQL Server are available for download.

▣ In This Section

Topic	Description
Running Setup to Install AdventureWorks Sample Databases and Samples	Describes how to install the AdventureWorks , AdventureWorksDW , and Adventure Works DW sample databases and samples using Setup.
Reinstalling Sample Databases from Script	Describes how to reinstall the AdventureWorks or AdventureWorksDW sample databases by running an installation script.
Downloading Northwind and pubs Sample Databases	Describes how to download the Northwind and pubs sample databases from the Microsoft Download Center.

▣ **See Also**

Other Resources

Sample Databases and Business Scenarios

[Installing SQL Server 2005 Components](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Running Setup to Install AdventureWorks Sample Databases and Samples

 [See Also](#)



[Installing SQL Server 2005 Components](#) > [Installing AdventureWorks Sample Databases and Samples](#) >

Updated: 17 July 2006

The **AdventureWorks** (OLTP), **AdventureWorksDW** (data warehouse), and **Adventure Works DW** (analysis services) sample databases, as well as the companion samples, are not installed by default in SQL Server 2005. You can download these from [SQL Server 2005 Samples and Sample Databases](#) at the Microsoft Download Center, or you can use the following procedures to install the sample databases and samples during or after setup. Additional instructions for deploying the **Adventure Works DW** analysis services project are also provided.

Note:

The following procedures do not apply to SQL Server 2005 Express Edition or SQL Server 2005 Compact Edition. To download **AdventureWorks** for these editions, see [Installing Sample Databases for Express Editions](#).

Important:

When the **AdventureWorks** and **AdventureWorksDW** sample databases are installed, the collation used is SQL_Latin1_General_CP1_CI_AS and not the collation selected for the instance of SQL Server. To change the collation of a sample database to match the server collation, drop the database and recreate it from the installation script. For more information, see [Reinstalling Sample Databases from Script](#).

▣ Installation during setup

To install the sample databases and companion samples during setup, use the following instructions.

1. On the **Components to Install** page, select **Workstation components, Books Online and development tools**.
2. Click **Advanced** and then expand **Documentation, Samples, and Sample Databases**.
3. Select **Sample Code and Applications**.
4. Expand **Sample Databases** and then select the sample databases to install.

☑ **Note:**

The OLAP database (**Adventure Works DW**) cannot be installed unless the data warehouse (**AdventureWorksDW**) database is selected for installation. The analysis services project, **Adventure Works DW** must be deployed before the **Adventure Works DW** analysis services database can be accessed.

5. To complete the installation of the samples, after setup, perform one of the following steps:
 1. From the **Start** menu, click **All Programs**, click **Microsoft SQL Server 2005**, click **Documentation and Tutorials**, click **Samples**, and then click **Microsoft SQL Server 2005 Samples**.
 2. Alternatively, using Windows Explorer, navigate to <drive>:\Program Files\Microsoft SQL Server\90\Tools\Samples\ and double-click **SqlServerSamples.msi** to launch the installer.

☑ **Important:**

The samples are installed to <drive>:\Program Files\Microsoft SQL Server\90\Samples. Be sure to read and follow the installation instructions in the Readme files that accompany the samples you want to use. For a list of samples, see **Samples**.

▣ Installation after setup

If you did not install the sample databases or samples during the initial setup of SQL Server 2005, you can install them later.

🚩 Important:

You cannot use the following procedure to re-create a sample database that has been deleted. To re-create a deleted sample database, reinstall the database from the installation script. For more information, see [Reinstalling Sample Databases from Script](#).

1. From **Add or Remove Programs**, select **Microsoft SQL Server 2005** and click **Change**. Follow the steps in the **Microsoft SQL Server 2005 Maintenance** wizard.
2. From **Component Selection**, select **Workstation Components** and then click **Next**.
3. From **Welcome to the SQL Server Installation Wizard**, click **Next**.
4. From **System Configuration Check**, click **Next**.
5. From **Change or Remove Instance**, click **Change Installed Components**.
6. From **Feature Selection**, expand the **Documentation, Samples, and Sample Databases** node.
7. Select **Sample Code and Applications**.
8. Expand **Sample Databases** and then select the sample databases to be installed. Click **Next**.
9. To install and attach the sample databases, from **Sample Databases Setup**, select **Install and attach sample databases**, and then click **Next**.
The database files are created and stored in the folder <drive>:\Program Files\Microsoft SQL Server\MSSQL.n\MSSQL\Data. The database is attached and ready for use.
10. To install the sample database files without attaching, from **Sample Databases Setup**, select **Install sample databases without attaching**, and then click **Next**.
The **AdventureWorks** database files are created in the folder <drive>:\Program Files\Microsoft SQL Server\90\Tools\Samples\AdventureWorks OLTP. **AdventureWorksDW** files are created in the <drive>:\Program Files\Microsoft SQL

Server\90\Tools\Samples\AdventureWorks Data Warehouse folder. You must attach the database before you can use it. For more information, see **Attaching and Detaching a Database**.

11. Select the instance of SQL Server on which to install the sample databases and samples.
12. Complete the steps in the wizard.
13. To complete the installation of the samples, after setup, perform one of the following steps:
 1. From the **Start** menu, click **All Programs**, click **Microsoft SQL Server 2005**, click **Documentation and Tutorials**, click **Samples**, and then click **Microsoft SQL Server 2005 Samples**.
 2. Alternatively, using Windows Explorer, navigate to <drive>:\Program Files\Microsoft SQL Server\90\Tools\Samples\ and double-click **SqlServerSamples.msi** to launch the installer.

 **Important:**

The samples are installed to <drive>:\Program Files\Microsoft SQL Server\90\Samples. Be sure to read and follow the installation instructions in the Readme files that accompany the samples you want to use. For a list of samples, see **Samples**.

▣ **Deploying the Adventure Works DW Analysis Services project**

1. Make sure that the **AdventureWorksDW** and **Adventure Works DW** databases have been installed.
2. Make sure Analysis Services has been installed.
3. From the **SQL Server Business Intelligence Development Studio** toolbar, click **File**, point to **Open**, and then click **Project/Solution**.
4. Browse to <drive>:\Program Files\Microsoft SQL Server\90\Tools\Samples\AdventureWorks Analysis Services Project, select the file **Adventure Works.sln**, and then click **Open**.
5. From **Solution Explorer**, right-click **Adventure Works DW** and select **Deploy** or **Process**.

▣ Change History

Release	History
17 July 2006	New content: <ul style="list-style-type: none">• Added a link to the SQL Server 2005 Samples and Sample Databases download page.
5 December 2005	New content: <ul style="list-style-type: none">• Added SQL Server 2005 Express Edition and SQL Server 2005 Compact Edition note. Changed content: <ul style="list-style-type: none">• Corrected user interface references.

▣ **See Also**

Tasks

[Reinstalling Sample Databases from Script](#)

Concepts

[Downloading Northwind and pubs Sample Databases](#)

Other Resources

Installing Samples

[Installing AdventureWorks Sample Databases and Samples](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Reinstalling Sample Databases from Script



[See Also](#)



[Installing SQL Server 2005 Components](#) > [Installing AdventureWorks Sample Databases and Samples](#) >

If you have made changes to a sample database and want to return to the original schema, you can reinstall the database by running the appropriate installation script from Microsoft SQL Server Management Studio Query Editor. The installation script removes the existing sample database, re-creates the database objects, and populates the tables.

Alternatively, you can download the AdventureWorks database at the SQL Server Developer [Web site](#).

Note:

These procedures do not apply to SQL Server 2005 Express Edition or SQL Server 2005 Compact Edition. To download **AdventureWorks** for these editions, see [Installing Sample Databases for Express Editions](#).

The following procedures describe how to reinstall the **AdventureWorks** database and the **AdventureWorksDW** data warehouse.

To reinstall AdventureWorks from script

1. On the Management Studio toolbar, click **File**, point to **Open** and then click **File**.
2. Browse to the file **instawdb.sql** and click **Open**. The default location of this file is C:\Program Files\Microsoft SQL Server\90\Tools\Samples\AdventureWorks OLTP.
3. Before running the script, locate the statement `SET @data_path = @sql_path + 'AWDB\' ;` in the script and change the statement to point

to the location of the **instawdb.sql** script. For example, SET @data_path = 'C:\Program Files\Microsoft SQL Server\90\Tools\Samples\AdventureWorks OLTP\';.

4. Execute the script.

▣ **To reinstall AdventureWorksDW from script**

1. On the Management Studio tool bar, click **File**, point to **Open** and then click **File**.
2. Browse to the file **instawdwdb.sql** and click **Open**. The default location of this file is C:\Program Files\Microsoft SQL Server\90\Tools\Samples\AdventureWorks Data Warehouse.
3. Before running the script, locate the statement SET @data_path = @sql_path + 'awdwdb\'; in the script and change the statement to point to the location of the **instawdwdb.sql** script. For example, SET @data_path = 'C:\Program Files\Microsoft SQL Server\90\Tools\Samples\AdventureWorks Data Warehouse\';.
4. Execute the script.

▣ **See Also**

Concepts

[Running Setup to Install AdventureWorks Sample Databases and Samples](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Downloading Northwind and pubs Sample Databases



[See Also](#)



[Installing SQL Server 2005 Components](#) > [Installing AdventureWorks Sample Databases and Samples](#) >

The **Northwind** and **pubs** sample databases are not installed by default in Microsoft SQL Server 2005. These databases can be downloaded from this [Microsoft Web site](#).

After downloading **SQL2000SampleDb.msi**, extract the sample database scripts by double-clicking **SQL2000SampleDb.msi**. **SQL2000SampleDb.msi** will extract the database scripts and a Readme file into this default folder: C:\SQL Server 2000 Sample Databases. Follow the instructions in the Readme file for running the installation scripts.

Note:

The scripts for creating these databases have been updated to remove the guest account to improve security. The **pubs** installation script has been updated to include all of the data for the database, eliminating the requirement to run separate jobs to add data after creating **pubs**.

▣ **See Also**

Tasks

[Reinstalling Sample Databases from Script](#)

Concepts

[Running Setup to Install AdventureWorks Sample Databases and Samples](#)

Other Resources

AdventureWorks to Northwind Comparison

AdventureWorks to pubs Comparison

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Installing SQL Server Components How-to Topics



[See Also](#)



[Installing SQL Server 2005 Components](#) >

The topics in the following sections describe how to install components of Microsoft SQL Server 2005.

▣ In This Section

- [Database Engine Installation How-to Topics](#)
- [Analysis Services Installation How-to Topics](#)
- [Reporting Services Installation How-to Topics](#)
- [Notification Services Installation How-to Topics](#)
- [Integration Services Installation How-to Topics](#)
- [Installing SQL Server Replication](#)

▣ **See Also**

Tasks

[How to: Install SQL Server 2005 from the Command Prompt](#)

Reference

[Version and Edition Upgrades](#)

Concepts

[Preparing to Install SQL Server 2005](#)

[SQL Server Surface Area Configuration](#)

Other Resources

[Installing SQL Server Components How-to Topics](#)

[Upgrading to SQL Server 2005](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Database Engine Installation How-to Topics



[See Also](#)



[Installing SQL Server 2005 Components](#) > [Installing SQL Server Components How-to Topics](#) >

This section contains the following procedures used for installing, maintaining, and troubleshooting Microsoft SQL Server 2005.

▣ In This Section

Installation topics

- [How to: Install SQL Server 2005 \(Setup\)](#)
- [How to: Install SQL Server 2005 from the Command Prompt](#)
- [How to: Upgrade to SQL Server 2005 \(Setup\)](#)
- [How to: Install Microsoft Internet Information Services](#)
- [How to: Verify a Successful Installation of SQL Server 2005 Services](#)

Maintenance topics

- [How to: Change Components in an Existing Instance of SQL Server 2005 \(Setup\)](#)
- [How to: Rebuild the Master Database for SQL Server 2005](#)
- [How to: Rebuild the Registry for SQL Server 2005](#)
- [How to: Uninstall an Existing Instance of SQL Server 2005 \(Setup\)](#)

Troubleshooting topics

- [Troubleshooting an Installation of the SQL Server Database Engine](#)
- [Failover Cluster Troubleshooting](#)
- [How to: Work Around COM+ System Configuration Check Failure in SQL Server Setup](#)
- [How to: Increment the Counter Registry Key for Setup in SQL Server 2005](#)
- [How to: View SQL Server 2005 Setup Log Files](#)

▣ **See Also**

Tasks

[How to: Access Books Online for SQL Server 2005](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Install SQL Server 2005 (Setup)



[See Also](#)



[Installing SQL Server 2005 Components](#) > [Installing SQL Server Components How-to Topics](#) > [Database Engine Installation How-to Topics](#) >

Updated: **5 December 2005**

The Microsoft SQL Server 2005 Installation Wizard is Windows Installer-based, and provides a single feature tree for installation of all SQL Server 2005 components:

- Database Engine
- Analysis Services
- Reporting Services
- Notification Services
- Integration Services
- Replication
- Management Tools
- Connectivity Components
- Sample databases, samples, and SQL Server 2005 documentation

You do not have to install each component individually.

▣ Prerequisites

📌 Note:

If you are installing a failover cluster, do not follow this procedure. See [How to: Create a New SQL Server 2005 Failover Cluster \(Setup\)](#).

If you are upgrading a previous SQL Server version to SQL Server 2005, do not follow this procedure. See [Upgrading to SQL Server 2005](#).

Before you install SQL Server 2005, review topics in [Preparing to Install SQL Server 2005](#).

📌 Note:

For local installations, you must run Setup as an administrator. If you install SQL Server from a remote share, you must use a domain account that has read and execute permissions on the remote share.

▣ To install SQL Server 2005

1. To begin the installation process, insert the SQL Server 2005 DVD into your DVD drive. If the autorun feature on your DVD drive does not launch the installation program, navigate to the root of the DVD and launch **splash.hta**. If installing from a network share, navigate to the network folder and launch **splash.hta**.
2. From the autorun dialog, click **Run the SQL Server Installation Wizard**.
3. On the **End User License Agreement** page, read the license agreement, and then select the check box to accept the licensing terms and conditions. Accepting the license agreement activates the **Next** button. To continue, click **Next**. To end Setup, click **Cancel**.
4. On the **SQL Server Component Update** page, Setup installs software

required for SQL Server 2005. For more information about component requirements, click the **Help** button at the bottom of the page. To begin the component update process, click **Install**. To continue after the update completes, click **Finish**.

5. On the **Welcome** page of the SQL Server Installation Wizard, click **Next** to continue.
6. On the **System Configuration Check** (SCC) page, the installation computer is scanned for conditions that may block Setup. For information about configuration check items, click **Help** at the bottom of the page or see [Check Parameters for the System Configuration Checker](#). To interrupt the scan, click **Stop**. To display a list of check items grouped by result, click the **Filter** button and then select a category from the drop-down list. To view a report of SCC results, click the **Report** button and then select an option from the drop-down list. Options include viewing the report, saving the report to a file, copying the report to the Clipboard, and sending the report as e-mail. To proceed with Setup after the SCC scan completes, click **Continue**.
7. On the **Registration Information** page, enter information in the **Name** and **Company** text boxes. To continue, click **Next**.
8. On the **Components to Install** page, select the components for your installation. A description for each component group appears in the **Components to be Installed** pane when you select it. You can select any combination of check boxes. When you select SQL Server or Analysis Services, if Setup detects that you are installing to a virtual server, the **Install as a Virtual Server** check box is enabled. You must select this option to install a failover cluster.

 **Important:**

Because the steps are different for installing a failover cluster, see [How to: Create a New SQL Server 2005 Failover Cluster \(Setup\)](#) if you are installing a failover cluster.

To install individual components, click **Advanced**. Otherwise, click **Next** to continue.

9. If you clicked **Advanced** on the previous page, the **Feature Selection** page is displayed. On the **Feature Selection** page, select the program features to install using the drop-down boxes. To install components to a custom directory, select the feature and then click **Browse**. For more information about the functionality of this page, click **Help**. To continue when your feature selections are complete, click **Next**.
10. On the **Instance Name** page, select a default or named instance for your installation. If a default or named instance is already installed, and you select the existing instance for your installation, Setup upgrades it and provides you the option to install additional components. To install a new default instance, there must not be a default instance on the computer. To install a new named instance, click **Named Instance** and then type a unique instance name in the space provided. To install a new named instance side-by-side with an existing instance, click **Named Instance** and then type a unique instance name in the space provided. For more information about instance naming rules, click **Help** at the bottom of the page, or see the [Instance Name](#) topic in SQL Server 2005 Books Online.
11. On the **Service Account** page, specify the user name, password, and domain name for SQL Server service accounts. You can use one account for all of the services.

Optionally, you can specify an individual account for each service. To specify an individual account for each service, select **Customize for each service account**, select a service name from the drop-down box, and then provide login credentials for the service. To proceed, click **Next**.

 **Note:**

The domain name cannot be a full Domain Name System (DNS) name. For example, if your DNS name is my-domain-name.com, use **my-domain-name** in the **Domain** field. SQL Server does not accept **my-domain-name.com** in the **Domain** field.

12. On the **Authentication Mode** page, choose the authentication mode to use for your SQL Server installation.

If you select Windows Authentication, Setup creates an **sa** account, which is disabled by default. Enter and confirm the system administrator (sa) login when you choose Mixed Mode Authentication. Passwords are the first line of defense against intruders, so setting strong passwords is essential to the security of your system. Never set a blank or weak sa password. To utilize Mixed Mode Authentication and activate the **sa** account after Setup finishes, see "How to: Change Server Authentication Mode" in SQL Server Books Online. To proceed, click **Next**.

 **Note:**

When possible, use Windows Authentication. Do not use a blank password. Use a strong password. For strong password guidelines, see [Authentication Mode](#).

 **Note:**

For this release, SQL Server Setup may not enforce the strong password requirement on some default configurations of Microsoft Windows Server 2003 where the computer is not a member of a domain. On the **Collation Settings** page, specify the collation for your SQL Server instance. You can use one account for SQL Server and Analysis Services, or you can specify collations for individual components.

To set separate collation settings for SQL Server and Analysis Services, select the **Customize for each service account** check box. After you select the check box, a drop-down selection box appears. Select a service from the drop-down selection box and then assign its collation. Repeat for each service. To proceed, click **Next**.

13. If you selected Reporting Services as a feature to install, the **Report Server Installation Options** page is displayed. Use the radio buttons to select whether to configure the report server with default values. If the requirements for installing Reporting Services in the default configuration are not satisfied, you must choose the **Install but do not configure the server** installation option. To view installation details for this page, click the **Details** button. For more information, click **Help** at

the bottom of the page. To proceed, click **Next**.

14. On the **Error Reporting** page, optionally clear the check box to disable error reporting. For more information about error reporting functionality, click **Help** at the bottom of the page. To proceed, click **Next**.
15. On the **Ready to Install** page, review the summary of features and components for your SQL Server installation. To proceed, click **Install**.
16. On the **Installation Progress** page, you can monitor installation progress as Setup proceeds. To view the log file for a component during installation, click the product or status name on the **Installation Progress** page.
17. On the **Completing the Microsoft SQL Server Installation Wizard** page, you can view the Setup summary log by clicking the link provided on this page. To exit the SQL Server Installation Wizard, click **Finish**.
18. If you are instructed to restart the computer, do so now. It is important to read the message from the Setup program when you are done with installation. Failure to restart the computer may cause failures when you run the Setup program in the future.

▣ **Next Steps**

After Setup completes the installation of SQL Server 2005, you can verify that the installation was successful. For more information, see [How to: Verify a Successful Installation of SQL Server 2005 Services](#).

After Setup completes the installation of SQL Server 2005, you can further configure SQL Server using graphical tools and command prompt utilities. For more information, see [Configuring a SQL Server Installation](#).

To reduce the attackable surface area of your system, SQL Server 2005 selectively installs and activates key services and features. In the default configuration of new installations, many features are disabled. For more information, see [SQL Server Surface Area Configuration](#).

▣ **See Also**

Tasks

[How to: Upgrade to SQL Server 2005 \(Setup\)](#)

[How to: Create a New SQL Server 2005 Failover Cluster \(Setup\)](#)

[How to: Upgrade to a SQL Server 2005 Failover Cluster \(Setup\)](#)

Concepts

[Preparing to Install SQL Server 2005](#)

[Failover Cluster Troubleshooting](#)

[How to: View SQL Server 2005 Setup Log Files](#)

Other Resources

How to: Upgrade to SQL Server 2005 with the Copy Database Wizard

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Work Around COM+ System Configuration Check Failure in SQL Server Setup

 [See Also](#)



[Installing SQL Server 2005 Components](#) > [Installing SQL Server Components How-to Topics](#) > [Database Engine Installation How-to Topics](#) >

Microsoft SQL Server 2005 Setup checks that COM+ is properly configured. If configuration errors are found, Setup will continue, but the following warning appears in the System Configuration Check (SCC) report:

"If SQL Server Setup fails, Setup will roll back the installation but may not remove all .manifest files. The workaround is to rename the files and then rerun Setup."

COM+ errors can result when Microsoft Distributed Transaction Coordinator (MS DTC) is not running or, in the case of Microsoft Cluster Server, if MS DTC is not a clustered resource. COM+ depends upon MS DTC, and the Message Queue Task in Integration Services depends upon COM+. If COM+ errors occur, the Message Queue Task in Integration Services will not be available until the COM+ system is properly configured.

To use Message Queuing (also known as MSMQ), ensure that MS DTC is running and properly configured. If SQL Server is installed on a cluster, MS DTC must be a cluster resource.

Use the following procedures to reinstall COM+.

To install the Component Services snap-in

1. From the Windows desktop, click **Start** and then click **Run**.
2. In **Open**, type **MMC**, and then click **OK**.

3. In the Console window, click **File** on the menu bar, and then click **Add/Remove Snap-in**.
4. In the Add/Remove Snap-in window, click **Add**.
5. In the Add Standalone Snap-in window, select **Component Services** from the list of snap-ins, and then click **Add**.
6. Click **Close** to close the Add Stand-alone Snap-in window, and then click **OK** to close the Add/Remove Snap-in window.
7. In the Console Root\Component Services window, expand the **Component Services** tree. This is where any error message would occur if there was a problem with COM+.
8. Run SQL Server 2005 Setup again. If you get an error message, reinstall COM+.

▣ **To reinstall COM+**

1. From Add or Remove Programs in Control Panel, click **Add/Remove Windows Components**.
2. In the Windows Components Wizard, click **Next** without making any selection changes.
3. Click through to complete the wizard, and then run SQL Server 2005 Setup again.

▣ **See Also**

Tasks

[How to: Read a SQL Server 2005 Setup Log File](#)

Concepts

[Troubleshooting an Installation of the SQL Server Database Engine](#)

[How to: View SQL Server 2005 Setup Log Files](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Increment the Counter Registry Key for Setup in SQL Server 2005

 [See Also](#)



[Installing SQL Server 2005 Components](#) > [Installing SQL Server Components How-to Topics](#) > [Database Engine Installation How-to Topics](#) >

Setup Configuration Checker (SCC) in Microsoft SQL Server Setup verifies the value of the counter registry key before SQL Server installation begins. If SCC cannot verify the existing registry key, or if SCC cannot run the lodctr.exe system program, the SCC check will fail, and Setup will be blocked.

Incorrectly editing the registry can severely damage your system. Before making changes to the registry, we recommend that you back up any valued data on the computer.

To manually set the increment for the counter registry key

1. On the Microsoft Windows 2003 or Windows XP desktop, click **Start**, click **Run**, type **regedit.exe** in **Open**, and then click **OK**. On Windows 2000, use **regedt32.exe** The Registry Editor launches.

2. Navigate to the following registry key:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows  
NT\CurrentVersion\Perflib]
```

```
"Last Counter"=dword:00000ed4 (5276)
```

```
"LastHelp"=dword:00000ed5 (5277)
```

3. The "Last Counter" value from the previous step (5276) must match the maximum value of the "Counter" key from "Perflib\009" in the following registry key, and the "Last Help" value from the previous step

(5277) must match the maximum value of the "Help" key from "Perflib\009" in the following registry key:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib\009]
```

Note that 009 is an example from the English language. The "Last Counter" and "Last Help" values are dynamically assigned by Windows; they will vary from machine to machine.

4. If necessary, modify the value for the "Last Counter" and "Last Help" values in the "\Perflib" key: right-click **Last Counter** or **Last Help** in the right-hand pane, click **Modify**, click **Base = "Decimal,"** set the value in **Value data**, and then click **OK**. Repeat for the other key, if necessary, and then close the Registry Editor.
5. Run SQL Server Setup again.

▣ **See Also**

Concepts

[Preparing to Install SQL Server 2005](#)

[How to: View SQL Server 2005 Setup Log Files](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Install SQL Server 2005 from the Command Prompt



[See Also](#)



[Installing SQL Server 2005 Components](#) > [Installing SQL Server Components How-to Topics](#) > [Database Engine Installation How-to Topics](#) >

Updated: 12 December 2006

Before you run Setup for SQL Server 2005, review [Preparing to Install SQL Server 2005](#). If you are installing a failover cluster, also review [Before Installing Failover Clustering](#).

When you install SQL Server 2005 from the command prompt, you can specify Setup parameters at the command prompt, or you can use the **/settings** switch to load Setup parameters from an .ini file. You must use either Setup parameters at the command prompt, or an .ini file, but you cannot use both.

To specify Setup parameters by loading them from an .ini file, use the **/settings filename.ini** argument to specify the .ini file. The **template.ini** file on the SQL Server 2005 installation media provides detailed instructions for creating an .ini file. Note in particular that the first noncommented line in the .ini file must contain the string **[Options]**. The following example illustrates the format of an .ini file:

```
[Options]
USERNAME=MyName
COMPANYNAME=MyCompany
PIDKEY=ABCDE12345FGHIJ67890KLMNO
INSTALLSQLDIR="C:\Program Files\Microsoft SQL Server\'
INSTALLLOLAPDATADIR="C:\Program Files\Microsoft SQL Ser
ADDLOCAL=SQL_Engine,Analysis_Server,RS_Server...
INSTANCENAME=MyInstance
...
```

For more information about specifying SQL Server 2005 Setup parameters by using an .ini file, see the template.ini file in the root folder of the SQL Server 2005 installation media.

 **Note:**

For local installations, you must run Setup as an administrator. If you install SQL Server from a remote share, you must use a domain account that has read and execute permissions on the remote share. For failover cluster installations, you must be a local administrator with permissions to logon as a service, and to act as part of the operating system on all nodes of the virtual server.

Use the following sample scripts to install, update, or remove SQL Server 2005 components from the command prompt. The scripts provide command prompt syntax to install SQL Server 2005 in typical user scenarios. Test and modify samples to meet the needs of your organization.

Sample scripts are divided into the following categories:

- [Installation scripts](#)
- [Maintenance scripts](#)
- [Uninstall scripts](#)

For detailed information about specific parameters for installing from command prompt, see [Parameters](#).

▣ Installation Scripts

Test and modify the following installation scripts to meet the needs of your organization.

▣ To install a new, stand-alone instance with all SQL Server 2005 components

1. Insert the SQL Server 2005 installation media into the disk drive.
2. Run the following command from the command prompt:

```
Start /wait <CD or DVD Drive>\servers\setup.exe /qb INSTANCENAME
```

☑ **Important:**

Do not use a blank password. Use a strong password.

☑ **Note:**

For the default instance, specify "MSSQLSERVER" for <InstanceName>. For a named instance, specify the instance name. For instance naming guidelines, see [Instance Name](#).

☑ **Important:**

Installations from CD media are split into \servers and \tools folders. Installations from CD must specify the \servers or \tools folder in the path to Setup.exe. Installations from DVD media are on a single disk. Installations from DVD should not include the \servers or \tools folder in the path to Setup.exe.

The /qn switch suppresses all Setup dialog boxes and error messages. If the /qn switch is specified, all Setup messages, including error messages, are written to Setup log files. For more information about log files, see [How to: View SQL Server 2005 Setup Log Files9d77af64-9084-4375-908a-](#)

d90f99535062.

The /qb switch allows the display of basic Setup dialog boxes. Error messages are also displayed.

▣ **To install a new, stand-alone instance with instance-unaware components of SQL Server 2005 - Notification Services, Integration Services, Full-Text Search, SQLXML, and Replication**

1. Insert the SQL Server 2005 installation media into the disk drive.
2. Run the following command from the command prompt:

Start /wait <CD or DVD Drive>\servers\setup.exe /qb ADDLOCAL=Cl

☑ **Important:**

Do not use a blank password. Use a strong password.

☑ **Note:**

For the default instance, specify "MSSQLSERVER" for <InstanceName>. For a named instance, specify the instance name. For instance naming guidelines, see [Instance Name](#).

☑ **Important:**

Installations from CD media are split into \servers and \tools folders. Installations from CD must specify the \servers or \tools folder in the path to Setup.exe. Installations from DVD media are on a single disk. Installations from DVD should not include the \servers or \tools folder in the path to Setup.exe.

The /qn switch suppresses all Setup dialog boxes and error messages. If the /qn switch is specified, all Setup messages, including error messages, are written to Setup log files. For more information about log files, see [How to: View SQL Server 2005 Setup Log Files](#).

The /qb switch allows the display of basic Setup dialog boxes. Error messages are also displayed.

☐ **To install SQL Server 2005 Client Tools only**

1. Insert the SQL Server 2005 installation media into the disk drive.
2. Run the following command from the command prompt:

```
Start /wait <CD or DVD Drive>\servers\setup.exe /qb INSTANCENAME
```

☑ **Important:**

Do not use a blank password. Use a strong password.

☑ **Note:**

For the default instance, specify "MSSQLSERVER" for <InstanceName>. For a named instance, specify the instance name. For instance naming guidelines, see [Instance Name](#).

☑ **Important:**

Installations from CD media are split into \servers and \tools folders. Installations from CD must specify the \servers or \tools folder in the path to Setup.exe. Installations from DVD media are on a single disk. Installations from DVD should not include the \servers or \tools folder in the path to Setup.exe.

The /qn switch suppresses all Setup dialog boxes and error messages. If the /qn switch is specified, all Setup messages, including error messages, are written to Setup log files. For more information about log files, see [How to: View SQL Server 2005 Setup Log Files](#).

☐ **To install SQL Server 2005 connectivity and SQLXML components**

1. Insert the SQL Server 2005 installation media into the disk drive.
2. Run the following command from the command prompt:

```
start /wait <CD or DVD Drive>\servers\setup.exe /qb ADDLOCAL=cli
```

☑ **Important:**

Do not use a blank password. Use a strong password.

☑ **Note:**

For the default instance, specify "MSSQLSERVER" for <InstanceName>. For a named instance, specify the instance name. For instance naming guidelines, see [Instance Name](#).

☑ **Important:**

Installations from CD media are split into \servers and \tools folders. Installations from CD must specify the \servers or \tools folder in the path to Setup.exe. Installations from DVD media are on a single disk. Installations from DVD should not include the \servers or \tools folder in the path to Setup.exe.

The /qn switch suppresses all Setup dialog boxes and error messages. If the /qn switch is specified, all Setup messages, including error messages, are written to Setup log files. For more information about log files, see [How to: View SQL Server 2005 Setup Log Files](#).

☐ **To upgrade a SQL Server instance to SQL Server 2005**

1. Insert the SQL Server 2005 installation media into the disk drive.
2. Run the following command from the command prompt:

```
start /wait setup.exe /qn INSTANCENAME=<Instancename> UPGRADE=SQ  
SAPWD=<strongpassword> SQLACCOUNT=<domain\user> SQLPASSWORD=<dc  
SQLBROWSERACCOUNT=<domain\user> SQLBROWSERPASSWORD=<domainuserpa
```

 **Important:**

Do not use a blank password. Use a strong password.

 **Note:**

For the default instance, specify "MSSQLSERVER" for <InstanceName>. For a named instance, specify the instance name. For instance naming guidelines, see [Instance Name](#).

 **Important:**

Installations from CD media are split into \servers and \tools folders. Installations from CD must specify the \servers or \tools folder in the path to Setup.exe. Installations from DVD media are on a single disk. Installations from DVD should not include the \servers or \tools folder in the path to Setup.exe.

The /qn switch suppresses all Setup dialog boxes and error messages. If the /qn switch is specified, all Setup messages, including error messages, are written to Setup log files. For more information about log files, see [How to: View SQL Server 2005 Setup Log Files](#).

To install a new, clustered instance of SQL Server 2005 from the command prompt

1. Insert the SQL Server 2005 installation media into the disk drive.

2. For a new failover cluster, use the following syntax to install SQL Server Database Services and management tools:

```
start /wait <CD or DVD Drive>\servers\setup.exe /qn VS=<VSName>
```

 **Important:**

Do not use a blank password. Use a strong password.

The /qn switch suppresses all Setup dialog boxes and error messages. If the /qn switch is specified, all Setup messages, including error messages, are written to Setup log files. For more information about log files, see [How to: View SQL Server 2005 Setup Log Files](#).

The /qb switch allows the display of basic Setup dialog boxes. Error messages are also displayed.

To upgrade a SQL Server failover cluster to SQL Server 2005

1. Insert the SQL Server 2005 installation media into the disk drive.
2. Run the following command from the command prompt:

```
start /wait setup.exe /qn VS=<VSNAME> INSTALLVS=SQL_Engine INSTA/  
ADDNODE=<NodeName1,NodeName2> GROUP=<Diskgroup> ADMINPASSWORD=<S  
SQLACCOUNT=<domain\user> SQLPASSWORD=<domainuserpassword> AGTACC  
SQLBROWSERACCOUNT=<domain\user> SQLBROWSERPASSWORD=<domainuserpa  
SQLCLUSTERGROUP="MYDOMIAN\MYSQLDOMAINGROUP" AGTCLUSTERGROUP="MY  
FTSCLUSTERGROUP="MYDOMIAN\MYFTSDOMAINGROUP"
```

 **Important:**

Do not use a blank password. Use a strong password.

 **Note:**

For the default instance, specify "MSSQLSERVER" for <InstanceName>. For a named instance, specify the instance name. For instance naming guidelines,

see [Instance Name](#).

 **Important:**

Installations from CD media are split into \servers and \tools folders. Installations from CD must specify the \servers or \tools folder in the path to Setup.exe. Installations from DVD media are on a single disk. Installations from DVD should not include the \servers or \tools folder in the path to Setup.exe.

The /qn switch suppresses all Setup dialog boxes and error messages. If the /qn switch is specified, all Setup messages, including error messages, are written to Setup log files. For more information about log files, see [How to: View SQL Server 2005 Setup Log Files](#).

Return to [top](#).

Maintenance Scripts

Test and modify the following maintenance scripts to meet the needs of your organization.

To add components to an existing, stand-alone instance of SQL Server 2005 from the command prompt

1. Insert the SQL Server 2005 installation media into the disk drive.
2. To add Full-Text Search and remove Replication, use the following syntax:

```
Start /wait <CD or DVD Drive>setup.exe /qb INSTANCENAME=<Instanc
```

For the default instance, specify "MSSQLSERVER" for <InstanceName>. For a complete list of parameters, see [Parameters](#) below.

☐ **To add components to an existing, SQL Server 2005 cluster from the command prompt**

1. Insert the SQL Server 2005 installation media into the disk drive.
2. To add Analysis Services to an existing cluster, use the following syntax:

```
start /wait <CD or DVD Drive>\setup.exe /qn VS=<VSName> INSTALL
```

☑ **Note:**

You cannot install Analysis Services to the same cluster group as the Database Engine. You must install Analysis Services to its own group and then, after installation, you can move Analysis Services to the same group as SQL Server.

For a complete list of parameters, see [Parameters](#) below.

☐ **To add nodes to an existing SQL Server 2005 cluster from the command prompt**

1. Insert the SQL Server 2005 installation media into the disk drive.
2. Use the following syntax:

```
start /wait <CD or DVD Drive>\setup.exe /qn VS=<VSName> INSTALL
```

☑ **Important:**

Do not use a blank password. Use a strong password.

For the default instance, specify "MSSQLSERVER" for *<InstanceName>*.

The /qn switch suppresses all Setup dialog boxes and error messages. If the /qn switch is specified, all Setup messages, including error messages, are

written to Setup log files. For more information about log files, see [How to: View SQL Server 2005 Setup Log Files](#). The /qb switch allows the display of basic Setup dialog boxes. Error messages are also displayed.

▣ **To remove nodes from an existing SQL Server 2005 cluster from the command prompt**

1. Insert the SQL Server 2005 installation media into the disk drive.
2. Use the following syntax:

```
start /wait <CD or DVD Drive>\setup.exe /qn VS=<VSName> INSTALL
```

Important:

Do not use a blank password. Use a strong password.

For the default instance, specify "MSSQLSERVER" for <InstanceName>.

The /qn switch suppresses all Setup dialog boxes and error messages. If the /qn switch is specified, all Setup messages, including error messages, are written to Setup log files. For more information about log files, see [How to: View SQL Server 2005 Setup Log Files](#). The /qb switch allows display of basic Setup dialog boxes. Error messages are also displayed.

Caution:

If Microsoft SQL Server 2000 and SQL Server 2005 are installed side-by-side in a clustered environment, Microsoft SQL Native Client should not be uninstalled if you uninstall SQL Server 2005. The SQL Server Resource DLL from SQL Server 2005 always uses SQL Native Client to connect to SQL Server. When SQL Server 2005 is installed, the SQL Server Cluster Resource DLL is upgraded. When SQL Server 2005 is uninstalled, the SQL Server Resource DLL is not replaced with the version required by SQL Server 2000. Leaving SQL Native Client on cluster nodes allows the SQL Server Resource DLL to continue to connect to instances of SQL Server 2000.

Rebuilding System Databases, Rebuilding the Registry

REINSTALLMODE is used to repair installed SQL Server 2005 components. For additional information about use of REINSTALLMODE, see the [\[REINSTALLMODE\]](#) and [\[REBUILDDATABASE\]](#) sections below.

▣ To rebuild system databases for a default instance of SQL Server 2005 from the command prompt

1. Insert the SQL Server 2005 installation media into the disk drive.
2. Use the following syntax:

```
start /wait <CD or DVD Drive>\setup.exe /qn INSTANCENAME=<Instar
```

📌 **Important:**

Do not use a blank password. Use a strong password.

For the default instance, specify "MSSQLSERVER" for <InstanceName>.

Important The **resource** database (**mssqlsystemresource**) is restored from SQL Server 2005 distribution media. When you rebuild the **resource** database from distribution media, all Service Packs and QFE updates are lost, and therefore must be reapplied. Before you proceed, see [\[REBUILDDATABASE\]](#) for more information.

The /qn switch suppresses all Setup dialog boxes and error messages. If the /qn switch is specified, all Setup messages, including error messages, are written to Setup log files. For more information about log files, see [How to: View SQL Server 2005 Setup Log Files](#). The /qb switch allows display of basic Setup dialog boxes. Error messages are also displayed.

Rebuilding the **master** database installs all system databases to their initial location. If you have moved one or more system databases to a different

location, you must move the databases again. For more information about moving system databases, see **Moving System Databases**.

☐ **To rebuild system databases for a clustered instance of SQL Server 2005 from the command prompt**

1. Insert the SQL Server 2005 installation media into the disk drive.
2. Use the following syntax:

```
start /wait <CD or DVD Drive>\setup.exe /qn VS=<VSName> INSTANCE
```

Important:

Do not use a blank password. Use a strong password.

For the default instance, specify "MSSQLSERVER" for *<InstanceName>*.

Important The **resource** database (**mssqlsystemresource**) is restored from SQL Server 2005 distribution media. When you rebuild the **resource** database from distribution media, all Service Packs and QFE updates are lost, and therefore must be reapplied. Before you proceed, see [\[REBUILDDATABASE\]](#) for more information.

The /qn switch suppresses all Setup dialog boxes and error messages. If the /qn switch is specified, all Setup messages, including error messages, are written to Setup log files. For more information about log files, see [How to: View SQL Server 2005 Setup Log Files](#). The /qb switch allows display of basic Setup dialog boxes. Error messages are also displayed.

Rebuilding the **master** database installs all system databases to their initial location. If you have moved one or more system databases to a different location, you must move the databases again. For more information about moving system databases, see **Moving System Databases**.

Specifying a New System Collation

Collation settings, which include character set, sort order, and other locale-specific settings, are fundamental to the structure and function of all SQL Server databases. You should develop a standard within your organization for these options, and apply them at the time of installation. Many server-to-server activities can fail if collation settings are not consistent across servers.

When installing SQL Server 2005, use the correct collation settings. You can change collation settings after running Setup, but you must rebuild the system and user databases and reload user data if you do.

To change collation settings for an existing instance of SQL Server 2005, you must use unattended Setup to rebuild the system databases. This functionality is exposed using a special switch called `REBUILD_DATABASE=1`. When system databases are rebuilt, all database objects and data in **master**, **model** and **msdb** system databases are removed. Rebuilding the **master** database installs all system databases to their initial location. If you have moved one or more system databases to a different location, you must move the databases again. For more information about moving system databases, see **Moving System Databases**.

Your syntax must specify either the `/qb` or `/qn` option. The `/qb` option displays basic Setup user interface and error message dialog boxes. If the `/qn` option is specified, all Setup dialog boxes, including error messages, are written to Setup log files.

Important:

In order to rebuild system databases, a new System Administrator (**sa**) password is REQUIRED. Do not use a blank password. Use a strong password.

To rebuild system databases and specify a new system collation

1. Insert the SQL Server 2005 installation media into the disk drive.
2. Run the following command from the command prompt:

```
start /wait <CD or DVD Drive>\setup.exe /qn INSTANCENAME=MSSQLSE
```

For INSTANCENAME, use **MSSQLSERVER** for a default instance. For a named instance, specify the instance name.

 **Important:**

The **resource** database (**mssqlsystemresource**) is restored from SQL Server 2005 distribution media. When you rebuild the **resource** database from distribution media, all service packs and hotfix updates are lost, and therefore must be reapplied. Before you proceed, see the [\[REBUILDDATABASE\]](#) section for more information.

The /qn switch suppresses all Setup dialog boxes and error messages. If the /qn switch is specified, all Setup messages, including error messages, are written to Setup log files. For more information about log files, see [How to: View SQL Server 2005 Setup Log Files](#). The /qb switch allows display of basic Setup dialog boxes. Error messages are also displayed.

Rebuilding the **master** database installs all system databases to their initial location. If you have moved one or more system databases to a different location, you must move the databases again. For more information about moving system databases, see **Moving System Databases**.

Rebuilding the Registry

To rebuild the SQL Server 2005 registry, you must use the Setup command with the REINSTALL=**ALL** and REINSTALLMODE=**omus** parameters. Running Setup.exe with these parameter settings rebuild, verify, and repair a SQL Server instance and rebuild the registry. This procedure is most often used to rebuild the registry for a damaged SQL Server installation.

 **Note:**

To rebuild the registry, you must use the same package file and options that you specified during the initial installation. If you do not know this information, uninstall and then reinstall SQL Server, rather than rebuild the registry.

Your syntax must specify either the /qb or /qn option. The /qb option displays basic Setup user interface and error message dialog boxes. If the /qn option is specified, all Setup dialog boxes, including error messages, are written to Setup log files.

 **Important:**

In order to rebuild system databases, a new **sa** password is REQUIRED. Do not use a blank password. Use a strong password.

▣ **To repair all files, rebuild the registry, and replace all SQL Server shortcuts from the command prompt**

1. Insert the SQL Server 2005 installation media into the disk drive.
2. Run the following command from the command prompt:

```
start /wait <CD or DVD Drive>\setup.exe /qb INSTANCENAME=MSSQLSE
```

For INSTANCENAME, use MSSQLSERVER for a default instance. For a named instance, specify the instance name.

 **Important:**

The **resource** database (**mssqlsystemresource**) is restored from SQL Server 2005 distribution media. When you rebuild the **resource** database from distribution media, all Service Packs and QFE updates are lost, and therefore must be reapplied. Before you proceed, see the [\[REBUILDDATABASE\]](#) section more information.

▣ **To repair SQL Server Management Studio shortcuts from the command prompt**

1. Insert the SQL Server 2005 installation media into the disk drive.

2. Run the following command from the command prompt:

```
start /wait <CD or DVD Drive>\setup.exe /qb REINSTALL=SQL_Tools9
```

For INSTANCENAME, use **MSSQLSERVER** for a default instance. For a named instance, specify the instance name.

▣ **To repair Full-Text Search Engine registry keys from the command prompt**

1. Insert the SQL Server 2005 installation media into the disk drive.
2. Run the following command from the command prompt:

```
start /wait <CD or DVD Drive>\setup.exe /qb REINSTALL=SQL_FullTe
```

For INSTANCENAME, use **MSSQLSERVER** for a default instance. For a named instance, specify the instance name.

Return to [top](#).

Uninstall Scripts

Test and modify the following removal scripts to meet the needs of your organization.

On a computer running SQL Server 2005 along with previous SQL Server versions, Enterprise Manager and other programs that depend on SQL-DMO may be disabled. This may occur in the following situations:

- Side-by-side installations of SQL Server 2005, SQL Server 2000, and/or SQL Server 7.0, where any of the versions is uninstalled.
- Side-by-side installations of SQL Server 2000 with SQL Server 2005,

where SQL Server 2000 is installed after SQL Server 2005.

This issue is due to removal of the registration for the SQL Server 2005 SQL-DMO COM library. To re-enable Enterprise Manager and other programs with SQL-DMO dependencies, register SQL-DMO by running regsvr32.exe sqldmo.dll from a command prompt. For more information, see [Troubleshooting an Installation of the SQL Server Database Engine](#).

▣ **To remove a default, stand-alone installation of SQL Server 2005 from the command prompt**

1. Insert the SQL Server 2005 installation media into the disk drive.
2. Uninstall SQL Server, using the following syntax:

```
Start /wait <CD or DVD Drive>\setup.exe /qb REMOVE=SQL_Engine,C]
```

For the default instance, specify "MSSQLSERVER" for <InstanceName>. Using **REMOVE=ALL** to remove instance components also removes shared components: SQL Server Tools, Integration Services, and Notification Services.

3. Uninstall Microsoft SQL Native Client, using the following syntax:

```
Start /wait C:\Windows\System32\msiexec /qb /X <CD or DVD Drive>
```

4. Uninstall Microsoft .NET Framework, using the following syntax:

```
Start /wait <CD or DVD Drive>\redist\2.0\dotnetfix.exe /q:a /c:'
```

Caution:

Uninstalling .NET Framework 1.1 or 2.0 affects existing Microsoft Visual Studio installations, as Visual Studio depends on the Microsoft .NET Framework. If you have Visual Studio installed, do not uninstall the .NET Framework.

▣ To remove a default clustered instance of SQL Server 2005 from the command prompt

1. Insert the SQL Server 2005 installation media into the disk drive.
2. Uninstall SQL Server using the following syntax:

```
start /wait <CD or DVD Drive>\setup.exe /qn VS=<VSName> INSTANCE
```

For the default instance, specify "MSSQLSERVER" for *<InstanceName>*.

The /qn switch suppresses all Setup dialog boxes and error messages. If the /qn switch is specified, all Setup messages, including error messages, are written to Setup log files. For more information about log files, see [How to: View SQL Server 2005 Setup Log Files](#). The /qb switch allows display of basic Setup dialog boxes. Error messages are also displayed.

3. Uninstall SQL Native Client, using the following syntax:

```
Start /wait <CD or DVD Drive>\msiexec /qb /X <CD or DVD Drive>\s
```

If SQL Server Setup fails to uninstall the cluster node, use the following steps:

4. Use the Registry Editor utility, regedit.exe, locate the registry key HLKM\Software\Microsoft\Microsoft SQL Server*<instid>*\Setup, where *<instid>* is the specific SQL Server instance being uninstalled; for example, MSSQL.1 for the first instance and MSSQL.2 for the second instance.

Caution:

Incorrectly editing the registry can cause serious problems that may require you to reinstall your operating system. Microsoft cannot guarantee that problems resulting from editing the registry incorrectly can be resolved. Before editing the registry, back up any valuable data. For information about how to back up, restore, and edit the registry, see this

[Microsoft Knowledge Base article.](#)

5. In this key, set **SqlCluster=2**.
6. Repeat the previous 2 steps to update the **SqlCluster** key on all nodes in the cluster.
7. Move the shared disk to the computer that is currently being uninstalled and, on that node, use **Add or Remove Programs** to remove SQL Server 2005.
8. Move the shared disk to the next node, and use **Add or Remove Programs** to remove SQL Server 2005.
9. Repeat this move-and-uninstall process for each remaining node.

If SQL Server 2005 is running along with previous SQL Server versions, Enterprise Manager, or other programs that depend on SQL-DMO, the previous versions may be disabled. This may occur in the following situations:

- Side-by-side installations of SQL Server 2005, SQL Server 2000, and/or SQL Server 7.0, where any of the versions are uninstalled.
- Side-by-side installations of SQL Server 2000 with SQL Server 2005, where SQL Server 2000 is installed after SQL Server 2005.

This issue is due to removal of the registration for the SQL Server 2005 SQL-DMO COM library. To re-enable Enterprise Manager and other programs with SQL-DMO dependencies, register SQL-DMO by running **regsvr32.exe sqldmo.dll** from a command prompt. For more information, see [Troubleshooting an Installation of the SQL Server Database Engine](#).

If SQL Server 2000 and SQL Server 2005 are installed side-by-side in a clustered environment, SQL Native Client should not be uninstalled if you uninstall SQL Server 2005. The SQL Server Resource DLL from SQL Server 2005 always uses SQL Native Client to connect to SQL Server. When SQL Server 2005 is installed, the SQL Server Cluster Resource DLL is upgraded. When SQL Server 2005 is uninstalled, the SQL Server Resource DLL is not replaced with the version required by SQL Server

2000. Leaving SQL Native Client on cluster nodes allows the SQL Server Resource DLL to continue to connect to instances of SQL Server 2000.

▣ **To run a command prompt installation of SQL Server 2005 using an .ini file**

1. Run Setup.exe from the command prompt:

An example command for an installation that references an .ini file would look like this:

```
<CD or DVD Drive>\Setup.exe /settings C:/set
```

where the Setup.exe command is followed by the **/settings** switch that specifies the name of an .ini file containing installation parameters, followed by the /qn switch to suppress the Setup user interface. In this example, the "C:\set.ini" file is a text file that contains the installation parameters.

If the /qn switch is specified, all Setup messages, including error messages, are written to Setup log files. For more information about parameters for an .ini file, refer to the content that follows this section. For more information about Setup log file locations, see [How to: View SQL Server 2005 Setup Log Files](#).

Return to [top](#).

Parameters

You can use the parameters listed in the table below to develop command prompt installation scripts.

[Options]	[PIDKEY]	[INSTALLSQL]
[INSTALLSQLSHAREDDIR]	[INSTALLSQLDATADIR]	[INSTALLASD]

[ADDLOCAL]	[REMOVE]	[INSTANCENAME]
[UPGRADE]	[SAVESYSDB]	[USESYSDB]
[SQLACCOUNT] , [SQLPASSWORD] , [AGTACCOUNT] , [AGTPASSWORD] , [ASACCOUNT] , [ASPASSWORD] , [RSACCOUNT] , and [RSPASSWORD]	[AUTOSTART]	[SECURITYMODE]
[SQLCOLLATION] and [ASCOLLATION]	[REBUILDDATABASE]	[REINSTALLMIRROR]
[RSCONFIGURATION]	[SAMPLEDATABASESERVER]	[DISABLENETBIOS]
[ERRORREPORTING]	[SKUUPGRADE]	

For detailed information about each parameter, see the parameter descriptions below.

To install from a command prompt on a cluster, see [Parameters Required for Cluster Installation](#).

Return to [top](#).

[Options]

The **[Options]** line is REQUIRED when using a settings (.ini) file. This is information for the **Registration Information** dialog box.

Default: If these parameters are not specified, the default user name and company name values for the server are used.

 **Note:**

If names contain spaces, surround the names with quotes.

 [Copy Code](#)

```
USERNAME=
COMPANYNAME=
```

[PIDKEY]

For the SQL Server 2005 milestones, the PIDKEY property is not required, because the parameter for the MSI file is preset. However, the final released version of the SQL Server 2005 requires the PIDKEY property.

Usage: PIDKEY=ABCDE12345FGHIJ67890KLMNO.

 **Note:**

This is not an actual key value.

Notes

- PIDKEY cannot be used for SQL Server Express installations.
- Do not include "-" in the PIDKEY.

 [Copy Code](#)

PIDKEY=

[INSTALLSQLDIR]

This is information for the **Feature Selection** dialog box. To use the default paths, do not specify this parameter.

 **Note:**

If there is a space in a specified path, surround the path with quotes, and end every path with a trailing backslash (\).

Usage: INSTALLSQLDIR specifies the location for the instance specific binary files. The default location is: \Program Files\Microsoft SQL Server\.

 **Important:**

INSTALLSQLDIR is REQUIRED for clustered installations and must point to a local drive for which the drive letter exists on all nodes in the cluster definition.



INSTALLSQLDIR=

For more information about installation directories, see [File Locations for Default and Named Instances of SQL Server 2005](#).

[INSTALLSQLSHAREDDIR]

The INSTALLSQLSHAREDDIR parameter specifies a custom location for Integration Services, Notification Services, and Workstation components. In the example: Setup.exe <options> INSTALLSQLSHAREDDIR="<InstallDir>", the installation paths are:

- <InstallDir>\90\DTS\
- <InstallDir>\90\Notification\
- <InstallDir>\90\Tools\



INSTALLSQLSHAREDDIR=

For more information about installation directories, see [File Locations for Default and Named Instances of SQL Server 2005](#).

[INSTALLSQLDATADIR]

INSTALLSQLDATADIR specifies the location for the SQL Server data files. The default location is the under the INSTALLSQLDIR location.

Important:

INSTALLSQLDATADIR is REQUIRED for clustered installations and must point to a shared drive which is a member of the cluster group specified for installation.



INSTALLSQLDATADIR=

For more information about installation directories, see [File Locations for Default and Named Instances of SQL Server 2005](#).

[INSTALLASDATADIR]

Usage: INSTALLASDATADIR specifies the location for the Analysis Services Data Files. The default location is: INSTALLSQLDIR\Data\.

 **Note:**

Do not use this parameter with a SQL Server Express installation.

If Analysis Services is already installed on the computer, this parameter is ignored.

 [Copy Code](#)

INSTALLASDATADIR=

For more information about installation directories, see [File Locations for Default and Named Instances of SQL Server 2005](#).

[ADDLOCAL]

Usage: ADDLOCAL specifies which components to install. If ADDLOCAL is not specified, Setup fails.

To install all components specify 'ADDLOCAL=**All**' from the command prompt.

ADDLOCAL Rules:

- Feature names are case sensitive.
- To use ADDLOCAL, provide a comma-delimited list with no spaces between the features to install. For example, to install everything except Analysis Services and Reporting Services components, the list would be as follows:
 - ADDLOCAL=SQL_Engine,SQL_Data_Files,SQL_FullText,Notificati
- Selecting a parent feature only installs the parent feature, not the parent and the child. Installing the child feature automatically installs the parent and the child features. Removing the parent feature removes both the parent and the child feature.
- You can also use ADDLOCAL to add components in maintenance mode. For example, after performing the installation shown above, you could add

Analysis Services by specifying the following:

- ADDLOCAL=Analysis_Server,AnalysisDataFiles,AnalysisSharedTool

The following table shows parent/child feature relationships and their ADDLOCAL requirements for X86, IA64, and AMD64 installations of SQL Server 2005 Enterprise Edition, Developer Edition, Evaluation Edition, and Standard Edition.

SQL Server 2005 feature	Parent feature name	Child feature name
SQL Server Database Services	SQL_Engine	
SQL Server data files		SQL_Data_Files
Replication		SQL_Replication
Full-Text Search Engine		SQL_FullText
Analysis Services	Analysis_Server	
Analysis Services data files		AnalysisDataFiles
Reporting Services	RS_Server	
Report Manager		RS_Web_Interface
Notification Services	Notification_Services	
Notification Services engine components		NS_Engine
Notification Services client components		NS_Client
Integration Services		SQL_DTS
Client Components	Client_Components	
Connectivity Components		Connectivity
Management Tools		SQL_Tools90
Business Intelligence		

Development Studio		SQL_WarehouseDevWorkbench
Software Development Kit		SDK
SQLXML Client Features		SQLXML
Legacy components: Data Transformation Services 2000 Runtime and DMO		Tools_Legacy
Documentation and Samples		SQL_Documentation
SQL Server Books Online		SQL_BooksOnline
Sample databases		SQL_DatabaseSamples
AdventureWorks sample OLTP		SQL_AdventureWorksSamples
AdventureWorks sample data warehouse		SQL_AdventureWorksDWSamples
AdventureWorks sample OLAP		SQL_AdventureWorksASSamples
Samples		SQL_Samples

 [Copy Code](#)

ADDLOCAL=

The following table shows parent/child feature relationships and their ADDLOCAL requirements for X86, IA64, and AMD64 installations of SQL Server 2005 Workgroup Edition.

SQL Server 2005 Workgroup feature	Parent feature name	Child feature name
SQL Server Database Services	SQL_Engine	
SQL Server data files		SQL_Data_Files
Replication		SQL_Replication
Full-Text Search Engine		SQL_FullText

Integration Services		SQL_DTS
Client Components	Client_Components	
Connectivity Components		Connectivity
Management Tools		SQL_Tools90
Software Development Kit		SDK
SQLXML Client Features		SQLXML
Legacy components: Data Transformation Services 2000 Runtime and DMO		Tools_Legacy
Documentation and Samples		SQL_Documentation
SQL Server Books Online		SQL_BooksOnline
Sample databases		SQL_DatabaseSamples
AdventureWorks sample OLTP		SQL_AdventureWorksSamples
AdventureWorks sample data warehouse		SQL_AdventureWorksDWSamples
AdventureWorks sample OLAP		SQL_AdventureWorksASSamples
Samples		SQL_Samples

 [Copy Code](#)

ADDLOCAL=

The following table shows parent/child feature relationships and their ADDLOCAL requirements for X86, IA64, and AMD64 installations of SQL Server 2005 Express Edition.

SQL Server 2005 Express feature	Parent feature name	Child feature name
SQL Server Database Services	SQL_Engine	
SQL Server data files		SQL_Data_Files

Replication		SQL_Replication
Client Components	Client_Components	
Connectivity Components		Connectivity
Software Development Kit		SDK

 [Copy Code](#)

ADDLOCAL=

[REMOVE]

The REMOVE parameter specifies which components to uninstall, like SQL Server, Reporting Services, or Analysis Services. This command must be used in conjunction with the INSTANCENAME parameter. For example, to remove the instance aware component AND any non-instance aware components, such as Client_Components, you would specify the following:

- REMOVE=All INSTANCENAME=MSSQLSERVER

 **Important:**

Do not use this parameter if you have several instances installed and you still want to use client tools.

To remove all non-instance aware components, such as Client_Components and SQL_DTS, specify REMOVE=ALL without reference to the instance name.

When REMOVE is specified for a new installation, components listed after 'REMOVE=' are not included for installation. For example, to install every component except for Reporting Services components during a new installation specify the following:

- ADDLOCAL=All REMOVE=RS_Server,RS_Web_Interface

REMOVE can also be used during maintenance to remove components from an existing installation. For example, to remove Full-Text Search, specify the following:

- REMOVE=SQL_FullText

 **Important:**

Removing the parent feature removes both the parent and the child

feature.

 [Copy Code](#)

```
REMOVE=
```

[INSTANCENAME]

This is information for the **Instance Name** dialog box. The INSTANCENAME parameter is REQUIRED for instance-aware components. For a default instance, use MSSQLSERVER. For shared, instance-unaware components -- Notification Services, Integration Services, and client components -- this parameter can be omitted.

 [Copy Code](#)

```
INSTANCENAME=
```

[UPGRADE]

The UPGRADE parameter specifies which product to upgrade. The allowed value is **SQL_Engine** to upgrade SQL Server. If the UPGRADE option is specified, the INSTANCENAME parameter is REQUIRED to define the instance to upgrade. For example, to upgrade a default instance of SQL Server, use the following command line:

- UPGRADE=SQL_Engine INSTANCENAME=MSSQLSERVER

To upgrade a named instance of SQL Server in which the SQL Server is running under a domain account, and you would like to add the client components and tools, run this command:

- UPGRADE=SQL_Engine INSTANCENAME=<InstanceName>
SQLACCOUNT=<domainname>\<accountname> SQLPASSWORD=
<StrongPassword> ADDLOCAL=Client_Components

 [Copy Code](#)

```
UPGRADE=
```

[SAVESYSDB]

The SAVESYSDB parameter can be used during uninstall to specify not to delete system databases. If SAVESYSDB is set to **1**, the system databases are left in the SQL data directory. This option is typically used in cases where a future upgrade is planned.

 [Copy Code](#)

SAVESYSDB=

[USESYSDB]

The USESYSDB parameter specifies the root path to the data directory of the system databases during upgrade. For example, if the database was installed to D:\Microsoft SQL Server\MSSQL.1\MSSQL\DATA the usesysdb parameter should be USESYSDB="D:\Microsoft SQL Server\MSSQL.1\MSSQL".

If you set an installation path for the Database Engine feature, SQL Server Setup uses that path as the root directory for all instance-specific folders for that installation, including SQL Server data files. In this case, if you set the root to "C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\," instance-specific directories are appended to that path.

Customers choosing to use the USESYSDB upgrade functionality can easily lead themselves into a situation where the product gets installed into a recursive folder structure - for example, <SQLProgramFiles>\MSSQL.1\MSSQL\MSSQL.1\MSSQL\Data\. Instead, to use the USESYSDB feature, set an installation path for the SQL Server data files instead of the Database Engine feature.

Note:

Data files are always expected to be found in a child directory named "**Data.**" For example, specify "**C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL**" to specify the root path to the data directory of the system databases during upgrade when data files are found under "**C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Data.**"

Important:

When using USESYSDB to install SQL Server 2005 with a set of data files from a previous installation, make sure that the .MDF and .LDF files for

system databases are in the same directory.



USESYSDB=

[SQLACCOUNT], [SQLPASSWORD], [AGTACCOUNT], [AGTPASSWORD], [ASACCOUNT], [ASPASSWORD], [RSACCOUNT], and [RSPASSWORD]

This is information for the **Service Accounts** dialog box. These parameters are required for both stand-alone and virtual server installations. If these parameters are not specified, Setup will fail.

If these parameters are not specified for installations of SQL Server Express Edition, Setup will default to a specific built in machine account depending on the underlying Operating System:

- For Windows XP and Windows 2003, the default service account will be NetworkService (NT AUTHORITY).
- For Windows 2000, the default service account is LocalSystem (NT AUTHORITY\SYSTEM).

If you would like SQL Express to be installed with a specific account, it is recommended that you do not use the default settings. Instead, use the *ACCOUNT settings and make sure to specify the DOMAIN\user name - e.g., SQLACCOUNT=DOMAINNAME\ACCOUNT.

 **Note:**

By default, the services for SQL Server and Analysis Services are set to auto-start.

To use the *ACCOUNT settings, make sure to specify the DOMAIN, e.g., SQLACCOUNT=DOMAINNAME\ACCOUNT.

 **Important:**

SQLACCOUNT, SQLPASSWORD, AGTACCOUNT, and AGTPASSWORD are required for Virtual Server installations. ASACCOUNT and

ASPASSWORD cannot be used with Virtual Server installations.

 [Copy Code](#)

```
SQLACCOUNT=  
SQLPASSWORD=  
AGTACCOUNT=  
AGTPASSWORD=  
ASACCOUNT=  
ASPASSWORD=  
RSACCOUNT=  
RSPASSWORD=
```

[AUTOSTART]

To use the *AUTOSTART parameter, specify **1** to start automatically or **0** to start manually. If you decide to start SQL Server Agent automatically, then SQL is also started automatically. The default values for these parameters are specified below.

 [Copy Code](#)

```
SQLAUTOSTART=1  
AGTAUTOSTART=0  
SAUTOSTART=1  
RSAUTOSTART=1
```

[SECURITYMODE] and [SAPWD]

This is information for the **Authentication Mode** dialog box. To use Mixed Mode Authentication, use SECURITYMODE=SQL. If SECURITYMODE=SQL is not specified, then Windows Authentication only is used.

 **Important:**

When possible, use Windows Authentication.

To set the **sa** password specify SAPWD. A strong **sa** password is required by SQL Server Setup.

 **Important:**

Do not use a blank password. Use a strong password.

```
SECURITYMODE=SQL
```

 [Copy Code](#)

```
SAPWD=
```

For more information about strong password guidelines, see [Authentication Mode](#).

[SQLCOLLATION] and [ASCOLLATION]

This is information for the **Collation Settings** dialog box. The SQLCOLLATION parameter sets the SQL Server collation, and the ASCOLLATION parameter sets the Analysis Services collation.

 [Copy Code](#)

```
SQLCOLLATION=  
ASCOLLATION=
```

For more information about collation settings, see [Collation Settings in Setup](#), [Windows Collation Sorting Styles](#), and [Using Binary Collations](#).

[REBUILDDATABASE]

This is information for rebuilding system databases. When system databases are corrupted, or to reset the system collation, they must be brought back to the state to which they were when originally installed. This functionality is exposed using the REBUILDDATABASE parameter, specifying REBUILDDATABASE=1. Additional parameters required when using REBUILDDATABASE include:

- SQLACCOUNT, SQLPASSWORD
- SQLAGTACCOUNT, SQLAGTPASSWORD
- SQLBROWSERACCOUNT, SQLBROWSERPASSWORD

Implications of rebuilding the master database

Rebuilding the **master** database reverts all of the system databases to their original content and attributes.

After rebuilding the master database

After rebuilding **master**, you must:

1. Restore your most recent full backups of the **master**, **model**, and **msdb** databases. If a backup is not available, go on to Step 2.
2. If a restored backup of a database is not current, re-create any missing entries. For example, in the **master** database, re-create all missing entries for your user databases, backup devices, SQL Server logins, endpoints, and so forth. The best way to re-create entries is to run the original scripts that were used to create those entries, if available.

Security Note:

We recommend that you secure your scripts to prevent their content from being altered by unauthorized individuals.

We also recommend that when you finish updating a database, you back it up. For more information, see the topics for **Backing Up the Master Database** and **Backing Up the model and msdb Databases** in SQL Server Books Online.

3. If the server instance is configured as a replication Distributor, you must restore the **distribution** database. For more information, see the topic for **Backing Up and Restoring Replicated Databases** in SQL Server 2005 Book Online.
4. In SQL Server, system objects are stored in the **resource** database. This database is updated whenever a SQL Server update is applied to the SQL Server executable, **sqlservr.exe**. After rebuilding the master database, you must reapply any SQL Server updates that you have previously applied to the server instance. For more information, see the topic for **Resource Database** in SQL Server Books Online.

Important:

Do not include the **mssqlsystemresource.mdf** file - the **resource** database - in regular database backup and restore processes. It contains neither data nor metadata; it contains code. By default, SQL Server backup and restore components automatically omit the **resource** database.

5. Rebuilding the **master** database installs all system databases to their initial location. If you have moved one or more system databases to a different location, you must move the databases again. For more information about moving system databases, see **Moving System Databases**.

 **Note:**

The REBUILDDATABASE option is only available in unattended setup. Either the /qn or /qb option must be used. The /qn switch suppresses all Setup dialog boxes and error messages. If the /qn switch is specified, all Setup messages, including error messages, are written to Setup log files. For more information about log files, see [How to: View SQL Server 2005 Setup Log Files](#).

The /qb switch allows the display of basic Setup dialog boxes. Error messages are also displayed.

 **Important:**

In order to rebuild system databases, a new **sa** password is REQUIRED.

 [Copy Code](#)

REBUILDDATABASE=

[REINSTALLMODE]

This is information for repairing installed components. When using REINSTALLMODE, you are required to specify a REINSTALL parameter. REINSTALL parameters use the same values as ADDLOCAL parameters.

 **Note:**

The REINSTALLMODE option is only available in unattended Setup and either the /qn or /qb option must be used. The /qn switch suppresses all Setup dialog boxes and error messages. If the /qn switch is specified, all Setup messages, including error messages, are written to Setup log files. For more information about log files, see [How to: View SQL Server 2005 Setup Log Files](#).

The /qb switch allows display of basic Setup dialog boxes. Error messages are also displayed.

The supported values for REINSTALLMODE are:

- **O** - Reinstall if the file is missing or if it is an older version
- **M** - Rewrite machine-specific registry keys under HKLM
- **U** - Rewrite user-specific registry keys under HKCU
- **S** - Reinstall all shortcuts

 [Copy Code](#)

```
REINSTALLMODE=
```

[RSCONFIGURATION]

This is information for the **Reporting Services Configuration** dialog box. This parameter should be specified when Reporting Services or Report Manager features are being installed, and can be either **FilesOnly** or **Default**. The **FilesOnly** option only installs the files without actually configuring the reporting service. The Default option installs the Reporting Services using the default options.

 [Copy Code](#)

```
RSCONFIGURATION=
```

For more information, see [Report Server Installation Options](#).

[SAMPLEDATABASESERVER]

This is information for the **Sample Databases** dialog box. This parameter is used when **AdventureWorks Samples** or **AdventureWorksDW Samples** features are being installed.

Sample databases can be attached to any existing local SQL Server instance that is the same version as the sample databases. The parameter specifies the machine and instance names to which the samples should be attached, as follows:

- SAMPLEDATABASESERVER - **MachineName\InstanceName**

 [Copy Code](#)

SAMPLEDATABASESERVER=

[DISABLENETWORKPROTOCOLS]

This parameter is used for setting the startup type of the network protocols. It has the following 3 options:

- **0** - Shared memory = On, Named Pipes = On, TCP/IP = On
- **1** - Shared memory = On, Named Pipes = Off (local only), TCP/IP = Off
- **2** - Shared memory = On, Named Pipes = Off (local only), TCP/IP = On

 [Copy Code](#)

DISABLENETWORKPROTOCOLS=

[ERRORREPORTING]

This is information for the **Error Reporting** dialog box. If you enable this feature, SQL Server and its components, and Analysis Services are configured to automatically send a report to Microsoft if a fatal error occurs in the SQL Server Database Engine, SQL Server Agent, or Analysis Services. Microsoft uses error reports to improve SQL Server functionality, and treats all information as confidential.

To enable error reporting specify **ERRORREPORTING=1**. If **ERRORREPORTING=1** is not specified, the Error Reporting feature is disabled.

 [Copy Code](#)

ERRORREPORTING=

[SKUUPGRADE]

Use the SKUUPGRADE parameter when upgrading from one edition of SQL Server to another edition of SQL Server.

Important:

If you use the SKUUPGRADE parameter, Setup will remove all hotfix and service pack updates from the SQL Server instance being upgraded. Once the edition upgrade is complete, you must reapply all hotfix and service pack

updates.

To perform an edition upgrade of a default instance of SQL Server (SQL Engine only) via unattended Setup, run the following command using either /qb or /qn:

```
start /wait setup.exe ADDLOCAL=SQL_Engine  
INSTANCENAME=MSSQLSERVER UPGRADE=SQL_Engine  
SKUUPGRADE=1 /qb
```

 [Copy Code](#)

SKUUPGRADE=

Parameters Required for Cluster Installations

Note:

To run a cluster installation, you must be on a cluster and specify /qn or /qb in the command line. The /qn switch suppresses all Setup dialog boxes and error messages. If the /qn switch is specified, all Setup messages, including error messages, are written to Setup log files. For more information about log files, see [How to: View SQL Server 2005 Setup Log Files](#).

The /qb switch allows display of basic Setup dialog boxes. Error messages are also displayed. For more information about failover clustering, see [Configuring High Availability](#) and [Before Installing Failover Clustering](#).

The following parameters are required for cluster installations from command prompt.

[INSTALLSQLDIR]	[INSTALLSQLDATADIR]	[SQLACCOUNT] , [SQLPASSWORD] , [AGTACCOUNT] , and [AGTPASSWORD]
[INSTANCENAME]	[SAPWD]	[VS]
[INSTALLVS]	[IP]	[GROUP]
[ADDNODE]	[REMOVENODE]	[SQLCLUSTERGROUP] and [AGTCLUSTERGROUP]

[ADMINPASSWORD] **[SKUUPGRADE]**

[INSTALLSQLDIR]

This parameter is required for failover cluster installations. The drive letter must exist on both machines.

 [Copy Code](#)

INSTALLSQLDIR=

[INSTALLSQLDATADIR]

This parameter is required for failover cluster installations. The shared drive letter must exist in the disk group used in the GROUP parameters.

 [Copy Code](#)

INSTALLSQLDATADIR

[SQLACCOUNT], [SQLPASSWORD], [AGTACCOUNT], and [AGTPASSWORD]

These parameters are required for failover cluster installations.

 [Copy Code](#)

SQLACCOUNT=
SQLPASSWORD=
AGTACCOUNT=
AGTPASSWORD=

[INSTANCENAME]

This parameter is required for failover cluster installations. For a default instance use MSSQLSERVER. For more information about instance naming rules, see [Instance Name](#).

 [Copy Code](#)

INSTANCENAME=

[SAPWD]

This parameter is required for failover cluster installations.

 **Important:**

Do not use a blank password. Use a strong password.

 [Copy Code](#)

SAPWD=

For more information about strong password guidelines, see [Authentication Mode](#).

[VS]

This parameter is required for failover cluster installations. This parameter is used to specify the name of the virtual server. The name cannot exceed 15 characters and follows the same rules as for a computer name.

 [Copy Code](#)

VS=

[INSTALLVS]

This parameter is required for failover cluster installations. Specify the server to be installed as a cluster. Use "SQL_Engine" for SQL Server and "Analysis_Server" for Analysis Services.

 [Copy Code](#)

INSTALLVS=

[IP]

This parameter is required for failover cluster installations. Specify the IP addresses, providing one entry for each network adapter.

- Format: "IP *address*,*network*"

- Example: IP="xxx.xxx.xxx.xxx,Local Area Connection"

 [Copy Code](#)

IP=

[GROUP]

This parameter is required for failover cluster installations. Specify the Cluster group, the group that contains the disk to which the shared data files are to be written, containing SQL Server resources.

- Example: GROUP="Group 2"

 [Copy Code](#)

GROUP=

[ADDNODE]

This parameter is required for failover cluster installations. Specify the nodes for your virtual server. The nodes must exist in the cluster. You can also use this parameter to add nodes to an existing cluster.

 [Copy Code](#)

ADDNODE=

[REMOVENODE]

This parameter is required only for cluster maintenance. Use this option to remove nodes from an existing cluster.

 [Copy Code](#)

REMOVENODE=

[SQLCLUSTERGROUP] and [AGTCLUSTERGROUP]

These parameters are required for failover cluster installations. In order to install a failover cluster, you must specify a global or local domain and group name in the format *<DomainName>\<GroupName>* that exist at the time Setup is run.

SQL Server Setup will not create groups for you as part of installation. For more information, see [Domain Groups for Clustered Services](#).

 [Copy Code](#)

```
SQLCLUSTERGROUP=  
AGTCLUSTERGROUP=
```

[ADMINPASSWORD]

This parameter is required for failover cluster installations. Specify the password for the logged-on user account. The specified account must be an administrator on all remote cluster nodes.

 [Copy Code](#)

```
ADMINPASSWORD=
```

▣ **See Also**

Tasks

[How to: Install SQL Server 2005 \(Setup\)](#)

Concepts

[How to: View SQL Server 2005 Setup Log Files](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Install Microsoft Internet Information Services



[See Also](#)



[Installing SQL Server 2005 Components](#) > [Installing SQL Server Components How-to Topics](#) > [Database Engine Installation How-to Topics](#) >

Microsoft Internet information Services (IIS) is a software prerequisite for some Microsoft SQL Server 2005 components. For more information, see [Hardware and Software Requirements for Installing SQL Server 2005](#). For information on how to install IIS 7.0 on Windows Vista, see this [Microsoft Web site](#).

To install Internet Information Services

1. Close all applications that are currently running on your computer.
2. In Control Panel, double-click **Add or Remove Programs**.
3. In the left pane, click **Add/Remove Windows Components**. The Windows Components Wizard will launch.
4. Select **Internet Information Services**, and then click **Next**. If the check box is already selected, IIS is already installed on your computer.

Note:

For some server operating systems, IIS may be a component of the Application Server selection. If this is the case for your operating system, select **Application Server**, and then click **Next**.

IIS will be installed and configured on your computer.

▣ **See Also**

Concepts

[How to: View SQL Server 2005 Setup Log Files](#)

Other Resources

[Database Engine Installation How-to Topics](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Verify a Successful Installation of SQL Server 2005 Services

 [See Also](#)



[Installing SQL Server 2005 Components](#) > [Installing SQL Server Components How-to Topics](#) > [Database Engine Installation How-to Topics](#) >

To verify a successful installation of Microsoft SQL Server 2005, make sure that the installed services are running on your computer.

To verify a successful installation of SQL Server 2005 services

1. Check to see whether the SQL Server service is running: In Control Panel, double-click **Administrative Tools**, double-click **Services**, and then look for the corresponding service display name.

The following table lists service display names and the services they provide.

Note:

The actual service names differ slightly from their display names. Service names can be viewed by right-clicking a service and selecting **Properties**.

Name	Service
SQL Server (MSSQLSERVER)	The default instance of the SQL Server Database Engine.
SQL Server (<i>instancename</i>)	A named instance of the SQL Server Database Engine, where <i>instancename</i> is the name of the instance.
SQL Server Agent (MSSQLSERVER)	The default instance of SQL Server Agent. SQL Server Agent runs jobs, monitors SQL Server, fires alerts, and allows automation of some administrative tasks.

SQL Server Agent (<i>instancename</i>)	A named instance of SQL Server Agent, where <i>instancename</i> is the name of the instance. SQL Server Agent runs jobs, monitors SQL Server, fires alerts, and allows automation of some administrative tasks.
Analysis Services (MSSQLSERVER)	The default instance of Analysis Services.
Analysis Server (<i>instancename</i>)	A named instance of Analysis Services, where <i>instancename</i> is the name of the instance.
Reporting Services	The default instance of Microsoft Reporting Services.
Reporting Services (<i>instancename</i>)	A named instance of Reporting Services, where <i>instancename</i> is the name of the instance.

1. If the service is not running, start it by right-clicking the service and then clicking **Start**. If a service fails to start, look at the service properties for the path to the .exe. Make sure the .exe exists in the path specified.

▣ **See Also**

Tasks

[How to: Read a SQL Server 2005 Setup Log File](#)

Concepts

[How to: View SQL Server 2005 Setup Log Files](#)

[Troubleshooting an Installation of the SQL Server Database Engine](#)
[SQL Server Surface Area Configuration](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Change Components in an Existing Instance of SQL Server 2005 (Setup)

 [See
Also](#)



[Installing SQL Server 2005 Components](#) > [Installing SQL Server Components How-to Topics](#) > [Database Engine Installation How-to Topics](#) >

In this Microsoft SQL Server release, feature maintenance is supported only through Add or Remove Programs in Control Panel in Microsoft Windows.

To change components in an existing instance of SQL Server 2005

1. In Control Panel, double-click **Add or Remove Programs**.
2. Under **Currently installed programs**, click the instance of SQL Server to configure, and then click **Change**.
3. On the **Component Selection - Maintenance** page, select the instance or shared component to update.

Click the **Report** button to view the list of SQL Server 2005 components and features installed on your computer. The report includes version, edition, update level, and language information for each installed component and feature.

To continue, click **Next**.

4. On the **Feature Maintenance** page, select a server component to update. If you selected a shared component on the previous page, this page will not be displayed.

Click the **Report** button to view the list of SQL Server 2005 components and features installed on your computer. The report

includes version, edition, update level, and language information for each installed component and feature.

To continue, click **Next**.

5. On the **Change/Remove** page, click **Change**.
6. On the **Feature Selection** page, use the feature tree to make selection changes for your SQL Server 2005 instance. To continue, click **Next**.
7. On the **Ready to Update** page, review the changes to be made by SQL Server 2005 Setup. To continue, click **Next**.
8. On the **Setup Progress** page, Setup displays Setup status.
9. On the **Completing Setup** page, click **Finish** to exit the Installation Wizard.

▣ **See Also**

Tasks

[How to: Install SQL Server 2005 \(Setup\)](#)

[How to: Uninstall an Existing Instance of SQL Server 2005 \(Setup\)](#)

[How to: Read a SQL Server 2005 Setup Log File](#)

Concepts

[How to: View SQL Server 2005 Setup Log Files](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Rename a Computer that Hosts a Stand-Alone Instance of SQL Server 2005

 [See Also](#)



[Installing SQL Server 2005 Components](#) > [Installing SQL Server Components How-to Topics](#) > [Database Engine Installation How-to Topics](#) >

Updated: 17 July 2006

When you change the name of the computer that is running Microsoft SQL Server 2005, the new name is recognized during SQL Server startup. You do not have to run Setup again to reset the computer name. The following steps cannot be used to rename an instance of SQL Server 2005. These steps can be used only to rename the part of the instance name that corresponds to the computer name. For example, you can change a computer named MB1 that hosts an instance of SQL Server named Instance1 to another name, such as MB2. However, the instance portion of the name, Instance1, will remain unchanged. In this example, the *\\ComputerName\InstanceName* would be changed from *\\MB1\Instance1* to *\\MB2\Instance1*.

Before you begin

Before you begin the renaming process, review the following information:

- When an instance of SQL Server is part of a SQL Server failover cluster, the process of renaming the computer differs from the process of renaming a computer that hosts a stand-alone instance. For information about renaming a computer that hosts a failover cluster instance of SQL Server 2005, see [How to: Rename a SQL Server 2005 Virtual Server](#).
- SQL Server does not support renaming computers that are involved in replication, except in the case of using log shipping with replication. The secondary computer in log shipping can be renamed if the primary computer is permanently lost. For more information, see **Replication and Log Shipping**.
- When renaming a computer that is configured to use database mirroring, you must turn off database mirroring before the renaming operation, and

then re-establish database mirroring with the new computer name. Metadata for database mirroring will not be updated automatically to reflect the new computer name.

- After the computer renaming operation, users who connect to SQL Server through a Windows group that uses a hard-coded reference to the computer name will not be able to connect to SQL Server if the Windows group specifies the old computer name. To ensure that such Windows groups have SQL Server connectivity following the renaming operation, update the Windows group to specify the new computer name.

You can connect to SQL Server using the new computer name after you have restarted SQL Server. However, to ensure that @@servername returns the updated name of the local server instance, you should manually run one of the following procedures, depending on whether you are updating a default or named instance.

☐ **To rename a computer that hosts a stand-alone instance of SQL Server 2005**

- For a renamed default instance, run the following procedures:

 [Copy Code](#)

```
sp_dropserver <old_name>
GO
sp_addserver <new_name>, local
GO
```

Restart the SQL Server instance.

- For a renamed named instance, run the following procedures:

 [Copy Code](#)

```
sp_dropserver <old_name\instancename>
GO
sp_addserver <new_name\instancename>, local
GO
```

Restart the SQL Server instance.

☐ **Verify the renaming Operation**

After a computer has been renamed, any connections that used the old computer name must connect using the new name.

☐ **To verify that the renaming operation has completed successfully**

- Select information from either @@servername or sys.servers. The @@servername function will return the new name, and the sys.servers table will show the new name.

▣ Issues with Remote Logins

If the computer has any remote logins, running **sp_dropserver** may generate an error similar to this:

 [Copy C](#)

```
Server: Msg 15190, Level 16, State 1, Procedure sp_dropserver, Line  
There are still remote logins for the server 'SERVER1'.
```

To resolve the error, you must drop remote logins for this server.

▣ To drop remote logins

- For a default instance, run the following procedure:

 [Copy Code](#)

```
sp_dropremotelogin old_name  
GO
```

- For a named instance, run the following procedure:

 [Copy Code](#)

```
sp_dropremotelogin old_name\instancename  
GO
```

▣ **See Also**

Tasks

[How to: Rename a SQL Server 2005 Virtual Server](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Uninstall an Existing Instance of SQL Server 2005 (Setup)



[See Also](#)



[Installing SQL Server 2005 Components](#) > [Installing SQL Server Components How-to Topics](#) > [Database Engine Installation How-to Topics](#) >

Use the following procedure to uninstall an instance of Microsoft SQL Server 2005.

 **Note:**

You must perform the uninstall steps listed here in the exact order given.

 **Important:**

To maintain or update a SQL Server instance, you must be a local administrator with permission to log on as a service.

Before you begin, consider the following important information before you uninstall SQL Server 2005:

On a computer running SQL Server 2005 along with previous SQL Server versions, Enterprise Manager and other programs that depend on SQL-DMO may be disabled. This may occur in the following situations:

- Side-by-side installations of SQL Server 2005, SQL Server 2000, and/or SQL Server 7.0, where any of the versions is uninstalled.
- Side-by-side installations of SQL Server 2000 with SQL Server 2005, where SQL Server 2000 is installed after SQL Server 2005.

This issue is due to removal of the registration for the SQL Server 2005 SQL-DMO COM library. To re-enable Enterprise Manager and other programs with SQL-DMO dependencies, register SQL-DMO by running 'regsvr32.exe

sqldmo.dll' from a command prompt. For more information, see [Troubleshooting an Installation of SQL Server 2005](#).

 **Note:**

Before you remove SQL Server 2005 components from a computer that has the minimum required amount of physical memory you must ensure that the page file size is equal to two times the amount of physical memory. In some situations insufficient virtual memory may result in an incomplete removal of SQL Server 2005.

For information on command prompt syntax, see [How to: Install SQL Server 2005 from the Command Prompt](#).

For information on removing a failover cluster, see [How to: Remove a SQL Server 2005 Failover Clustered Instance \(Setup\)](#).

▣ **To remove a SQL Server 2005 instance**

1. To begin the uninstall process, on the Microsoft Windows desktop click **Start**, click **Control Panel**, and then double-click **Add or Remove Programs**.
2. Select the SQL Server 2005 component to uninstall, and then click **Remove**. This will start the SQL Server 2005 Installation Wizard.
3. On the **Component Selection - Uninstall** page, select any server and/or shared components to remove.

Click the **Report** button to view the list of SQL Server 2005 components and features installed on your computer. The report includes version, edition, update level, and language information for each installed component and feature.

To continue, click **Next**.

4. On the **Confirmation - Uninstall** page, review the list of components

and features that will be removed. To continue, click **Next**.

5. The **Setup Progress** page will display Setup status.
6. On the **Completing Setup** page, click **Finish** to exit the Installation Wizard.

▣ **See Also**

Tasks

[How to: Read a SQL Server 2005 Setup Log File](#)

Concepts

[How to: View SQL Server 2005 Setup Log Files](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Manually Uninstall an Existing Instance of SQL Server 2005



[See Also](#)



[Installing SQL Server 2005 Components](#) > [Installing SQL Server Components How-to Topics](#) > [Database Engine Installation How-to Topics](#) >

New: 17 July 2006

Follow these steps to uninstall an instance of Microsoft SQL Server 2005.

Important:

To maintain or update an instance of SQL Server, you must be a local administrator with permission to log on as a service.

This article describes how to manually uninstall a stand-alone instance of Microsoft SQL Server 2005. By following the steps in this topic, you also prepare the system so that you can reinstall SQL Server.

For information about how to uninstall a SQL Server 2005 failover cluster, see [How to: Manually Uninstall a SQL Server 2005 Failover Cluster](#) or [How to: Remove a SQL Server 2005 Failover Clustered Instance \(Setup\)](#). For information about how to manually uninstall an instance of SQL Server 2000, see [How to: manually remove SQL Server 2000 default, named, or virtual instance](#).

Consider the following important information before you use this procedure to uninstall SQL Server 2005:

- We recommend that you use **Add or Remove Programs** in Control Panel to uninstall SQL Server 2005. However, if an installation fails, or if **Add or Remove Programs** does not uninstall the instance of SQL Server 2005, you can follow the steps in this topic to manually uninstall the instance.

- This article contains information about how to modify the registry. Make sure that you back up the registry before you modify it, and know how to restore the registry if a problem occurs. For more information about how to back up, restore, and modify the registry, see [Description of the Microsoft Windows registry](#) in the Microsoft knowledge base.
- On a computer that is running SQL Server 2005 along with previous SQL Server versions, Enterprise Manager and other programs that depend on SQL-DMO might be disabled. This can occur in the following situations:
 - Side-by-side installations of any combination of SQL Server 2005, SQL Server 2000, and SQL Server 7.0, where any one or more of these instances is uninstalled.
 - Side-by-side installations of SQL Server 2000 with SQL Server 2005, where SQL Server 2000 is installed after SQL Server 2005.

This issue is due to removal of the registration for the SQL Server 2005 SQL-DMO COM library. To re-enable Enterprise Manager and other programs that have SQL-DMO dependencies, register SQL-DMO by running **regsvr32.exe sqlldmo.dll** at the command prompt. For more information, see [Troubleshooting an Installation of the SQL Server Database Engine](#).

- Before you remove SQL Server 2005 components from a computer that has the minimum required amount of physical memory, you must make sure that the page file size is equal to two times the amount of physical memory. Sometimes, insufficient virtual memory can cause an incomplete removal of SQL Server 2005.
- If you receive the following error message during SQL Server 2005 Setup, a SQL Server 2005 component was not uninstalled from the computer:
A component that you have specified in the ADD_LOCAL property is already installed. To upgrade the existing component, refer to the template.ini and set the UPGRADE property to the name of the component.

Before you remove SQL Server 2005, follow these steps:

- **Back up your data.** You might have databases that you want to save in their present state. You might also want to save changes that were made to the system databases. If either situation is true, make sure that back up the data before you uninstall SQL Server 2005. Alternatively, save a

copy of all the data and log files in a folder other than the MSSQL folder. The MSSQL folder is deleted during uninstallation.

The files that you must save include the following database files. These files are installed as part of SQL Server 2005:

- Distmdl.*
 - Master.*
 - Mastlog.*
 - Model.*
 - Modellog.*
 - Msdbdata.*
 - Msdblog.*
 - Mssqlsystemresource.*
 - Northwind.* (This database is an optional installation.)
 - Pubs.*
 - Pubs_log.*
 - Tempdb.*
 - Templog.*
 - ReportServer[\$InstanceName] (This is the Reporting Services default database.)
 - ReportServer[\$InstanceName]TempDB (This is the Reporting Services default temporary database.)
- **Delete the local security groups.** Before you uninstall SQL Server 2005, delete the local security groups for SQL Server 2005 components.
 - **Save or rename SQL Server Reporting Services folders.** If you use the SQL Server installation together with Reporting Services, save or rename the following folders and subfolders:
 - <drive>\Microsoft SQL Server\Reporting Services
 - <drive>\Microsoft SQL Server\MSSQL\Reporting Services
 - <drive>\Microsoft SQL Server\<SQL Server instance name>\Reporting Services
 - <drive>\Microsoft SQL Server\90\Tools\Reporting Services

 **Note:**

If you used the SSRS configuration tool to configure the installation, the names might differ from the names in this list. Additionally, the databases might be located on a remote computer that is running SQL Server.

- **Delete the Reporting Services virtual directories.** Use Microsoft Internet Information Services (IIS) Manager to delete the following virtual directories:
 - ReportServer[\$InstanceName]
 - Reports[\$InstanceName]
- **Delete the ReportServer application pool.** Use IIS Manager to delete the ReportServer application pool.
- **Stop all SQL Server services.** We recommend that you stop all SQL Server services before you uninstall SQL Server 2005 components. Active connections can prevent successful uninstallation.
- **Use an account that has the appropriate permissions.** Log on to the server by using the SQL Server service account or by using an account that has equivalent permissions. For example, you can log on to the server by using an account that is a member of the local Administrators group.



Caution:

Incorrectly editing the registry can severely damage your system. Before making changes to the registry, we recommend that you back up any valued data on the computer.



Note:

You must perform the uninstallation steps listed here in the exact order given.

□ **To manually uninstall an instance of SQL Server 2005**

1. Make sure that the SQL Server Setup Support Files component is installed.

In **Add or Remove Programs**, make sure that **Microsoft SQL Server Setup Support Files** appears in the list of installed programs. If **Microsoft SQL Server Setup Support Files** appears in the list, go to

step 2.

If **Microsoft SQL Server Setup Support Files** does not appear in the list, install this component before you continue. To do this, from the SQL Server 2005 installation media, double-click the Servers\setup\sqlsupport.msi file. In step 5 of this procedure, you will uninstall the Microsoft SQL Server Setup Support Files component.

2. At the command prompt, run the following command to uninstall the SQL Server components:

```
%ProgramFiles%\Microsoft SQL Server\90\Setup Bootstrap\ARPWr
```

Uninstall the SQL Server components one at a time until all the SQL Server components are uninstalled.

 **Note:**

Add or Remove Programs also runs the ARPWrapper.exe program by using the /Remove option. However, the reference to the ARPWrapper.exe program might have been deleted.

If you receive any of the following error messages, see the "If you experience problems" section that follows:**Registry Enumeration Failed**

If you have an instance of SQL Server that has more than one server component, such as the Database Engine and Analysis Services installed, setup will fail, and you receive the following error message:

The setup has encountered an unexpected error in datastore. The action is RestoreSetupParams.

This error is expected because of refcounting. A refcount is used to track how many programs use the same .dll file or the same .msi file. When a program is uninstalled, the program decrements the refcount for an .msi file by one. When the last program is uninstalled, the .msi file is deleted. In this case, the .msi file is the SQL Server Setup Support Files component (SqlSupport.msi). If you receive this error message, use one

of the following methods to resolve the error:

- Reinstall the SQL Server Setup Support Files component (SqlSupport.msi) before you uninstall each component in the instance.
- Temporarily refcount the SqlSupport.msi file to itself. To do this, follow these steps:
 1. Obtain the GUID for the Microsoft SQL Server Setup Support Files component. To obtain the GUID, follow steps 1 and 2 from the "If you experience problems" section.
 2. Use Registry Editor to create or to modify the following registry key with these values:

Key Name:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\90\Bootstrap\MSIRefCount

Value: Uninstall

Type: REG_SZ

Data: {11111111-1111-1111-1111-111111111111},{GUID}

Note GUID is a placeholder for the GUID of the SQL Server support files. For example, the GUID of the English x86 SQL Server support files is as follows:

Data: {11111111-1111-1111-1111-111111111111},{53F5C3EE-05ED-4830-994B-50B2F0D50FCE}

If you experience problems

If you experience problems when you try to uninstall the SQL Server components, follow these steps:

- Start Registry Editor, and then locate the following registry key:
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\Cu
- In the left pane, click each GUID. For each GUID that you click, look for a display name in the right pane that includes "Microsoft SQL Server 2005." For example, look for the following names:
- Microsoft SQL Server 2005
- Microsoft SQL Server 2005Analysis Services
- Microsoft SQL Server 2005Reporting Services
- Microsoft SQL Server 2005Notification Services
- Microsoft SQL Server 2005Integration Services

- Microsoft SQL Server 2005 Tools
 - Microsoft SQL Server 2005 Books Online
- When you see a display name that includes "Microsoft SQL Server 2005," note the GUID that you clicked in the left pane.
- For each GUID that you noted in the previous step, run the following command at the command prompt:

 [Copy Code](#)

```
start /wait msixec /x {GUID} /l*v c:\sql_uninstall.log
```

Notes:

- In this command, replace GUID with the GUID that you noted in the previous step.
- If you receive the following error message, contact product support.
**The setup has encountered an unexpected error in datastore.
Please contact a Customer Support representative.**

For a complete list of product support telephone numbers and for information about support costs, see the [support contact information](#) Microsoft Web site.

- Open the file at C:\Sql_uninstall.log. Approximately 15 lines from the bottom of the file, locate a line that resembles the following:
MSI (s) (EC:F8) [12:52:18:007]: Product: Microsoft SQL Server 2005 Tools -- Removal completed successfully.

If the uninstallation failed, note the component name and the GUID, and then save the Sql_uninstall.log file.

3. Use **Add or Remove Programs** to uninstall the support components in the following order:
 - MSXML 6.0 Parser
 - SQLXML4
 - SQL Server VSS Writer
 - SQL Server 2005 Backward Compatibility
 - Anything else that is related to SQL Server 2005, except for the SQL Server Native Client component and for the SQL Server Setup Support Files component.

If the uninstallation fails for any one of the support components, you might receive the following error message:

Setup failed due to missing prerequisites

If you receive this error message, follow these steps:

- Start Registry Editor, and then locate the following registry key:
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\Cu
- In the left pane, click each GUID. For each GUID that you click, look for a display name in the right pane that matches the name of the redistributable file. When you see a display name that matches the name of the redistributable file, note the GUID that you clicked in the left pane.
- For each GUID that you noted in the previous step, run the following command at a command prompt:

```
start /wait msixec /x {GUID} SKIPREDISTPREREQS=1 /l*v c
```

In this command, replace GUID with the GUID that you noted in the previous step.

- Open the file at C:\Sqlredist_uninstall.log. At the bottom of the file, locate a line that resembles the following line:
MSI (s) (EC:F8) [12:52:18:007]: Product: Microsoft SQL Server 2005 Backward compatibility -- Removal completed successfully.

If the uninstallation failed, note the component name and the GUID, and then save the Sqlredist_uninstall.log file.

4. If all the other steps were successful, uninstall the SQL Server Native Client component by using **Add or Remove Programs**.

Do not uninstall the SQL Server Native Client component if you have SQL Server components installed.

5. If all the other steps were successful, uninstall the SQL Server Setup Support Files component by using **Add or Remove Programs**.

If these steps did not uninstall all the components and all the files that are related to the instance of SQL Server, contact product support. For a complete list of product support telephone numbers, see the [Microsoft](#)

[Help and Support Web site.](#)

▣ **See Also**

Tasks

[How to: Read a SQL Server 2005 Setup Log File](#)

Concepts

[How to: View SQL Server 2005 Setup Log Files](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Uninstall an Existing Instance of SQL Server 2000 or 7.0



[See Also](#)



[Installing SQL Server 2005 Components](#) > [Installing SQL Server Components How-to Topics](#) > [Database Engine Installation How-to Topics](#) >

Use the following procedure to remove an existing instance of Microsoft SQL Server 2000 or SQL Server 7.0.

Before you begin, consider the following important information before you uninstall SQL Server:

On a computer running SQL Server 2005 along with previous SQL Server versions, Enterprise Manager and other programs that depend on SQL-DMO may be disabled. This may occur in the following situations:

- Side-by-side installations of SQL Server 2005, SQL Server 2000, and/or SQL Server 7.0, where any of the versions is uninstalled.
- Side-by-side installations of SQL Server 2000 with SQL Server 2005, where SQL Server 2000 is installed after SQL Server 2005.

This issue is due to removal of the registration for the SQL Server 2005 SQL-DMO COM library. To re-enable Enterprise Manager and other programs with SQL-DMO dependencies, register SQL-DMO by running 'regsvr32.exe sqldmo.dll' from a command prompt. For more information, see [Troubleshooting an Installation of SQL Server 2005](#).

To uninstall an existing instance of SQL Server

1. Insert the SQL Server 2000 CD in your CD-ROM drive. If the CD does not autorun, double-click **Autorun.exe** in the root directory of the compact disc.
2. Click **SQL Server 2000 Components**, and then click **Install Database Server**. Setup starts the SQL Server Installation Wizard. At the

Welcome screen, click **Next**.

3. In the **Computer Name** dialog box, click **Local Computer** or **Remote Computer**.
4. In the **Installation Selection** dialog box, click **Upgrade, Remove, or Add Components to an Existing Instance of SQL Server**, and then click **Next**.
5. In the **Instance Name** dialog box, the **Default** check box is selected if you have the default instance installed. If you want to uninstall a named instance, select it in the **Instance Name** list box, and then click **Next**.
6. In the **Existing Installation** dialog box, click **Uninstall your existing installation**, and then click **Next**.
7. Setup removes the selected installation. In the **Uninstalling** dialog box, click **Next**, and then in the **Setup Complete** dialog box, click **Finish**.

▣ **See Also**

Tasks

[How to: Upgrade to SQL Server 2005 \(Setup\)](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: View SQL Server 2005 Setup Log Files



[See Also](#)



[Installing SQL Server 2005 Components](#) > [Installing SQL Server Components How-to Topics](#) > [Database Engine Installation How-to Topics](#) >

Logging for Microsoft SQL Server Setup has changed in this release. The main log you should look at is located in

`%ProgramFiles%\Microsoft SQL Server\90\Setup Bootstrap\LOG\Summary.txt`

This file will have content like:

Machine : MYSERVER

Product : Microsoft XML Parser

Product Version : 8.60.1639.0

Installation : Successful

Machine : MYSERVER

Product : MSXML 6.0 Parser and SDK

Product Version : 6.00.3562.0

Installation : Successful

Machine : MYSERVER

Product : Microsoft SQL Server 2005 Analysis Services

Product Version : 9.00.787

Installation : Successful

Machine : MYSERVER

Product : Microsoft SQL Native Client

Product Version : 9.00.121

Installation : Successful

If the log file shows a failure for a "Product" you can investigate the root cause of the failure by looking at the log for that product (search for "return value 3"). The product logs can be found in:

%ProgramFiles%\Microsoft SQL Server\90\Setup Bootstrap\LOG\Files

The logs have the following naming convention:

SQLSetup[XXXX][s]_[COMPUTERNAME]_[PRODUCTNAME]_[Y].log, where

- *XXXXX* = the enumerated number of the install, where the last installation performed has the highest number
- *COMPUTERNAME*= The computer where Setup is being run
- *PRODUCTNAME*= Product name (the name of the .msi file); for example, SQLSetup0001_MachineName_RS.log for Reporting Services, SQLSetup0001_MachineName_NS.log for Notification Services
- *Y*= If a Microsoft Windows Installer file (.msi) is installed more than once during a single setup run this number is added and incremented to the log name. This is mainly seen for Microsoft XML Core Services (MSXML).

The following table contains sample log file names and log file descriptions for installation of each SQL Server component.

Log file	Description
SQLSetup0001_MachineName_Core.log	The primary SQL Server Setup log file that contains information about the success or failure of all .msi installations.
SQLSetup0001_MachineName_SCC.log	Detailed information from the SQL Server 2005 System Configuration Checker that runs system checks during SQL Server Setup.
SQLSetup0001_MachineName_SQL.log	The Database Engine Setup log file.
SQLSetup0001_MachineName_AS.log	The Analysis Services Setup log file.
SQLSetup0001_MachineName_RS.log	The Reporting Services Setup log file.
SQLSetup0001_MachineName_NS.log	The SQL Server 2005 Notification Services Setup log file.
SQLSetup0001_MachineName_DTS.log	The Integration Services Setup log file.
SQLSetup0001_MachineName_Tools.log	The client components, SQL Server Management Studio, and Books Online Setup log file.

SQLSetup0001_MachineName_WI.log	The Setup user interface (UI) Setup log file.
SQLSetup0001_MachineName_SNAC.log	The SQL Server Native Client Setup log file.
SQLSetup0001_MachineName_.NET Framework 2.0.log	The .NET Framework 2.0 Setup log file.
SQLSetup0001_MachineName_MSXML.log	MSXML Setup log file.
SQLSetup0001_MachineName_SQLXML.log	SQLXML Setup log file.
SQLSetup0001_MachineName_Logs.cab	For failover cluster installations, this file will contain logs from all cluster nodes.
SQLSetup0001_MachineName_Datastore.xml	A dump of all properties that are cached during Setup. This information may be requested by Product Support Services to troubleshoot and debug failures.

▣ **See Also**

Tasks

[How to: Read a SQL Server 2005 Setup Log File](#)

Concepts

[Troubleshooting an Installation of the SQL Server Database Engine](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Read a SQL Server 2005 Setup Log File



[See Also](#)



[Installing SQL Server 2005 Components](#) > [Installing SQL Server Components How-to Topics](#) > [Database Engine Installation How-to Topics](#) >

SQL Server 2005 Setup creates log files for each SQL Server component during installation; logging starts as soon as Setup.exe is launched.

To read a SQL Server Setup log file

1. Open Windows Explorer and navigate to:

`%ProgramFiles%\Microsoft SQL Server\90\Setup Bootstrap\LOG`

2. Open the Summary.txt file. Look through it for error messages.
3. If you do not find information in the Summary.txt file that tells you what failed, open the SQLSetup[xxxx].cab file in the same root directory.

4. If this .cab file does not exist, open up the CORE log file from:

`%ProgramFiles%\Microsoft SQL Server\90\Setup Bootstrap\LOG\Files`

Look at the most recently modified core log:
SQLSetup[xxxx]_[ComputerName]_Core.log.

5. If an error was displayed in the user interface during Setup, look for this log: SQLSetup[xxxx][s]_[ComputerName]_WI.log.
6. When parsing a SQLSetup_[xxxx] log file, search the file for 'UE 3'. This is short for 'Return Value 3'.
7. If a component failed during the installation progress dialog, open the log file SQLSetup[xxxx]_[ComputerName]_SQL.log, and perform the

'UE 3' search to scan for errors.

▣ **See Also**

Concepts

[How to: View SQL Server 2005 Setup Log Files](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Rebuild the Master Database for SQL Server 2005



[See Also](#)



[Installing SQL Server 2005 Components](#) > [Installing SQL Server Components How-to Topics](#) > [Database Engine Installation How-to Topics](#) >

Use of the Setup command to rebuild the **master** database has changed in this Microsoft SQL Server release. Run Setup.exe to rebuild, verify, and repair a SQL Server instance, and rebuild the system databases. This procedure is most often used to rebuild the **master** database for a corrupted installation of SQL Server.

 **Note:**

When possible, use Windows Authentication.

For more information and example command prompt syntax, see [How to: Install SQL Server 2005 from the Command Prompt](#).

▣ **See Also**

Tasks

[How to: Read a SQL Server 2005 Setup Log File](#)

Concepts

[How to: View SQL Server 2005 Setup Log Files](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Rebuild the Registry for SQL Server 2005



[See Also](#)



[Installing SQL Server 2005 Components](#) > [Installing SQL Server Components How-to Topics](#) > [Database Engine Installation How-to Topics](#) >

To rebuild the registry, from the command prompt run the **Setup** command with the **REINSTALL=ALL, REINSTALDATABASE=1, and REINSTALLMODE=omus** properties. This will rebuild, verify, and repair a Microsoft SQL Server instance. This procedure is most often used to rebuild the registry for a damaged installation of SQL Server.

Note:

To rebuild the registry, you must use the same package file and options that you specified during the initial installation. If you do not know this information, uninstall and then reinstall SQL Server, rather than rebuilding the registry.

Important:

When possible, use Windows Authentication. Do not leave the **sa** password blank; always use strong passwords.

For more information and sample command prompt syntax, see [How to: Install SQL Server 2005 from the Command Prompt](#).

▣ **See Also**

Tasks

[How to: Read a SQL Server 2005 Setup Log File](#)

Concepts

[How to: View SQL Server 2005 Setup Log Files](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Access Books Online for SQL Server 2005



[See Also](#)



[Installing SQL Server 2005 Components](#) > [Installing SQL Server Components How-to Topics](#) > [Database Engine Installation How-to Topics](#) >

Use the following procedure to start Microsoft SQL Server 2005 Books Online from the **Start** menu on the Microsoft Windows desktop, or to locate the SQL Server 2005 Books Online default installation directory.

To access SQL Server 2005 Books Online from the Start menu

1. Point to **All Programs**, point to **Microsoft SQL Server 2005**, and then click **SQL Server Books Online**.
2. SQL Server 2005 Books Online will start.

To access SQL Server 2005 Books Online from Windows Explorer

1. From the Windows desktop, right-click **Start**, and then click **Explore**.
2. Navigate to %ProgramFiles%\ Microsoft SQL Server\90\Tools\Books\1033. This is the default installation directory for the English-language SQL Server Books Online.

Note:

If you installed SQL Server Books Online in a language other than English, the SQL Server Books Online installation directory will be in a language-specific folder in the ...\\Tools\\Books\\ directory.

▣ **See Also**

Tasks

[How to: Change Components in an Existing Instance of SQL Server 2005 \(Setup\)](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Access Books Online for SQL Server 2000



[Installing SQL Server 2005 Components](#) > [Installing SQL Server Components How-to Topics](#) > [Database Engine Installation How-to Topics](#) >

Use the following information to locate Microsoft SQL Server 2000 Books Online.

☐ To locate SQL Server 2000 Books Online from the Start menu

1. Point to **All Programs**, point to **Microsoft SQL Server**, and then click **Books Online**.
2. SQL Server 2000 Books Online will start.

☐ To locate SQL Server 2000 Books Online from Windows Explorer

1. From the Microsoft Windows desktop, right-click **Start**, and then click **Explore**.
2. Navigate to %ProgramFiles%\Microsoft SQL Server\80\Tools\Books\
This is the default installation directory for the English-language SQL Server Books Online.

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Access Books Online for SQL Server 7.0



[Installing SQL Server 2005 Components](#) > [Installing SQL Server Components How-to Topics](#) > [Database Engine Installation How-to Topics](#) >

Use the following procedures to start Microsoft SQL Server 7.0 Books Online from the **Start** menu and to locate the SQL Server 7.0 Books Online default installation directory.

□ **To access SQL Server 7.0 Books Online from the Start menu**

1. Click **Start**, point to **All Programs**, point to **Microsoft SQL Server**, and then click **Books Online**.
2. SQL Server 7.0 Books Online will start.

□ **To access SQL Server 7.0 Books Online from Windows Explorer**

1. From the Microsoft Windows desktop, right-click **Start**, and then click **Explore**.
2. Navigate to %ProgramFiles%\MSSQL7\Books. This is the default installation directory for the English-language SQL Server Books Online.

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Analysis Services Installation How-to Topics



[See Also](#)



[Installing SQL Server 2005 Components](#) > [Installing SQL Server Components How-to Topics](#) >

This section contains the following common procedures used for installing Microsoft SQL Server 2005 Analysis Services (SSAS).

For common procedures used for upgrading from a previous version of Analysis Services and on migrating Analysis Services packages, see [Upgrading or Migrating Analysis Services How-to Topics](#).



Important:

The Analysis Services installation how-to topics in this section focus only on installation options that are specific to Analysis Services. However, these topics provide links to other topics for more information about installation of SQL Server.

▣ In This Section

- [How to: Install Analysis Services by Using Setup](#)
- [How to: Install Analysis Services from the Command Prompt](#)
- [How to: Install Analysis Services on a Failover Cluster](#)
- [How to: Install Multiple Instances of Analysis Services](#)
- [How to: Install Analysis Services Side-by-Side with an Earlier Version](#)
- [How to: Rename an Instance of Analysis Services](#)
- [How to: Add or Remove Features for an Instance of Analysis Services](#)
- [How to: Uninstall an Instance of Analysis Services](#)

▣ **See Also**

Concepts

[Installing SQL Server Analysis Services](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Install Analysis Services by Using Setup



[See Also](#)



[Installing SQL Server 2005 Components](#) > [Installing SQL Server Components How-to Topics](#) > [Analysis Services Installation How-to Topics](#) >

You can install Microsoft SQL Server 2005 Analysis Services (SSAS) by using the SQL Server Installation Wizard, or you can install it at the command prompt. The Installation Wizard provides a graphical user interface that guides you through each install-time decision, and is the recommended method for most users. The Installation Wizard provides guidance for initial setup of SQL Server 2005, including feature selection, instance naming rules, service account configuration, strong password guidelines, and collation selection.

Important:

This topic and the other Analysis Services installation how-to topics in this section focus only on installation options that are specific to Analysis Services. However, these topics provide links to other topics for more information about installation of SQL Server.

To Install Analysis Services by Using Setup

1. Start the SQL Server Installation Wizard.
2. Optionally, review [How to: Install SQL Server 2005 \(Setup\)](#) for a page-by-page description of the Installation Wizard, especially if you are installing other SQL Server components in addition to Analysis Services.
3. On the **Components to install** page of the Installation Wizard, select **Analysis Services**. Review [Installing SQL Server Analysis Services by Using Setup](#) for more information about other SQL Server features, such as documentation, samples, and development tools, that you may

want to include in your installation.

4. On the **Instance name** page, select a default or named instance for your installation. If a default instance already exists, or if an instance of SQL Server 2000 Analysis Services that you do not want to upgrade is already present on the computer, you can only select a named instance.
5. On the **Service account** page, specify the user name, password, and domain name for the Analysis Services service account and for other SQL Server service accounts. You can use one account for all the services.
6. On the **Collation settings** page, specify the default collation. You can use one collation for Analysis Services and another collation for the SQL Server 2005 Database Engine, or you can specify a single collation for all of the services.
7. Complete the Installation Wizard. The remaining pages of the wizard do not contain option settings that are pertinent to Analysis Services.

▣ **See Also**

Other Resources

[Analysis Services Installation How-to Topics](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Install Analysis Services from the Command Prompt



[See Also](#)



[Installing SQL Server 2005 Components](#) > [Installing SQL Server Components How-to Topics](#) > [Analysis Services Installation How-to Topics](#) >

You can install Microsoft SQL Server 2005 Analysis Services (SSAS) by using the SQL Server Installation Wizard, or you can install it from the command prompt. The Installation Wizard provides a graphical user interface that guides you through each install-time decision, and is the recommended method for most users. Installations at the command prompt are for advanced scenarios such as silent and remote installation. You can run the installation directly at the command prompt or from command prompt syntax that references an .ini file to specify installation options.

Important:

This topic and the other Analysis Services installation how-to topics in this section focus only on installation options that are specific to Analysis Services. However, these topics provide links to other topics for more information about installation of SQL Server.

To Install Analysis Services from the Command Prompt

1. Decide whether you want to specify setup options at the command prompt or in an .ini file. For more information about how to use an .ini file to specify these options, see [How to: Install SQL Server 2005 from the Command Prompt](#).
2. Start Setup.exe at the command prompt, and provide the required information at the command prompt or in an .ini file, as described in [How to: Install SQL Server 2005 from the Command Prompt](#). Review [Installing SQL Server Analysis Services from the Command Prompt](#) for more information about the installation options that are specific to

Analysis Services.

▣ **See Also**

Other Resources

[Analysis Services Installation How-to Topics](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Install Analysis Services on a Failover Cluster



[See Also](#)



[Installing SQL Server 2005 Components](#) > [Installing SQL Server Components How-to Topics](#) > [Analysis Services Installation How-to Topics](#) >

You can install Microsoft SQL Server 2005 Analysis Services (SSAS) on a failover cluster that benefits from Microsoft Cluster Service (MSCS) to provide superior reliability.

Important:

This topic and the other Analysis Services installation how-to topics in this section focus only on installation options that are specific to Analysis Services. However, these topics provide links to other topics for more information about installation of SQL Server.

Note:

The SQL Server 2005 Analysis Services (SSAS) Instance Rename tool is not supported for use in a cluster environment.

To install Analysis Services on a failover cluster

1. Before you start installation, review [Installing SQL Server Analysis Services by Using Setup](#) to understand the options that are specific to Analysis Services that you will have to select during Setup.
2. Review the information and complete the prerequisites that are specific to cluster installation, described in [How to: Create a New SQL Server 2005 Failover Cluster \(Setup\)](#). Be prepared to provide a virtual server instance name and IP address for the clustered Analysis Services installation that is unique and distinct from the names and IP addresses

of the cluster itself and from the nodes in the cluster.

3. The steps to install SQL Server 2005 Analysis Services on a failover cluster, and the pages of the SQL Server Installation Wizard that are specific to a cluster installation, are the same for Analysis Services as for the SQL Server 2005 Database Engine. Follow the steps described in [How to: Create a New SQL Server 2005 Failover Cluster \(Setup\)](#) to run the Installation Wizard. Make selections that are appropriate for Analysis Services on the **Components to Install**, **Instance Name**, **Service Account**, and **Collation Settings** pages, as discussed in [Installing SQL Server Analysis Services by Using Setup](#).

▣ **See Also**

Tasks

[How to: Create a New SQL Server 2005 Failover Cluster \(Setup\)](#)

Concepts

[Before Installing Failover Clustering](#)

[Installing SQL Server Analysis Services by Using Setup](#)

[Installing SQL Server Analysis Services from the Command Prompt](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Install Multiple Instances of Analysis Services



[See Also](#)



[Installing SQL Server 2005 Components](#) > [Installing SQL Server Components How-to Topics](#) > [Analysis Services Installation How-to Topics](#) >

You can run multiple instances of Microsoft SQL Server 2005 Analysis Services (SSAS) on the same computer. To install an additional instance, run Setup again and specify a unique name for the new instance.

Important:

This topic and the other Analysis Services installation how-to topics in this section focus only on installation options that are specific to Analysis Services. However, these topics provide links to other topics for more information about installation of SQL Server.

To install an additional instance of Analysis Services

1. Run Setup either by using the SQL Server Installation Wizard or from the command prompt. For more information about how to run Setup by using the SQL Server Installation Wizard, see [How to: Install Analysis Services by Using Setup](#). For more information about how to run Setup from the command prompt, see [How to: Install Analysis Services from the Command Prompt](#).
2. When you use the SQL Server Installation Wizard, specify a unique instance name on the **Instance Name** page. When you run Setup from the command prompt, specify a unique instance name by using the **instancename** option.

▣ **See Also**

Other Resources

[Analysis Services Installation How-to Topics](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Install Analysis Services Side-by-Side with an Earlier Version



[See Also](#)



[Installing SQL Server 2005 Components](#) > [Installing SQL Server Components How-to Topics](#) > [Analysis Services Installation How-to Topics](#) >

You can install Microsoft SQL Server 2005 Analysis Services (SSAS) side-by-side with an instance of SQL Server 2000 Analysis Services. However the instance of SQL Server 2000 Analysis Services must always be the default instance, and the instance of SQL Server 2005 Analysis Services must always be a named instance, because SQL Server 2000 Analysis Services does not support named instances.

If you later remove the default instance of SQL Server 2000 Analysis Services, you can make an instance of SQL Server 2005 Analysis Services the default instance by using the Analysis Services Instance Rename tool. For more information, see [How to: Rename an Instance of Analysis Services](#).

Important:

This topic and the other Analysis Services installation how-to topics in this section focus only on installation options that are specific to Analysis Services. However, these topics provide links to other topics for more information about installation of SQL Server.

To Install Analysis Services Side-by-Side with an Earlier Version

1. Run Setup either by using the SQL Server Installation Wizard or from the command prompt. For more information about how to run Setup by using the SQL Server Installation Wizard, see [How to: Install Analysis Services by Using Setup](#). For more information about how to run Setup from the command prompt, see [How to: Install Analysis Services from](#)

[the Command Prompt.](#)

2. When you use the SQL Server Installation Wizard, specify a unique instance name on the **Instance Name** page. When you run Setup from the command prompt, specify a unique instance name by using the **instancename** option.

▣ **See Also**

Other Resources

[Analysis Services Installation How-to Topics](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Rename an Instance of Analysis Services



[See Also](#)



[Installing SQL Server 2005 Components](#) > [Installing SQL Server Components How-to Topics](#) > [Analysis Services Installation How-to Topics](#) >

You can rename an existing instance of Microsoft SQL Server 2005 Analysis Services (SSAS) by using the **Rename Instance** dialog box.

Important:

This topic and the other Analysis Services installation how-to topics in this section focus only on installation options that are specific to Analysis Services. However, these topics provide links to other topics for more information about installation of SQL Server.

Note:

On a 64-bit computer, you must use the 64-bit version of the Instance Rename tool.

Note:

The SQL Server 2005 Analysis Services (SSAS) Instance Rename tool is not supported for use in a cluster environment.

To rename an instance of Analysis Services

1. Launch the **Instance Rename** tool, **asinstancereName.exe**, from C:\Program Files\Microsoft SQL Server\90\Tools\Binn\VSShell\Common7\IDE.

2. In the **Rename Instance** dialog box, in the **Instance to rename** list, select the instance that you want to rename.
3. In the **New instance name** box, enter the new name for the instance.
4. Verify that the user name and password are correct, and then click **Rename**.

▣ **See Also**

Other Resources

Rename Analysis Services Instance Dialog Box (SSAS)

[Analysis Services Installation How-to Topics](#)

Managing and Maintaining Analysis Services

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Add or Remove Features for an Instance of Analysis Services

 [See Also](#)



[Installing SQL Server 2005 Components](#) > [Installing SQL Server Components How-to Topics](#) > [Analysis Services Installation How-to Topics](#) >

In Microsoft SQL Server 2005, feature maintenance is supported only through Add or Remove Programs in Control Panel in Microsoft Windows.

Important:

This topic and the other Analysis Services installation how-to topics in this section focus only on installation options that are specific to Analysis Services. However, these topics provide links to other topics for more information about installation of SQL Server.

To change components in an existing instance of SQL Server 2005 Analysis Services

1. On the Microsoft Windows desktop click **Start**, click **Control Panel**, and then double-click **Add or Remove Programs**.
2. Under **Currently installed programs**, select **Microsoft SQL Server 2005**, and then click **Change**. This will start the **Microsoft SQL Server 2005 Maintenance Wizard**.
3. On the **Component Selection** page, under SQL Server 2005 instances, select the instance that includes the instance of Analysis Services that you want to modify.
4. On the **Feature Maintenance** page, select Analysis Services. The **Microsoft SQL Server 2005 Maintenance Wizard** then launches the

Microsoft SQL Server 2005 Installation Wizard.

5. On the **Change or Remove Instance** page, click **Change Installed Components**.
6. Click **Next**, and then follow the instructions on subsequent wizard pages.

▣ **See Also**

Other Resources

[Analysis Services Installation How-to Topics](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Uninstall an Instance of Analysis Services



[See Also](#)



[Installing SQL Server 2005 Components](#) > [Installing SQL Server Components How-to Topics](#) > [Analysis Services Installation How-to Topics](#) >

Use the following steps to uninstall an instance of Microsoft SQL Server 2005 Analysis Services (SSAS).

Important:

This topic and the other Analysis Services installation how-to topics in this section focus only on installation options that are specific to Analysis Services. However, these topics provide links to other topics for more information about installation of SQL Server.

To remove an instance of SQL Server 2005 Analysis Services

1. On the Microsoft Windows desktop click **Start**, click **Control Panel**, and then double-click **Add or Remove Programs**.
2. Under **Currently installed programs**, select **Microsoft SQL Server 2005**, and then click **Change**. This will start the **Microsoft SQL Server 2005 Maintenance Wizard**.

Note:

If you click **Remove** instead of **Change**, you will not be able to uninstall an instance of Analysis Services without also uninstalling the SQL Server Database Engine and Reporting Services installed under the same instance name.

1. On the **Component Selection** page, under SQL Server 2005 instances, select the instance that includes the instance of Analysis Services that you

want to remove.

2. On the **Feature Maintenance** page, select Analysis Services. The **Microsoft SQL Server 2005 Maintenance Wizard** then launches the **Microsoft SQL Server 2005 Installation Wizard**.
3. On the **Change or Remove Instance** page, click **Remove Microsoft SQL Server**.
4. Click **Next**, and then follow the instructions on subsequent wizard pages.

▣ **See Also**

Other Resources

[Analysis Services Installation How-to Topics](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Reporting Services Installation How-to Topics



[See Also](#)



[Installing SQL Server 2005 Components](#) > [Installing SQL Server Components How-to Topics](#) >

This section contains procedures for installing Reporting Services.

▣ In This Section

- [How to: Install Reporting Services on a Single Computer](#)
- [How to: Install a Local Report Server and Remote Report Server Database](#)
- [How to: Install Reporting Services Side-by-Side with an Earlier Version](#)
- [How to: Uninstall an Instance of Reporting Services](#)
- [How to: Verify a Reporting Services Installation](#)
- [How to: Troubleshoot a Reporting Services Installation Problem](#)
- [How to: Troubleshoot Side-by-Side Installations of SQL Server 2000 and SQL Server 2005 Reporting Services](#)
- [How to: Configure IIS to Run 32-bit Reporting Services in WOW64 Mode](#)

▣ **See Also**

Concepts

[Upgrading Reporting Services](#)

[Installing SQL Server Reporting Services](#)

Other Resources

[Installing SQL Server Components How-to Topics](#)

[Remote Setup Information for SQL Server 2005](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Install Reporting Services on a Single Computer



[See Also](#)



[Installing SQL Server 2005 Components](#) > [Installing SQL Server Components How-to Topics](#) > [Reporting Services Installation How-to Topics](#) >

Use these procedures to install the complete set of client and server components on a single computer. In addition to client and server components, these instructions also include steps for installing the following components:

- The Database Engine, used for hosting the report server database and necessary for installing the report server in default configuration mode.
- The Reporting Services samples, useful for trying out Reporting Services features. Note that SQL Server 2005 setup only copies the samples setup program to the disk drive. After setup is complete, you must run the samples setup program to install the files and then upload the samples to the report server.

These instructions assume you are installing Reporting Services for the first time. If you are upgrading or reinstalling, you might not be able to use the default configuration installation option. In this case, you should run the Reporting Services Configuration tool and the Surface Area Configuration tool after setup is complete to finish configuring the report server. For more information, see [Upgrading Reporting Services](#).

Install client and server components on a single computer

1. Run Setup.
2. In the **Components to Install** page, select **SQL Server Database Services, Reporting Services, and Workstation components, Books Online, and development tools**.
3. Click **Advanced**.

4. In **Client Components**, open **Documentation, Samples, and Sample Databases**.
5. Click **Sample Databases**. You should select both AdventureWorks databases if you want to run all of the sample reports. Select **Entire feature will be installed on local hard drive**.
6. Click **Sample Code and Applications**. Select **Entire feature will be installed on local hard drive**. Click **Next**.
7. In the **Instance Name** page, select **Default instance**. Click **Next**.
8. In the **Service Account** page, specify a domain user account. In **Start services at end of setup**, select **SQL Server Agent**. Click **Next**.
9. In the **Authentication Mode** page, specify **Windows Authentication**. Click **Next**.
10. In the **Collation Settings** page, accept the default value. Click **Next**.
11. In the **Report Server Installation Options** page, select **Install the default configuration**. Click **Next**.
12. In the **Error and Usage Report Settings** page, click **Next**.
13. Click **Install** to begin installation.

▣ **Configure the Report Server and install samples**

1. Click **Start**, point to **All Programs**, point to **Microsoft SQL Server 2005**, point to **Configuration Tools**, click **SQL Server Surface Area Configuration**.
2. In Surface Area Configuration for Services and Connections, verify that the Report Server Windows service is running. In Surface Area Configuration for Features, verify that **Scheduled Events and Report Delivery** is enabled. Verify that **HTTP and Web Service Requests** is

enabled. Verify that **Windows Integrated Security** is enabled.

3. Verify that installation succeeded by launching Report Manager. In the URL address, type **https://localhost/reports** or **http://localhost/reports**. You should see the Report Manager Home page. If you installed the report server as a named instance, the URL will be different. For more information, see [Configuring Report Server Virtual Directories](#).
4. To install samples, click **Start**, point to **All Programs**, point to **Microsoft SQL Server 2005**, point to **Documentation and Tutorials**, point to **Samples**, click **Microsoft SQL Server 2005 Samples**. Use the Installation wizard to install the samples.
5. To publish the reports, use Business Intelligence Development Studio. Click **Start**, point to **All Programs**, point to **Microsoft SQL Server 2005**, click **Business Intelligence Development Studio**.
6. From the **File** menu, select **Open Project/Solution**.
7. From C:\Program Files\Microsoft SQL Server\90\Samples\Reporting Services\Report Samples\AdventureWorks Sample Reports, select AdventureWorks Sample Reports.sln. Click **Open**.
8. From the **Build** menu, select **Deploy Solution**. If the solution does not deploy, open each report and then retry the deployment. Status for a successful deployment is indicated in the Output window. You should see **Deploying to http://localhost/reportserver** in the Output window, and status for each report and data source uploaded to the server.
9. Open Report Manager. If it is already open, click the browser **Refresh** button to display the published items.
10. Open the AdventureWorks Samples folder. Click a report to view it.

 **Note:**

The Sales Reason Comparison sample report uses an Analysis Services database. The report has additional requirements that must be satisfied before you can run the report. For more information, see **AdventureWorks Report Samples**.

▣ **See Also**

Tasks

[How to: Install SQL Server 2005 \(Setup\)](#)

[How to: Verify a Reporting Services Installation](#)

Concepts

[Installing SQL Server Reporting Services](#)

Other Resources

Reporting Services Tutorials

SQL Server Reporting Services Samples

[Reporting Services Installation How-to Topics](#)

How to: Install Report Samples (Report Manager)

Reporting Services Configuration Tool

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Install a Local Report Server and Remote Report Server Database



[See Also](#)



[Installing SQL Server 2005 Components](#) > [Installing SQL Server Components How-to Topics](#) > [Reporting Services Installation How-to Topics](#) >

Hosting a report server database on a remote SQL Server Database Engine instance requires additional configuration steps after setup is complete. You must have administrator permissions on both the local and remote computers. Both computers must be in the same domain or in a trusted domain.

Note:

The following procedure is for a Windows Server 2003 operating system. If you are using a different operating system, the steps may vary.

Install a Local Report Server and Remote Report Server Database

1. Run Setup. For more information, see [How to: Install SQL Server 2005 \(Setup\)](#).
2. In the **Feature Selection** page, select **Reporting Services**. Click **Next**.
3. In the **Instance Name** page, select **Default Instance**. Click **Next**.
4. In the **Service Account** page, specify a domain user account. If you want to support scheduled operations, make sure that SQL Server Agent service starts automatically. In **Start services at end of setup**, select **SQL Server Agent**. Click **Next**.
5. In the **Authentication Mode** page, specify **Windows Authentication**.

Click **Next**.

6. In the **Report Server Installation Options** page, select **Install but do not configure the server**. Click **Next**.
7. In the **Error and Usage Report Settings** page, click **Next**.
8. Click **Install** to begin installation.

▣ **Configure the Report Server and Create the Remote Database**

1. Click **Start**, point to **All Programs**, point to **Microsoft SQL Server 2005**, point to **Configuration Tools**, click **SQL Server Surface Area Configuration**.
2. In Surface Area Configuration for Services and Connections, verify that the Report Server Windows service is running.
3. In Surface Area Configuration for Features, verify that **Scheduled Events and Report Delivery** is enabled. Verify that **HTTP and Web Service Requests** is enabled. Verify that **Windows integrated security** is enabled.
4. Open **Reporting Services Configuration**. Click **Start**, point to **All Programs**, point to **Microsoft SQL Server 2005**, point to **Configuration Tools**, click **Reporting Services Configuration**.
5. Select the local report server instance you just installed. Click **Connect**.
6. Configure the Web service account and virtual directories. If you are using Windows XP or Windows 2000 server, you must use the default ASP.NET account. If you are using Windows Server 2003, you can specify a custom account.
7. Configure the database.
 1. Open the Database Setup page.

2. In **Server Name**, select the remote SQL Server Database Engine instance, and then click **Connect**. You must have permission to create a database and configure roles on that server. The Reporting Services Configuration tool will use your credentials to create the database.
 3. Click **New** to create a new report server database. The SQL Server Connection dialog box opens. Type the name of the new database and the credentials used to create the database, and then click **OK**. If you already have a report server database you want to use, you can select it rather than create a new one. Click **Upgrade** if you want to upgrade a SQL Server 2000 report server database to the new schema. Upgrading the database is required if you want to use it with a SQL Server 2005 version of Reporting Services (note that you cannot reverse this step once the database has been upgraded; be sure that you have a backup copy of the database before you proceed).
 4. In **Credentials Type**, select the kind of account used to connect the report server to the report server database. You can use the service account credentials or a specific Windows domain account or SQL Server login.
 5. If you chose Windows Credentials or SQL Server Credentials, type the user name and password that the report server uses to connect to the report server database. For more information, see [Configuring a Report Server Database Connection](#).
 6. Click **Apply** to save your changes.
8. Configure other settings as needed, and then verify the installation. For more information, see [How to: Verify a Reporting Services Installation](#).

Note that some editions and releases of SQL Server 2005 have different default settings for remote connections. If the report server cannot connect to the database, verify that remote connections are enabled. For instructions, see the section "How to Configure Remote Connections to the Report Server Database" in **Configuring a Report Server for Remote Administration**.

▣ **See Also**

Tasks

[How to: Install SQL Server 2005 \(Setup\)](#)

Reference

[Setting Up Windows Service Accounts](#)

Concepts

[Installing SQL Server Reporting Services](#)

Other Resources

Troubleshooting Server and Database Connection Problems

[Reporting Services Installation How-to Topics](#)

Reporting Services Configuration Tool

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Install Reporting Services Side-by-Side with an Earlier Version



[See Also](#)



[Installing SQL Server 2005 Components](#) > [Installing SQL Server Components How-to Topics](#) > [Reporting Services Installation How-to Topics](#) >

You can install multiple versions of Reporting Services, where a SQL Server 2000 Reporting Services instance runs side-by-side with a SQL Server 2005 Reporting Services instance.

Note:

If you install the .NET Framework 2.0 or register ASP.NET 2.0 on a server that is running Internet Information Services (IIS) 5.0 or runs in IIS 5.0 compatibility mode, your installation of SQL Server 2000 Reporting Services might stop running. For more information about how to resume server operations, see [How to: Troubleshoot Side-by-Side Installations of SQL Server 2000 and SQL Server 2005 Reporting Services](#).

To run the two versions side-by-side, you must install the SQL Server 2005 version as a named instance, and continue to run the SQL Server 2000 version as the default instance. The SQL Server 2000 version does not support named instances and cannot be configured to run as one.

When running multiple versions of Reporting Services on the same computer, keep the following points in mind:

- You can use a SQL Server 2000 Database Engine to host a report server database for a SQL Server 2005 Reporting Services installation.
- The virtual directories must be unique for each report server and Report Manager instance. Consider adding the instance name or version information to each virtual directory name to distinguish each component (for example, ReportServer2005 and Reports2005 for the virtual directories you create for the SQL Server 2005 components).

- You can use the same Web site for all virtual directories. However, you should create a separate application pool for the SQL Server 2005 report server and Report Manager applications.
- Report namespaces and SOAP endpoints are different for each version. For more information, see **Reporting Services Programmability Enhancements** in SQL Server Books Online.
- Report definition (.rdl) files that you create in the SQL Server 2005 version of Report Designer can only be published to a SQL Server 2005 report server.
- Report definition (.rdl) files that you create in the SQL Server 2000 version of Report Designer can be published to either a SQL Server 2000 or SQL Server 2005 report server. If you publish a SQL Server 2000 .rdl file to a SQL Server 2005 report server, it will be converted and stored on the report server in the SQL Server 2005 format. If you open a SQL Server 2000 Reporting Services report in the SQL Server 2005 version of Report Designer, you will be prompted to upgrade it. For more information, see [Upgrading Reports](#).
- The Reporting Services Configuration tool cannot be used to configure a SQL Server 2000 report server. However, you can use Management Studio to connect to a SQL Server 2000 report server. Note that when you install different versions of Reporting Services, each report server installation has a corresponding Report Manager instance that can only be used with that installation. You can expect to see minor differences in the different versions of Report Manager.

You can run newer and earlier versions of Report Designer side-by-side. If you do so, be aware that different versions of Visual Studio must be used (or installed) for each instance:

- SQL Server 2000 Report Designer requires Visual Studio 2003. It does not run in Visual Studio 2005. You can publish reports from SQL Server 2000 Report Designer to a SQL Server 2005 report server.
- SQL Server 2005 uses Visual Studio 2005. If you do not have Visual Studio 2005 installed, Setup will install the version of the Visual Studio shell that is necessary for running Report Designer. You cannot use SQL Server 2005 Report Designer to publish reports to a SQL Server 2000 report server.

▣ To Install Reporting Services Side-by-Side With an Earlier Version

1. Run Setup for SQL Server 2005. When you run Setup, be sure to do the following:
 1. Specify a unique instance name on the **Instance Name** page or through the **instancename** option if you are running a command line installation.
 2. Specify a files-only installation for the report server. To do this, select the **Install but do not configure** option in the **Report Server Installation Options** page in the SQL Server Installation Wizard or **RSCONFIGURATION=filesonly** in a command line installation.
2. After Setup is finished, run the Reporting Services Configuration tool to deploy the new SQL Server 2005 instance.
3. Connect to the report server instance you just installed. To do so, specify the name of the server and the named SQL Server instance, and then click **Connect**.
4. On the Report Server Virtual Directory and the Report Manager Virtual Directory pages, specify a unique virtual directory for the report server and Report Manager. The names cannot be identical to those used in the SQL Server 2000 instance.

The Windows service and Web service accounts should be configured automatically when you specify the virtual directories. In some cases, automatic configuration might not succeed for the Web service account. If you are using Microsoft Internet Information Services (IIS) 6.0, you can resolve this error by specifying a different application pool.

5. On the Database Setup page, connect to the SQL Server instance that you want to use to host the report server database. The local Database Engine instance is specified by default. If you want to use another server, you must type the server name. Click **Connect** to connect to the server.

6. Click **New** to create a report server database. In the SQL Server Connection dialog box, type a database name. If you are connecting to a remote SQL Server, specify credentials that have permission to create a database. Click **OK**.
7. Optionally, specify credentials that the report server uses to connect to the database. Service account credentials are used by default. Click **Apply** to save the database configuration changes.
8. On the Encryption Keys page, create a backup copy of the keys.
9. To verify the new instance is operational, you can publish sample reports and then view them using Report Manager. Remember to specify the Report Manager virtual directory that you defined for the new instance. For step-by-step instructions on how to publish the sample reports, see the "Configure the report server and install samples" section in the topic **How to: Publish Report Samples (Report Designer)** in SQL Server Books Online.

▣ **See Also**

Tasks

[How to: Start Reporting Services Configuration](#)

Reference

[Instance Name](#)

Concepts

[Installing Reporting Services Using Setup](#)
[Configuring Report Server Virtual Directories](#)
[Installing SQL Server Reporting Services](#)

Other Resources

[Reporting Services Installation How-to Topics](#)
Reporting Services Configuration Tool

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Install the Windows SharePoint Services Object Model on a Report Server Computer

 [See Also](#)



[Installing SQL Server 2005 Components](#) > [Installing SQL Server Components How-to Topics](#) > [Reporting Services Installation How-to Topics](#) >

If you are configuring a report server to run in SharePoint integrated mode in a distributed server deployment that places a report server and a SharePoint technology instance on separate computers, you must install a local instance of the Windows SharePoint Services 3.0 object model for a farm installation on the report server computer. In a distributed server deployment, the report server uses the object model to access the SharePoint databases.

The object model cannot be installed as a separate component. However, you can install a Web front end to join the report server to a SharePoint farm. Configuring the Web front end provides connection information that the report server uses to connect to the farm.

Note:

A local instance of the object model is necessary only if you are integrating the server products in a distributed server topology that places each server on a separate computer. If you are integrating a report server and a SharePoint product or technology on the same computer (that is, the two server technologies will run side-by-side), you do not have to install the Web front end.

To install the Web front end, you must be a SharePoint farm administrator. You must also know the name of the database server and the configuration database that is used by the deployment of the SharePoint technology for which you are configuring report server integration.

To install the Windows SharePoint Services object model

1. On the server that runs Reporting Services, run Setup for Windows SharePoint Services 3.0.
2. On Installation Types, click **Advanced**.
3. On Server Type, select **Web Front End**.
4. Click **Install Now**.

▣ **To configure the Windows SharePoint Services object model**

1. Run the SharePoint Products and Technologies Configuration Wizard to join the Web front end to the remote SharePoint farm.
2. In the Connect to a Server Farm page, select **Yes** to connect to an existing server farm. Click **Next**.
3. In the Specify Configuration Database Settings page, specify the SQL Server Database Engine instance that hosts the configuration database. If you do not know the name of the configuration database, click **Retrieve Database Names** and then select the configuration database.
4. Enter a user name and password that the Web front end will use to connect to the configuration database. Click **Next**.
5. In the Completing the SharePoint Products and Technologies Wizard page, click **Advanced Settings**.
6. In Advanced Settings, verify that **Do not use this machine to host the Web site** is selected.
7. Click **OK**, click **Next**, and then finish the Wizard.

Installing the Web front end is just one step in the overall configuration process. For more information about the complete set of steps, see [Configuring Reporting Services for SharePoint 3.0 Integration](#).

▣ **See Also**

Concepts

[Reporting Services and SharePoint Technology Integration](#)

[Installing the Reporting Services Add-in](#)

[Deployment Modes for Reporting Services](#)

[Requirements for Running Reporting Services in SharePoint Integrated Mode](#)

[Configuring Reporting Services for SharePoint 3.0 Integration](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Uninstall the Reporting Services Add-in



[See Also](#)



[Installing SQL Server 2005 Components](#) > [Installing SQL Server Components How-to Topics](#) > [Reporting Services Installation How-to Topics](#) >

New: **12 December 2006**

You can uninstall the Reporting Services Add-in for SharePoint Technologies by running Setup in uninstall mode. Uninstalling the add-in will remove server integration features that are used to process reports and models on a report server.

The steps for uninstalling the add-in are the same for both stand-alone servers and server farms. Setup will remove program files and any configuration settings that were added during installation.

Setup will not remove the following:

- Logins created for the Report Server Web service account and Windows service account that were used to access the SharePoint configuration and content databases. You must delete these logins from the SQL Server Database Engine instance used to host the SharePoint databases.
- Permissions or groups that you created for report users. If you created custom permission levels or SharePoint groups to grant access to report server features, you should revoke any permissions that are no longer required.
- Data files that you uploaded to a SharePoint library, including report definition (.rdl), report model (.smdl), and shared data source (.rsds) files. They are not deleted, but they will no longer run. You must delete the files manually.

Setup will not delete the report server database or modify the report server instance that was used for integrated operations. After you uninstall the add-in, consider switching the report server to run in native mode if you want to use the instance as a stand-alone reporting application.

☐ **To uninstall the Reporting Services Add-in**

1. Remove any reports and other report server items that you no longer use. They will not run after the Reporting Services Add-in is removed.
2. In Control Panel, in **Add and Remove Programs**, select Microsoft SQL Server 2005 Reporting Services Add-in for Microsoft SharePoint Technologies.
3. Click **Remove**.

▣ **See Also**

Tasks

[How to: Switch Server Modes \(Reporting Services Configuration\)](#)

Concepts

[Installing the Reporting Services Add-in](#)

[Storing and Synchronizing Report Server Content With SharePoint Databases](#)

[Viewing and Managing Report Server Items from a SharePoint Site](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Uninstall an Instance of Reporting Services



[See Also](#)



[Installing SQL Server 2005 Components](#) > [Installing SQL Server Components How-to Topics](#) > [Reporting Services Installation How-to Topics](#) >

You can use Setup to uninstall SQL Server Reporting Services or remove specific components. Both the Installation Wizard and command line setup can be used to remove or modify an existing installation. If you choose to remove an entire installation, some files and accounts will remain after Setup is finished. Databases, log files, report definition files, model files, project files, solutions, and jobs waiting to be processed must be removed manually. The steps in this topic provide instructions on how to delete items that are not removed when you uninstall the software. To delete scheduled report processes that are currently in the queue, see **Modifying and Viewing Jobs**.

Note:

If you have installed 32-bit and 64-bit versions of Reporting Services on the same computer, be sure to use the 64-bit version of Setup to uninstall the 64-bit version of Reporting Services. You cannot use the 32-bit version of Setup to uninstall 64-bit Reporting Services.

To remove the report server databases

1. Use Management Studio or Enterprise Manager (if you are using SQL Server 2000 to host the database) to connect to the SQL Server Database Engine instance that hosts the databases.
2. Delete the databases. If you used default names, the database names are reportserver and reportservertemp. To verify the actual file names, you can run the Reporting Services Configuration tool and view the database name in the Database Setup page.

3. Delete the database role **RSExecRole** from Master and MSDB manually.

☐ **To delete data files**

1. Search the hard drive for the following file types:

Report definition (.rdl) files, semantic model (.smdl) files, shared data source (.rds) files, report project (.rptproj) files, solutions (.sln) files.

2. Delete the files.

☐ **To delete samples**

- Use **Add or Remove Programs** to uninstall the Reporting Services sample reports and applications. Setup uninstalls only the files it added. If additional user option files are created, or if you add new files to this folder, those files remain after the uninstall operation is finished. Sample files are located at Tools\Samples\<<lang>\ReportingServices.

☐ **To delete log files**

1. Navigate to the Log Files folder. Log files are located at Reporting Services\Log Files.
2. Delete the files.

▣ **See Also**

Other Resources

[Reporting Services Installation How-to Topics](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Verify a Reporting Services Installation



[See Also](#)



[Installing SQL Server 2005 Components](#) > [Installing SQL Server Components How-to Topics](#) > [Reporting Services Installation How-to Topics](#) >

When you install a report server using the default configuration, Setup installs and deploys the server. You can verify whether Setup deployed the report server by performing a few simple tests. You must be a local administrator to perform these steps. To enable other users to perform testing, you must configure report server access for those users. For more information, see **Server Deployment Checklist** in SQL Server Books Online.

Note:

The Reporting Services tutorials can help you learn important skills very quickly. There is also a tutorial that introduces each tool. It explains how to start the tool and shows you how each tool looks when it first opens. You can use this tutorial to confirm that all of the tools installed correctly. For more information, see **Reporting Services Tutorials** and **Tutorial: Reporting Services Tools** in SQL Server Books Online.

To verify that the report server is installed and running

1. Run the Reporting Services Configuration tool, connect to the report server instance you just installed, and review the status indicator for each setting to verify that it is configured.
2. Open the Services window and verify that the Report Server Windows service is running. To view the status of the Report Server Windows service, click **Start**, point to **Control Panel**, double-click **Administrative Tools**, and then double-click **Services**. When the list of services appears, scroll to **Report Server (MSSQLSERVER)**. The status should be **Started**.

3. Open a browser and type the report server URL in the address bar. The address consists of the server name and the virtual directory name that you specified for the report server during setup. By default, the report server virtual directory is named ReportServer. You can use the following URL to verify report server installation: `http://<computer name>/ReportServer<instance name>`. The URL will be different if you installed the report server as a named instance. For more information about the URL format, see [Configuring Report Server Virtual Directories](#).
4. Run reports to test report server operations. For this step, you can install and publish the sample reports if you do not have any reports available. For more information, see **AdventureWorks Report Samples** in SQL Server Books Online.

▣ **To verify that Report Manager is installed and running**

1. Open a browser and type the Report Manager URL in the address bar. The address consists of the server name and the virtual directory name that you specified for the Report Manager during setup. By default, the Report Manager virtual directory is Reports. You can use the following URL to verify Report Manager installation: `http://<computer name>/Reports<instance name>`.
2. Use Report Manager to create a new folder or upload a file to test whether definitions are passed back to the report server database. If these operations are successful, the connection is functional.

▣ **To verify that Report Designer is installed and running**

1. Open Business Intelligence Development Studio, and create a new project based on a Report Server project type. For more information on using the Report Server Project Wizard, see **How to: Create a Report Server Project (Report Designer)** in SQL Server Books Online.

2. If you installed report samples, open the sample report project files and publish the reports to a report server. For more information, see **Debugging and Publishing Reports** in SQL Server Books Online.
3. Run a script file (.rss) in the script environment to test Web service operations on the specified report server. For more information on running a sample script, see **Script Samples (Reporting Services)** in SQL Server Books Online.

▣ **See Also**

Other Resources

Troubleshooting HTTP Errors

Troubleshooting Server and Database Connection Problems

Cause and Resolution of Reporting Services Errors

[Reporting Services Installation How-to Topics](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Troubleshoot a Reporting Services Installation Problem



[See Also](#)



[Installing SQL Server 2005 Components](#) > [Installing SQL Server Components How-to Topics](#) > [Reporting Services Installation How-to Topics](#) >

The setup process for SQL Server 2005 Reporting Services supports two modes of installation: install a Reporting Services instance in the default configuration, or install a files-only installation that copies program files to disk. If you are encountering installation problems, consider using a files-only installation. Later, when you configure the report server for operation, you can adjust computer settings or the run time environment to support report server operations. If you are encountering installation problems with a files-only installation, use the information in this topic to resolve the problem.

Verify prerequisites

1. Internet Information Services (IIS) 5.0 or later must be installed.
2. On Windows Server 2003, IIS must not be in configured for IIS 5.0 isolation mode.
3. The IWAM_computername account must not be configured as a roaming profile.
4. MDAC 2.6 or later must be installed.
5. Distributed Transaction Coordinator (DTC) must be started.

ASP.NET must be registered with Internet Information Services

1. Run `aspnet_regiis -i` from the

%windir%\Microsoft.Net\Framework\v2.0.50110 folder.

2. Run **iisreset** to restart IIS.

☐ **Check the Setup log files**

- View the log files created by Setup. For more information, see [How to: View SQL Server 2005 Setup Log Files](#).

▣ **See Also**

Tasks

[How to: Troubleshoot Side-by-Side Installations of SQL Server 2000 and SQL Server 2005 Reporting Services](#)

Concepts

[Installing SQL Server Reporting Services](#)

Other Resources

Troubleshooting Reporting Services

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Troubleshoot Side-by-Side Installations of SQL Server 2000 and SQL Server 2005 Reporting Services

 [See Also](#)



[Installing SQL Server 2005 Components](#) > [Installing SQL Server Components How-to Topics](#) > [Reporting Services Installation How-to Topics](#) >

New: 5 December 2005

An installation of SQL Server 2000 Reporting Services that runs on Internet Information Services (IIS) 5.0 or IIS 5.0 isolation mode might stop running if you perform a side-by-side installation of SQL Server 2005 Reporting Services, or if you install the Microsoft .NET Framework version 2.0 or register ASP.NET 2.0 on the host computer. Errors that occur vary by operating system.

Operating system	Error message
Microsoft Windows XP	The report server cannot decrypt the symmetric key used to access sensitive or encrypted data in a report server database. You must either restore a backup key or delete all encrypted content and then restart the service.
Other Windows servers	Key not valid for use in specified state.

After you install SQL Server 2005 Reporting Services, perform the following steps to update your SQL Server 2000 version of Reporting Services installation.

 **To update a SQL Server 2000 Reporting Services installation**

1. Under <drive>:\Documents and Settings\
<machine_name>\ASPNET\Application Data\Microsoft\Crypto\RSA\
search for a file named '1aedd7b5699f3d6a88e354100b596aae*'.
2. Delete the file.
3. Run **RSActivate -r -c <rsreportserver.config file>**, specifying the
fully qualified path of the RSReportServer.config file of the SQL Server
2000 Reporting Services installation. If the path includes spaces, be sure
to enclose the entire path in quotation marks.

▣ **See Also**

Tasks

[How to: Install Reporting Services Side-by-Side with an Earlier Version](#)

[How to: Troubleshoot a Reporting Services Installation Problem](#)

Other Resources

[Reporting Services Installation How-to Topics](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Configure IIS to Run 32-bit Reporting Services in WOW64 Mode



[See Also](#)



[Installing SQL Server 2005 Components](#) > [Installing SQL Server Components How-to Topics](#) > [Reporting Services Installation How-to Topics](#) >

Updated: 5 December 2005

Internet Information Services (IIS) 6.0 returns a "Service Unavailable" error message when you run Reporting Services in Windows-on-Windows 64 (WOW64) x86 emulation mode. To avoid this error, you must configure IIS to allow 32-bit ASP.NET applications to run.

Important:

You must configure IIS before installing Reporting Services.

To configure IIS to run 32-bit Reporting Services

1. Uninstall the 64-bit version of Reporting Services. Side-by-side deployment of 64-bit and 32-bit versions is not supported.
2. Run dotnetfx64.exe to install the .NET Framework manually. You can find this file in the Redist folder on the SQL Server 2005 Setup media.
3. In IIS Manager, click the **Web Server Extensions** folder.
4. In the Details pane, right-click **ASP.NET V2.0.50727**, and then click **Allowed**.
5. Right-click the **Web Sites** folder and then select **Properties**.

6. Click **ISAPI Filters**.
7. In **Filter Name**, select **ASP.NET_2.0.50727**, and then click **Edit**.
8. Replace
"C:\WINDOWS\Microsoft.NET\Framework64\v2.0.50727\aspnet_filter.
with the 32-bit version that is located in the Framework folder:
C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727\aspnet_filter.dll
9. Click **OK**. Save your changes, and close IIS Manager.
10. Open a command prompt window, and then run the following
command:


```
cscript %SystemDrive%\inetpub\AdminScripts\adsutil.vbs set w3
```
11. Install the 32-bit version of Reporting Services.
12. After Setup is complete, open IIS Manager, click the **Web Server
Extensions** folder.
13. In the Details pane, right-click **ASP.NET V2.0.50727 (32-bit)**, and then
click **Allowed**.

▣ **See Also**

Other Resources

[Reporting Services Installation How-to Topics](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Integration Services Installation How-to Topics



[See Also](#)



[Installing SQL Server 2005 Components](#) > [Installing SQL Server Components How-to Topics](#) >

This section contains the following common procedures used for installing Microsoft SQL Server 2005 Integration Services.

For common procedures used for upgrading from Data Transformation Services (DTS) and for migrating DTS packages, see [Upgrading or Migrating Data Transformation Services How-to Topics](#).



Important:

The Integration Services installation how-to topics in this section focus only on installation options specific to Integration Services, but provide links to other topics that provide additional coverage of SQL Server installation.

▣ In This Section

- [How to: Install Integration Services by Using Setup](#)
- [How to: Install Integration Services from the Command Prompt](#)
- [How to: Add Integration Services to an Existing Instance of SQL Server 2005](#)
- [How to: Configure Integration Services on a Cluster](#)
- [How to: Ensure Support for Data Transformation Services Packages](#)

▣ **See Also**

Concepts

[Installing SQL Server Integration Services](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Install Integration Services by Using Setup



[See Also](#)



[Installing SQL Server 2005 Components](#) > [Installing SQL Server Components How-to Topics](#) > [Integration Services Installation How-to Topics](#) >

Updated: 17 July 2006

SQL Server 2005 Integration Services (SSIS) can be installed by using the SQL Server Installation Wizard, or at the command prompt. The Installation Wizard provides a graphical user interface that guides you through each installation decision, and is the preferred method for most users. The Installation Wizard provides guidance for initial setup of SQL Server 2005, including feature selection, instance naming rules, service account configuration, strong password guidelines, and collation selection.

Important:

This topic and the other Integration Services installation how-to topics in this section focus only on installation options specific to Integration Services, but provide links to other topics that provide additional coverage of SQL Server installation.

To install Integration Services by using Setup

1. Start the SQL Server Installation Wizard as described in [How to: Install SQL Server 2005 \(Setup\)](#) if you are installing only Integration Services or if you are installing Integration Services together with SQL Server 2005 components.

If you want to add Integration Services to an existing installation of SQL Server 2005, see [How to: Change Components in an Existing Instance of SQL Server 2005 \(Setup\)](#).

2. Optionally, review [How to: Install SQL Server 2005 \(Setup\)](#) for a page-by-page description of the Installation Wizard, especially if you are installing other SQL Server components in addition to Integration Services.
3. On the **Components to install** page of the Installation Wizard, select **Integration Services**. Review [Installing SQL Server Integration Services by Using Setup](#) for information about other SQL Server features, such as documentation, samples, and development tools.
4. Selecting **Integration Services** also installs support for SQL Server 2000 Data Transformation Services (DTS), including the ability to enumerate and run DTS packages. For more information, see [How to: Ensure Support for Data Transformation Services Packages](#).
5. Complete the Installation Wizard. After the **Components to install** page, the remaining pages of the wizard pertain to components of SQL Server 2005 other than Integration Services.

▣ **See Also**

Tasks

[Installing SQL Server Integration Services by Using Setup](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Install Integration Services from the Command Prompt



[See Also](#)



[Installing SQL Server 2005 Components](#) > [Installing SQL Server Components How-to Topics](#) > [Integration Services Installation How-to Topics](#) >

SQL Server 2005 Integration Services (SSIS) can be installed through use of the SQL Server Installation Wizard, or at the command prompt. The Installation Wizard provides a graphical user interface that guides you through each installation decision, and is the preferred method for most users. Command prompt installations are for advanced scenarios such as remote and silent installations; they can be run directly at the command prompt, or from command prompt syntax that references an .ini file to specify installation options.



Important:

This topic and the other Integration Services installation how-to topics in this section focus only on installation options specific to Integration Services, but provide links to other topics that provide additional coverage of SQL Server installation.

To install Integration Services from the command prompt

1. Choose whether to specify setup options at the command prompt or in an .ini file. For more information about how to use an .ini file to specify these options, see [How to: Install SQL Server 2005 from the Command Prompt](#).
2. Start **Setup.exe** at the command prompt and provide the required selections at the command prompt or in an .ini file as described in [How to: Install SQL Server 2005 from the Command Prompt](#). Review [Installing SQL Server Integration Services from the Command Prompt](#) for information about the installation options specific to Integration Services.

▣ **See Also**

Concepts

[Installing SQL Server Integration Services from the Command Prompt](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Add Integration Services to an Existing Instance of SQL Server 2005

 [See Also](#)



[Installing SQL Server 2005 Components](#) > [Installing SQL Server Components How-to Topics](#) > [Integration Services Installation How-to Topics](#) >

New: **12 December 2006**

In this release of Microsoft SQL Server, feature maintenance is supported only through Add or Remove Programs in Control Panel in Microsoft Windows.

To add Integration Services to an existing instance of SQL Server 2005

1. In Control Panel, double-click **Add or Remove Programs**.
2. Under **Currently installed programs**, click the instance of SQL Server to be configured, and then click **Change**.
3. On the **Component Selection** page, click the **To install a new component, click here** link.

Before clicking the link, you can click the **Report** button to view the list of SQL Server 2005 components and features that are installed on your computer. The report includes version, edition, update level, and language information for each installed component and feature.

4. In the **Insert Installation Media** dialog box, enter the location of setup.exe on the SQL Server 2005 installation media, or click **Browse** to locate the setup program.
5. In the SQL Server 2005 Installation Wizard, on the **Components to Install** page, select **Integration Services**, and then click **Next** to

continue.

For a description of each page of the wizard, see [How to: Install SQL Server 2005 \(Setup\)](#).

6. Complete the steps in the Installation Wizard.

For more information about Integration Services setup options, see [How to: Install Integration Services by Using Setup](#).

▣ **See Also**

Tasks

[How to: Install Integration Services by Using Setup](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Configure Integration Services on a Cluster



[See Also](#)



[Installing SQL Server 2005 Components](#) > [Installing SQL Server Components How-to Topics](#) > [Integration Services Installation How-to Topics](#) >

New: 5 December 2005

SQL Server 2005 Integration Services does not provide special installation options for a failover cluster. However, after installing Integration Services separately on each node of a cluster, you can manually configure it to operate as a clustered service.

Note:

After configuring the Integration Services service on a cluster, or on any server, you may need to configure DCOM permissions before you can connect to the service from a client computer. For more information, see **Connecting to a Remote Integration Services Server**.

To prepare to install Integration Services on a cluster

1. Install and configure a cluster with one or more nodes.
2. Optionally, install clustered services such as the SQL Server 2005 SQL Server Database Engine.
3. Install Integration Services on each node of the cluster individually.
4. Decide whether you will configure Integration Services as a clustered service:
 1. in the same group as the SQL Server 2005 SQL Server Database Engine.

2. in a different group than the SQL Server Database Engine.

▣ **Configuring Integration Services in the Same Group as SQL Server**

In this configuration:

- The loading and saving of packages to the MSDB database is faster.
- The MSDB node is available in Management Studio.

▣ **To configure Integration Services as a clustered service in the same group as SQL Server**

1. Open the Cluster Administrator.
2. In the console tree, select the Groups folder.
3. In the details pane, select the group to which you plan to add Integration Services.
4. On the **File** menu, point to **New** and then click **Resource**.
5. On the **New Resource** page of the **Resource Wizard**, type a Name and choose “Generic Service” as the Service Type. Do not change the value of **Group**. Click **Next**.
6. On the **Possible Owners** page, add or remove the nodes of the cluster as the possible owners of the resource. Click **Next**.
7. To add dependencies on the **Dependencies** page, select a resource under **Available resources**, and then click **Add**. In case of a failover, both SQL Server and the shared disk that stores Integration Services packages should come back online before Integration Services is brought online. After you have selected the dependencies, click **Next**.
8. On the **Generic Service Parameters** page, enter **MsDtsServer** as the name of the service. Click **Next**.

9. On the **Registry Replication** page, click **Add** to add the Registry key that identifies the configuration file for the Integration Services service. This file needs to be located on a shared disk that is in the same group for the Integration Services service, and fails over to the next node along with for the Integration Services service. In the **Registry Key** dialog box, type **SOFTWARE\Microsoft\MSDTS\ServiceConfigFile**. Click **OK**, then click **Finish**. The Integration Services service has now been added as a clustered service.
10. Now you must modify the location and the content of the service configuration file so that both the configuration file and the package store are available to all nodes in the case of a failover. Locate the configuration file at **%ProgramFiles%\Microsoft SQL Server\90\DTS\Binn\MsDtsSrvr.ini.xml**. Copy it to the shared disk.
11. Create a new folder named **Packages** on the shared disk. Grant List Folders and Write permissions on the new folder to the built-in Users group.
12. Open the configuration file from the shared disk in a text or XML editor. Change the value of the **ServerName** element to the name of the virtual SQL Server which is in the same group.
13. Change the value of the **StorePath** element to the fully-qualified path of the **Packages** folder created on the shared disk in a previous step.
14. Update the value of **HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSDTS\ServiceC** in the Registry to the fully-qualified path and filename of the service configuration file on the shared disk.
15. Now bring the clustered Integration Services service online. In the **Cluster Administrator**, select the Integration Services service, right-click, and select **Bring Online** from the popup menu. The Integration Services service is now online as a clustered service.

▣ **Configuring Integration Services in a Different Group than SQL Server**

In this configuration:

- The Integration Services Service may be online while the SQL Server 2005 SQL Server Database Engine is offline and the packages stored in its **msdb** database are unavailable.
- The Integration Services Service can be moved more quickly to another node if necessary.
- The Integration Services Service does not compete with other SQL Server services for CPU resources (when other SQL Server services are not installed on the same computer).

▣ **To configure Integration Services as a clustered service in a different group than SQL Server**

1. In the **Cluster Administrator**, select a group other than the group to which SQL Server belongs.
2. Add and configure IP Address, Network Name, and Generic Services resources for the Integration Services service. For the Generic Service, follow steps 1 through 9 of the previous procedure.
3. Place the service configuration file and the **Packages** folder on the shared disk for this group. Follow steps 10 through 14 of the previous procedure to update the location and content of the configuration file.
4. Bring the group online.

▣ **See Also**

Tasks

[How to: Install Integration Services by Using Setup](#)

[How to: Install Integration Services from the Command Prompt](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Ensure Support for Data Transformation Services Packages



[See Also](#)



[Installing SQL Server 2005 Components](#) > [Installing SQL Server Components How-to Topics](#) > [Integration Services Installation How-to Topics](#) >

SQL Server 2000 Data Transformation Services (DTS) packages can continue to run alongside SQL Server 2005 Integration Services (SSIS) packages.

You can run SQL Server 2000 Data Transformation Services (DTS) packages and Integration Services packages on the same computer.

When you select **Integration Services** for installation, Setup also installs support for SQL Server 2000 Data Transformation Services (DTS) packages, including the DTS runtime and DTS package enumeration in SQL Server Management Studio. Run-time support has been updated to enable DTS packages to access SQL Server 2005 data sources. If you are not installing Integration Services, but you need support for DTS packages, you must ensure that **Legacy Components** is selected on the **Feature Selection** page.

The backward compatibility files that are installed also include the SQL Server 2000 Analysis Services Processing task and its dependency, Decision Support Objects (DSO). However they do not include the SQL Server 2000 Data Mining Prediction Query task.

Note:

SQL Server 2005 does not install the DTS package designer for editing DTS packages. For more information, see [SQL Server 2005 Integration Services Backward Compatibility](#). DTS packages cannot be opened or edited in BI Development Studio.

When you select Integration Services for installation, the Package Migration Wizard is also installed. You can optionally use the Package Migration Wizard to

upgrade SQL Server 2000 DTS packages to the Integration Services format. During migration, the wizard copies the DTS packages and then re-creates them in the Integration Services format. The source packages are left intact and are not modified. For more information, see [Migrating Data Transformation Services Packages](#).

 **Note:**

The Package Migration Wizard is available in the Standard, Enterprise, and Developer Editions of SQL Server 2005.

SQL Server 2005 Express Edition does not include Integration Services or support for SQL Server 2000 DTS packages.

- To run existing DTS packages on a SQL Server 2005 Express Edition server, you must leave the SQL Server 2000 client tools or the DTS redistributable files on the server, or reinstall them. SQL Server 2005 Express Edition does not include the DTS runtime.
- To edit existing DTS packages on a SQL Server 2005 Express Edition server, you must use SQL Server 2000, or edit them remotely from a server running SQL Server 2005 Workgroup, Standard, Enterprise, or Developer Edition. SQL Server 2005 Express Edition does not include SQL Server Management Studio or Business Intelligence Development Studio.
- To migrate existing DTS packages to SQL Server 2005 Integration Services, you must use SQL Server 2005 Standard, Enterprise, or Developer Edition. Other editions of SQL Server 2005 do not include the Integration Services Package Migration Wizard.

The Import and Export Utility that is included with SQL Server 2005 Express Edition is not the SQL Server Import and Export Wizard and does not use Integration Services.

□ **To ensure support for Data Transformation Services packages**

1. Install SQL Server 2005, including Integration Services, by using the

preferred method. When you select Integration Services for installation, Setup also installs support for SQL Server 2000 Data Transformation Services (DTS) packages.

▣ **See Also**

Tasks

[How to: Install Integration Services by Using Setup](#)

[How to: Install Integration Services from the Command Prompt](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Notification Services Installation How-to Topics



[See Also](#)



[Installing SQL Server 2005 Components](#) > [Installing SQL Server Components How-to Topics](#) >

This section contains procedures for installing Notification Services.

▣ In This Section

- [How to: Install Notification Services and the Database Engine](#)
- [How to: Install the Notification Services Engine Components](#)
- [How to: Install the Notification Services Client Components](#)
- [How to: Register the Core Notification Services Assembly for COM Interop](#)
- [How to: Uninstall Notification Services](#)

▣ **See Also**

Concepts

[Installing SQL Server Notification Services](#)

[Migrating Notification Services Instances](#)

[Installing SQL Server 2005](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Install Notification Services and the Database Engine



[See Also](#)



[Installing SQL Server 2005 Components](#) > [Installing SQL Server Components How-to Topics](#) > [Notification Services Installation How-to Topics](#) >

When you want to run an instance of Notification Services on the same server that hosts its databases, you must install Notification Services and the Database Engine on that server. You typically use a single-server configuration when you develop applications and for deploying small to medium applications.

To install Notification Services and the Database Engine

1. On the **Start** page, click **Run the SQL Server Installation Wizard**.
2. Follow the Setup instructions until you see the **Components to Install** page.
3. Select **SQL Server Database Services** and **Notification Services**.

Optionally, you can select other components, such as Workstation components, Books Online, and development tools.

4. Follow the instructions on the remaining Setup pages.
5. On the **Ready to Install** page, click **Install**.

▣ **See Also**

Concepts

[Installing SQL Server Notification Services](#)

[Migrating Notification Services Instances](#)

Other Resources

[Notification Services Installation How-to Topics](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Install the Notification Services Engine Components



[See Also](#)



[Installing SQL Server 2005 Components](#) > [Installing SQL Server Components How-to Topics](#) > [Notification Services Installation How-to Topics](#) >

An instance of Notification Services and its databases can be located on separate servers. On the server that runs the instance of Notification Services, you must install Notification Services. Installing all Notification Services is equivalent to installing the Notification Services engine components.

To install Notification Services

1. On the **Start** page, click **Run the SQL Server Installation Wizard**.
2. Follow the Setup instructions until you see the **Components to Install** page.
3. Select **Notification Services**.

Optionally, you can select other components, such as Workstation components, Books Online, and development tools.

4. Follow the instructions on the remaining Setup pages.
5. On the **Ready to Install** page, click **Install**.

▣ **See Also**

Concepts

[Installing SQL Server Notification Services](#)

[Migrating Notification Services Instances](#)

Other Resources

[Notification Services Installation How-to Topics](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Install the Notification Services Client Components



[See Also](#)



[Installing SQL Server 2005 Components](#) > [Installing SQL Server Components How-to Topics](#) > [Notification Services Installation How-to Topics](#) >

Client applications that interact with instances of Notification Services require the Notification Services client components. The most common remote applications are the following:

- Subscription management interfaces.
- Non-hosted event providers.
- Remote administration consoles.

Note:

The Notification Services client components are included when you install Notification Services, the SQL Server engine components, or Microsoft SQL Server Management Studio.

To install rules components

1. On the **Start** page, click **Run the SQL Server Installation Wizard**.
2. Follow the Setup instructions until you see the **Components to Install** page.
3. On the **Components to Install** page, click the **Advanced** button.
4. On the **Feature Selection** page, expand **Notification Services**.
5. Under Notification Services, click **Client Components** and select **Will be installed on local hard drive**.

Optionally, you can select other SQL Server components.

6. Follow the instructions on the remaining Setup pages.
7. On the **Ready to Install** page, click **Install**.

▣ **See Also**

Concepts

[Installing SQL Server Notification Services](#)

[Migrating Notification Services Instances](#)

Other Resources

[Notification Services Installation How-to Topics](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Uninstall Notification Services



[See Also](#)



[Installing SQL Server 2005 Components](#) > [Installing SQL Server Components How-to Topics](#) > [Notification Services Installation How-to Topics](#) >

If you are permanently deleting Notification Services, you should delete your instances of Notification Services. If you are upgrading instances of Notification Services to a new version of Microsoft SQL Server, leave your instances in place and following the Notification Services upgrade instructions. You do not need to uninstall Notification Services 2.0 to migrate to SQL Server 2005.

To uninstall Notification Services

1. Stop all instances of Notification Services that use the version you are removing.
2. (Optional) Delete your instances of Notification Services.

Do not delete instances if you are upgrading them to a newer version of SQL Server. For more information about deleting instances, see **Deleting Instances of Notification Services**.

3. From the **Start** menu, select **Control Panel**, and then select **Add or Remove Programs**.
4. Select **Microsoft SQL Server 2005**, and then click **Change**.
5. On the **Component Selection** page, select **Notification Services**.
6. Click **Next**, and then follow the instructions on subsequent wizard pages.
7. When the **Change or Remove Instance** page appears, select **Change Installed Components**.

8. Click **Notification Services** and then select **Entire feature will be unavailable**.
9. Click **Next**, and then follow the instructions on subsequent wizard pages.

▣ **See Also**

Concepts

[Installing SQL Server Notification Services](#)

[Migrating Notification Services Instances](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Register the Core Notification Services Assembly for COM Interop

 [See Also](#)



[Installing SQL Server 2005 Components](#) > [Installing SQL Server Components How-to Topics](#) > [Notification Services Installation How-to Topics](#) >

The Notification Services API was developed in managed code. However, many classes in the Microsoft.SqlServer.NotificationServices.dll assembly can be used from unmanaged code through COM interop. Setup registers this assembly for COM interop. If you have to reregister this assembly, follow the instructions in the following section.

Other Notification Services assemblies do not support COM interop.

To register the core assembly for COM interop

1. Open a Command Prompt window.
2. Change the current directory to the Microsoft .NET Framework version 2.0 directory. In the following command, Replace N.N.N with the full .NET Framework version number.

```
CD %WINDIR%\Microsoft.NET\Framework\v N.N.N
```

3. Register the assembly using the following command:

```
regasm /codebase /tlb "%ProgramFiles%\Microsoft SQL Server\90\NotificationServices\9.0.242\bin\microsoft.sqlserver.notification
```

▣ **See Also**

Concepts

[Installing SQL Server Notification Services](#)

Other Resources

COM Interop with Notification Services

[Notification Services Installation How-to Topics](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Installing SQL Server Express



[_See Also](#)



While installing Microsoft SQL Server 2005 Express Edition (SQL Server Express), you have to make decisions to select the options that best suit your environment. The topics in this section can help you make those decisions.

▣ In This Section

Topic	Description
Hardware and Software Requirements (SQL Server Express)	Lists the hardware and software requirements for installing SQL Server Express.
Before You Install SQL Server Express	Provides pointers to installing the Microsoft .NET Framework, which is a prerequisite for SQL Server Express.
Using Command Prompt Options to Install SQL Server Express	Describes how to modify SQL Server Express Setup by using the command line. Points to the SQL Server 2005 Books Online topic that lists all the command line options to install SQL Server.
Installing Sample Databases for Express Editions	Provides download locations for installing the Adventureworks, Northwind, and pubs sample databases.
Upgrading MSDE to SQL Server Express	Describes the scenarios in which MSDE can be upgraded to SQL Server Express.
Setup User Interface Reference (SQL Server Express)	Provides instructions to install SQL Server Express.
SQL Server Surface Area Configuration	Documents the Surface Area Configuration user interface and provides default settings for each dialog box.

▣ **See Also**

Other Resources

[SQL Server 2005 Books Online Download Center](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Hardware and Software Requirements (SQL Server Express)



[See Also](#)



[Installing SQL Server Express](#) >

Updated: 17 July 2006

The following tables list the prerequisite software and minimum hardware and software requirements for running Microsoft SQL Server 2005 Express Edition (SQL Server Express), Microsoft SQL Server 2005 Express Edition with Advanced Services (SQL Server Express), and Microsoft SQL Server 2005 Express Edition Toolkit (SQL Server Express Toolkit).

Note:

To determine if the computer meets the system requirements, from the **Start** menu, right-click **My Computer**, and then click **Properties**. The **General** tab displays the CPU type and speed, and the amount of installed memory (RAM).

▣ System requirements (32-bit) for SQL Server Express

Prerequisite software	<ul style="list-style-type: none"> • Microsoft .NET Framework 2.0 • SQL Server Setup requires Microsoft Data Access Components (MDAC) 2.8 SP1 or later. You can download MDAC 2.8 SP1 from this Microsoft Web site.
Internet Requirements	Microsoft Internet Explorer 6.0 SP1 or later (prerequisite for .NET Framework)
RAM	<p>Minimum: 192 MB</p> <p>Recommended: 512 MB or higher</p>
Hard Disk space	600 MB free space
Processor	<p>Pentium III Compatible or higher</p> <p>Minimum: 500 MHz</p> <p>Recommended: 1 GHz or higher</p>
	<p>Windows Server 2003 SP1</p> <p>Windows Server 2003 Enterprise Edition SP1</p> <p>Windows Server 2003 Datacenter Edition SP1</p> <p>Windows Small Business Server 2003 Standard Edition SP1</p> <p>Windows Small Business Server 2003 Premium Edition SP1</p> <p>Windows XP Professional SP2</p> <p>Windows XP Home Edition SP2</p> <p>Windows XP Tablet Edition SP2</p> <p>Windows XP Media Edition SP2</p>

Operating System	Windows 2000 Professional Edition SP4
	Windows 2000 Server Edition SP4
	Windows 2000 Advanced Edition SP4
	Windows 2000 Datacenter Server Edition SP4
	Virtual PC
	Virtual Server
	Windows Vista Home Basic ²
	Windows Vista Home Premium ²
	Windows Vista Business ²
	Windows Vista Enterprise ²
	Windows Vista Ultimate ²
Note There is no support in this release for Windows XP Embedded Edition.	

²With Windows Vista, all SQL Server Express SP2 editions are supported. The only SP1 edition supported is SQL Server Express SP1.

▣ System requirements for SQL Server Express with Advanced Services

The following table lists the prerequisite software and minimum hardware and software requirements for running Microsoft SQL Server 2005 Express Edition with Advanced Services (SQL Server Express).

Prerequisite software	<ul style="list-style-type: none"> • Microsoft .NET Framework 2.0 • SQL Server Setup requires Microsoft Data Access Components (MDAC) 2.8 SP1 or later. You can download MDAC 2.8 SP1 from this Microsoft Web site.
Internet Requirements	<ul style="list-style-type: none"> • Microsoft Internet Explorer 6.0 SP1 or later (prerequisite for .NET Framework) • Internet Information Services (IIS) 5.0 or later is required for Microsoft SQL Server 2005 Reporting Services (SSRS) installations. For more information on how to install IIS, see How to: Install Microsoft Internet Information Services in SQL Server 2005 Books Online.
RAM	<p>Minimum: 512 MB</p> <p>Recommended: 1GB or higher</p>
Hard Disk space	600 MB free space
Processor	<p>Pentium III Compatible or higher</p> <p>Minimum: 600 MHz</p> <p>Recommended: 1 GHz or higher</p>
	<p>Windows Server 2003 SP1</p> <p>Windows Server 2003 Enterprise Edition SP1</p> <p>Windows Server 2003 Datacenter Edition SP1</p>

Operating
System

Windows Server 2003 Web Edition SP1

Windows Small Business Server 2003 Standard Edition SP1

Windows Small Business Server 2003 Premium Edition SP1

Windows XP Professional SP2

Windows XP Tablet Edition SP2

Windows XP Home Edition SP2¹

Windows XP Media Edition SP2¹

Windows 2000 Professional Edition SP4

Windows 2000 Server Edition SP4

Windows 2000 Advanced Edition SP4

Windows 2000 Datacenter Server Edition SP4

Virtual PC

Virtual Server

Windows Vista Home Basic²

Windows Vista Home Premium²

Windows Vista Business²

Windows Vista Enterprise²

Windows Vista Ultimate²



Note:

There is no support in this release for Windows XP Embedded Edition

¹Reporting Services, which is installed as part of SQL Server Express with Advanced Services, will not install on operating systems that do not include Internet Information Services (IIS).

²With Windows Vista, all SQL Server Express SP2 editions are supported. The only SP1 edition supported is SQL Server Express SP1.

▣ System Requirements for SQL Server 2005 Express Edition Toolkit SP2

Prerequisite software	<ul style="list-style-type: none"> • Microsoft .NET Framework 2.0 • SQL Server Setup requires Microsoft Data Access Components (MDAC) 2.6 or later. You can download MDAC 2.8 SP1 from this Microsoft Web site.
Internet Requirements	<ul style="list-style-type: none"> • Microsoft Internet Explorer 6.0 SP1 or later (prerequisite for .NET Framework) • Internet Information Services (IIS) 5.0 or later is required for Microsoft SQL Server 2005 Reporting Services (SSRS) installations. For more information on how to install IIS, see How to: Install Microsoft Internet Information Services in SQL Server 2005 Books Online.
RAM	<p>Minimum: 512 MB</p> <p>Recommended: 1GB or higher</p>
Hard Disk space	600 MB free space
Processor	<p>Pentium III Compatible or higher</p> <p>Minimum: 600 MHz</p> <p>Recommended: 1 GHz or higher</p>
	<p>Windows Server 2003 SP1</p> <p>Windows Server 2003 Enterprise Edition SP1</p> <p>Windows Server 2003 Datacenter Edition SP1</p> <p>Windows Server 2003 Web Edition SP1</p> <p>Windows Small Business Server 2003 Standard Edition SP1</p>

Operating System	Windows Small Business Server 2003 Premium Edition SP1
	Windows XP Professional SP2
	Windows XP Tablet Edition SP2
	Windows XP Home Edition SP2
	Windows XP Media Edition SP2
	Windows 2000 Professional Edition SP4
	Windows 2000 Server Edition SP4
	Windows 2000 Advanced Edition SP4
	Windows 2000 Datacenter Server Edition SP4
	Virtual PC
	Virtual Server
	Windows Vista Home Basic ²
	Windows Vista Home Premium ²
	Windows Vista Business ²
Windows Vista Enterprise ²	
Windows Vista Ultimate ²	
 Note: There is no support in this release for Windows XP Embedded Edition	

¹Reporting Services, which is installed as part of SQL Server Express with Advanced Services, will not install on operating systems that do not include Internet Information Services (IIS).

² With Windows Vista, all SQL Server Express SP2 editions are supported. The only SP1 edition supported is SQL Server Express SP1.

▣ **SQL Server Express Install Packages**

There are two executables available to install SQL Server Express:

- **SQLEXPRESS.EXE**. Installs on x86-based computers and supports WoW64 on x64 but not IA64.
- **SQLSERVER32.EXE**. Install on x86-based computers only. This program will not run on any 64-bit platform (not even WoW64).

▣ Change History

Release	History
5 December 2005	Changed content: <ul style="list-style-type: none">• Updated operating system list.• Updated the description of install packages.
14 April 2006	Changed content: <ul style="list-style-type: none">• Added operating system list for new editions of SQL Server Express.
17 July 2006	Changed content: <ul style="list-style-type: none">• Added MDAC as a software requirement to all three editions of SQL Server Express.
12 December 2006	Changed content: <ul style="list-style-type: none">• Added Windows Vista support

▣ **See Also**

Tasks

[How to: Install SQL Server Express](#)

Concepts

[Before You Install SQL Server Express](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Before You Install SQL Server Express

 [See Also](#)



[Installing SQL Server Express](#) >

Before you install Microsoft SQL Server 2005 Express Edition (SQL Server Express), make sure that you install the correct version of .NET Framework associated with SQL Server Express.

SQL Server Express Version	.NET Framework version
SQL Server 2005 Express Edition - RTM	<p>If you are installing SQL Server 2005 Express Edition, perform the following steps in order:</p> <ol style="list-style-type: none"><li data-bbox="488 1037 1382 1205">1. Uninstall .NET Framework 1.2 and later versions from your computer (versions 1.0 and 1.1 do not have to be uninstalled). Then, install .NET Framework 2.0. from .NET Framework 2.0 Download Center.<li data-bbox="488 1209 1382 1293">2. Install SQL Server 2005 Express Edition from the SQL Server Express Web site.

▣ **How to Identify the .NET Framework Version on your Computer**

You can identify which .NET Framework version is on your computer in one of two ways:

- If the version is 2.0, to identify the exact version, go to %WINDIR%\Microsoft.NET\Framework\version, right-click mscorlib.dll, click **Properties**, and then click **Version**.
- Alternatively, on the **Start** menu, click **Control Panel**, open **Administrative Tools**, and then open **.NET Framework 2.0 Configuration**. At the top of the right pane, the .NET Framework version displays.

▣ **See Also**

Tasks

[How to: Install SQL Server Express](#)

[How to: Uninstall SQL Server Express](#)

Other Resources

[Installing SQL Server Express](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Using Command Prompt Options to Install SQL Server Express



[See Also](#)



[Installing SQL Server Express](#) >

SQL Server 2005 Express Edition (SQL Server Express) Setup provides a command prompt interface in addition to the graphical user interface. See [Running Setup from the Command Prompt](#) and [How to: Install SQL Server 2005 from the Command Prompt](#) in SQL Server 2005 Books Online to customize how Setup installs SQL Server Express.

Note:

SQL Server Express is a limited edition of SQL Server 2005. The setup experience using the command prompt for SQL Server Express is similar to that of SQL Server 2005.

▣ **Modifying SQL Server Express Installed Components**

If you install SQL Server Express from the download center on the [SQL Server Express Web site](#), you will use `sqlexpr.exe` to run Setup. If `sqlexpr.exe` is run directly without saving to a folder, the temporary directory where the binary files are extracted is deleted after the installation is complete. If you try to modify the SQL Server Express components by using **Add or Remove Programs** in Control Panel, the **Change** button will look for a setup directory but will not find it.

This scenario is not true when SQL Server Express is installed as part of Visual Studio 2005 Express. For more information about Visual Studio Express, see [Visual Studio 2005 Express Products](#).

To modify the installed SQL Server Express components, you will have to download `sqlexpr.exe` again from the Web site, and save the `.exe` to a folder. At the command prompt, change the path to the folder where the `.exe` is stored, and run Setup by using the `sqlexpr.exe -x` parameter. You will be prompted to select an extraction/installation directory.

When you have extracted the Setup files, you can use the **Change** button to make modifications to your SQL Server Express installation by specifying the extraction directory when prompted.

▣ **Running SQL Server Express in a Single User Mode**

In the following scenario, SQL Server Express will not start in single user mode:

After installing SQL Server Express, you start the service, verify the connection, and then stop the service by using SQL Server Configuration Manager. You then go to the command prompt and start SQL Server Express in single user mode by using the following command:

```
Sqlservr.exe -s SQLEXPRESS -m
```

The service will start. However, if you shut down the service and try to start it again, the service will not start.

▣ **See Also**

Tasks

[How to: Install SQL Server Express](#)

[How to: Uninstall SQL Server Express](#)

Other Resources

SQLCMD Utility Tutorial

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Installing Sample Databases for Express Editions



[See Also](#)



[Installing SQL Server Express](#) >

The sample databases that are supported by Microsoft SQL Server 2005 Express Edition (SQL Server Express) are Adventureworks, Northwind, and Pubs. By default, these sample databases are not installed with SQL Server Express. Instead, you can download them from the Microsoft Download Center.

▣ **AdventureWorks**

Adventureworks can be installed from this [Microsoft Web site](#). Download `SQLServerDatabasesandSamplesOverview.htm` and follow the instructions that it provides pertaining to SQL Server Express. Double-click `AdventureworksDB.msi` to install the sample database.

▣ **Northwind and pubs**

The Northwind and pubs can be installed from this [Microsoft Web site](#).

After downloading SQL2000SampleDb.msi, extract the sample database scripts by double-clicking SQL2000SampleDb.msi. SQL2000SampleDb.msi will extract the database scripts and a Readme file into the folder C:\Program Files\SQL Server 2000 Sample Databases. Follow the instructions in the Readme file to run the installation scripts.

📌 **Note:**

For more information, see **Samples and Sample Databases** in SQL Server 2005 Books Online.

▣ **See Also**

Concepts

[Running Setup to Install AdventureWorks Sample Databases and Samples](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Upgrading MSDE to SQL Server Express



[Installing SQL Server Express](#) >

If MSDE was installed with an MSI setup, the existing MSDE instance is upgraded when you install Microsoft SQL Server 2005 Express Edition (SQL Server Express) with the same instance name. Use the **Default instance** option when you run SQL Server Express Setup. The exception to this is if the version of SQL Server Express you are installing is not English. Then, the MSDE installation you are upgrading must either be the same language as the SQL Server Express version, or it must be English.

If MSDE was installed as part of another application that used its own install program, the SQL Server Express installation program will not know about it. In this scenario, the only way to upgrade SQL Server Express is to install SQL Server Express under another instance name. Use the **Named Instance** option to specify a new instance name when you run SQL Server Express Setup, and then detach the databases from MSDE and attach them to SQL Server Express.

To know if MSDE was installed as part of another application, go to **Add or Remove Programs** in **Control Panel**. If MSDE does not appear in **Add or Remove Programs**, it was installed as part of another application. In this scenario, the only way to remove MSDE is to uninstall the application that installed MSDE. In most situations, the application users will let the application vendor deal with the upgrade.

For more information about upgrading, see "[Upgrading MSDE 2000 to SQL Server 2005 Express](#)" at MSDN library.

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Enable Network Access During Installation (SQL Server Express)



[Installing SQL Server Express](#) >

When you install Microsoft SQL Server 2005 Express Edition (SQL Server Express), network protocols are disabled by default. You can enable protocols during installation or after the product is installed. You might also need to enable firewall settings.

To enable networking protocols

- When running setup.exe from the command prompt, specify `DisableNetworkProtocols = 0`, as in the following example:

```
setup.exe /qn addlocal=all InstanceName=SQLEXPRESS DisableNetv
```

To enable firewall settings

1. Click **Start**, click **Control Panel**, and then click **Network Connections**.
2. From the navigation bar on the left, click **Change Windows Firewall settings**.
3. On the **Exceptions** tab, in the **Programs and Services** box, you will probably see that SQL Server is listed, but not selected as an exception. If you select the checkbox, Windows will open the 1433 port to let in TCP requests. Alternatively, if you do not see SQL Server listed, do the following:
 1. Click **Add Program**.

2. Click **Browse**.
3. Navigate to
drive:/Program Files/Microsoft SQL Server/MSSQL.1\MSSQL\BIN
4. Add the file sqlservr.exe to the list of exceptions.

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Install SQL Server Express



[See Also](#)



[Installing SQL Server Express](#) >

This topic describes how to install Microsoft SQL Server 2005 Express Edition (SQL Server Express). It also identifies the setup options that differ from Microsoft SQL Server 2005.

For local installations, you must run Setup as an administrator. If you install SQL Server from a remote share, you must use a domain account that has read and execute permissions on the remote share.

To install SQL Server Express

1. To begin installation, double-click sqlexpr.exe.

Note:

SQL Server Express is available as a self-extracting executable file (sqlexpr.exe) from both the Visual Studio 2005 CD and the [SQL Server Express Web site](#). SQLEXPRESS.EXE installs on x86-based computers and to the 32-bit subsystem (WOW64). For more information, see [Hardware and Software Requirements \(SQL Server Express\)](#)

2. On the End User License Agreement page, read the license agreement, and then select **I accept the licensing terms and conditions** check box. Click **Next**.
3. **SQL Server Component Update** runs, installing the software required prior to installing SQL Server Express. For more information about component requirements, click the **Help** button at the bottom of the page. Click **Next**.

 **Important:**

.NET Framework 2.0 is not installed when **SQL Server Component Update** runs, but SQL Server Express Setup requires .NET Framework 2.0 to be installed prior to running the sqlexpr.exe. If .NET Framework 2.0 is not present, you will encounter an error. Make sure that you uninstall all previous versions of .NET Framework and install .NET Framework 2.0. from the [.NET Framework 2.0 Download Center](#) before initiating SQL Server Express Setup.

4. The Welcome to the Microsoft SQL Server Installation Wizard page appears. Click **Next**.
5. On the System Configuration Check page, the computer is scanned for potential installation problems. To interrupt the scan, click **Stop**. To proceed with setup after the scan completes, click **Continue**.
6. On the Registration Information page, enter information in the **Name** and **Company** text boxes. Click **Next**.
 1. Because SQL Server Express is free, there is no **Product Key** text box on this page.
 2. If you select the **Hide advanced configuration options** check box, the Service Account, Collation Settings, Instance Name, and User Instances pages do not appear during setup.
7. On the Feature selection page, select the program features to install and click **Next**.

 **Note:**

There is no Custom Feature Selection page in SQL Server Express Setup. Instead, the Feature Selection page lets you select the features that you want to be installed. Expand a feature to view its subfeatures by clicking the "+" symbol.

8. On the Instance Name page, select a **Default instance** or a **Named instance** for your installation. If you select **Default instance**, an existing default instance will be upgraded. If you select **Named Instance**, specify an instance name or the default named instance of

SQLExpress is used. Click **Next**.

Click **Installed Instances** to view a list of installed SQL Server Express instances and components on the computer where SQL Server Express Setup is running. The installed instance details pane will display upgrade options for installed instances.

SQL Server Express Setup checks whether the instance name *SQLExpress* exists at startup.

1. If *SQLExpress* instance exists, then by default, the Instance name page appears.
 2. If *SQLExpress* instance does not exist, then by default, the Instance name page is hidden from the user. However, if you clear the **Hide Advanced configuration options** check box on the Registration Information page, the **Instance name** dialog box appears at all times.
9. The **Service Account** page lets you assign local system account or a domain user account to SQL Server service. This page does not appear if you select the **Hide advanced configuration options** check box
1. Select **Use a Local System account** that does not require a password to connect to SQL Server on the same computer, and then click **Next**.
 2. Select **Use a Domain User** account, enter the **Username**, **Password**, and the **Domain** name, and then click **Next**.
 Note:
The **Start services at the end of SQL Server Setup** checkbox is selected by default for SQL Server Express. It automatically starts the SQL Server service when the operating system starts.
10. On the **Authentication Mode** page, specify the security mode used to connect to your instance of SQL Server. If you select Windows Authentication, Setup creates an sa account that is disabled by default. To activate the sa account after Setup completes, see the **How to: Change Server Authentication Mode** and **ALTER LOGIN (Transact-SQL)** topics in SQL Server 2005 Books Online. If you select

Mixed Mode Authentication you must set strong passwords for all SQL Server logins. Enter and confirm a strong sa logon password. Click **Next**.

 **Important:**

For security reasons, SQL Server Express does not automatically start the SQL Server Browser service. If your applications have to use SQL Server Browser, you must manually start the service or change its startup option. For more information, see **SQL Server Browser Service How to: Start and Stop the SQL Server Browser Service** in SQL Server 2005 Books Online.

11. On the **Collations Settings** page, change the default **Collation designator and sort order** only if you must match collation settings in another instance of SQL Server or in another computer. Select **SQL collations** to match the sort order settings in earlier versions of SQL Server. Click **Next**. This page does not appear if you select the **Hide advanced configuration options** check box.

 **Note:**

The **Customize for each service account** drop-down box is removed because in SQL Server Express the service should always point to SQL Server.

12. On the **User Instances** page, specify if you want to generate a separate instance of the Database Engine for non-administrator users. By default, user instance functionality is enabled. To turn off user instances, clear the check box. This page does not appear if you select the **Hide advanced configuration options** check box. Click **Next**.
13. On the **Error and Usage Report Setting** page, select whether to turn on error reporting and usage reporting for SQL Server and its components. By default, error reporting is turned on. To turn off error reporting, clear the check box. Click **Next**.
14. On the **Ready to Install** page, click **Install** to finish installing SQL Server.

15. On the **Setup Progress** page, you can monitor installation progress as setup proceeds. Click **Next**.
16. On the **Completing the Microsoft SQL Server Installation Wizard** page, you can view the setup summary log by clicking the link provided on this page. To complete the SQL Server Installation Wizard, click **Finish**.

In addition to using graphical user interface, you can install SQL Server Express by using the same command line parameters that are used to install SQL Server 2005. For more information about command line parameters, see "[Running Setup from the Command Prompt](#)" in SQL Server 2005 Books Online.

SQL Server Express also supports the `DISABLENETWORKPROTOCOLS` property to enable network access during installation. For more information, see [How to: Enable Network Access During Installation \(SQL Server Express\)](#).

▣ **See Also**

Tasks

[How to: Uninstall SQL Server Express](#)

Concepts

[Before You Install SQL Server Express](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Uninstall SQL Server Express



[Installing SQL Server Express](#) >

In this topic, you will learn how to completely uninstall Microsoft SQL Server 2005 Express Edition (SQL Server Express).

To Uninstall SQL Server Express

1. Click **Start**, and then click **Control Panel**.
2. Open **Add or Remove Programs**.
3. Select **Microsoft SQL Server 2005**, and then click **Remove**.
4. The Component Selection page appears. By default, the **Remove SQL Server 2005 instance components** check box is selected. Select the instance that you want to remove.
5. To remove the Workstation components, select the **Workstation Components** check box.
6. The **Confirmation** page appears, listing the components that will be removed. Click **Finish** to complete the process of uninstalling SQL Server Express.

You will see the System Configuration Check and Setup Progress pages as the uninstall process is completed.

7. To remove SQL Native Client, click **Start**, click **Control Panel**, and then click **Add or Remove Programs**.
8. Select **Microsoft SQL Native Client**, and then click **Remove**.
9. You will see a dialog box confirming whether you want to remove SQL Native Client. Click **Yes**.

SQL Native Client is removed from the list of **Add or Remove Programs**.

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Configuring a SQL Server Installation

 [See Also](#)



After Setup completes the installation of Microsoft SQL Server 2005, you can further configure SQL Server using graphical and command-prompt utilities. The following table describes support for tools used to manage an instance of SQL Server 2005.

Tool or utility	Description
SQL Server Management Studio	SQL Server Management Studio is used for editing and executing queries, and for launching standard wizard tasks. For more information, see Introducing SQL Server Management Studio .
SQL Server Profiler	SQL Server Profiler provides a graphical user interface for monitoring an instance of the SQL Server Database Engine or an instance of Analysis Services. For more information, see Introducing SQL Server Profiler .
Database Engine Tuning Advisor	Database Engine Tuning Advisor helps create optimal sets of indexes, indexed views, and partitions. For more information, see Using Database Engine Tuning Advisor .
Business Intelligence Development Studio	Business Intelligence Development Studio is an integrated development environment for Analysis Services and Integration Services solutions. For more information, see Introducing Business Intelligence Development Studio .
Command Prompt Utilities	Manage SQL Server objects from the command prompt. For more information, see Command Prompt Utilities .
SQL Server Configuration Manager	Manage server and client network configuration settings. For more information, see SQL Server Configuration Manager .
Import and Export Data	Integration Services provides a set of graphical tools and programmable objects for moving, copying, and transforming

	data. For more information, see Integration Services Overview .
SQL Server Setup	Install, upgrade to, or change components in an instance of SQL Server 2005. For more information, see Database Engine Installation How-to Topics .

Default Configuration

Default configurations for new instances of SQL Server 2005 disable some features and components to reduce the attackable surface area of the product.

The following components and features are off by default:

- Integration Services
- SQL Server Agent
- SQL Server Browser
- Full-Text Search

For more information, see [SQL Server Surface Area Configuration](#).

▣ **See Also**

Reference

[Features Supported by the Editions of SQL Server 2005](#)

Other Resources

Administering the Database Engine

Automating Administrative Tasks (SQL Agent)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

SQL Server Surface Area Configuration



[See Also](#)



[Configuring a SQL Server Installation](#) >

Updated: **17 July 2006**

Surface area reduction is a security measure that involves stopping or disabling unused components. Surface area reduction helps to improve security by providing fewer avenues for potential attacks on a system.

For new installations of Microsoft SQL Server 2005, some features, services, and connections are disabled or stopped to reduce the SQL Server surface area. For upgraded installations, all features, services, and connections remain in their pre-upgrade state.

Use SQL Server Surface Area Configuration to enable, disable, start, or stop the features, services, and remote connectivity of your SQL Server 2005 installations. You can use SQL Server Surface Area Configuration on local and remote servers.

SQL Server Surface Area Configuration uses Window Management Instrumentation (WMI) to view and change server settings. WMI provides a unified way for interfacing with the API calls that manage registry operations that configure SQL Server. For information about configuring permissions related to WMI, see the topic **How to: Configure WMI to Show Server Status in SQL Server Tools**.

▣ **When to Use the Surface Area Configuration Tools**

After you install SQL Server 2005 or upgrade to SQL Server 2005, you should run SQL Server Surface Area Configuration to verify which features and services are enabled and running, and to verify which types of connections SQL Server 2005 will accept. After initial configuration, you can use SQL Server Surface Area Configuration to verify or change the state of features, services, and connections.

▣ **Launching Surface Area Configuration**

SQL Server Surface Area Configuration is available on the SQL Server **Start** menu:

- On the **Start** menu, point to **All Programs, Microsoft SQL Server 2005, Configuration Tools**, and then click **SQL Server Surface Area Configuration**.

The first page to appear is the SQL Server Surface Area Configuration start page. On the start page, specify which server you want to configure:

1. Click the **change computer** link adjacent to **Configure Surface Area for**. The default value is **localhost**. If you previously selected a named server, you would see the server name.
2. In the **Select Computer** dialog box, do one of the following:
 - To configure SQL Server 2005 on the local computer, click **Local computer**.
 - To configure SQL Server 2005 on another computer, click **Remote computer**, and then enter the computer name in the text box.
 - To configure a failover cluster, click **Remote computer**, and then enter the failover cluster instance name in the text box.
3. Click **OK**.

▣ **Using the Surface Area Configuration Tools**

After selecting the computer to configure, you can launch two tools:

- Use Surface Area Configuration for Services and Connections to enable or disable Windows services and remote connectivity. For descriptions of the service and connectivity settings and defaults for those settings, see [Surface Area Configuration for Services and Connections](#).
- Use Surface Area Configuration for Features to enable and disable features of the Database Engine, Analysis Services, and Reporting Services. For descriptions of the features and information about default feature settings, see [Surface Area Configuration for Features](#).

▣ **sac Command Line Utility**

To import and export surface area settings, use the **sac** command-prompt utility. Using this utility, you can configure the surface area on one computer, and then apply the same settings to other computers.

The easiest way to use the **sac** utility is to use SQL Server Surface Area Configuration to configure one computer, and then use the **sac** utility to export the settings of that computer to a file. You can use that file to apply the same settings to SQL Server 2005 components on other computers.

For more information, see [sac Utility](#).

▣ Change History

Release	History
17 July 2006	Changed content: <ul style="list-style-type: none">• Updated the introduction to state why it is good to reduce the surface area of a system.• Added section about when to use the surface area configuration tools.

▣ **See Also**

Concepts

[Surface Area Configuration for Features](#)

[Surface Area Configuration for Services and Connections](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Surface Area Configuration for Features



[See Also](#)



[Configuring a SQL Server Installation](#) > [SQL Server Surface Area Configuration](#) >

The Surface Area Configuration for Features tool provides a single interface for enabling or disabling many Database Engine, Analysis Services, and Reporting Services features. Disabling unused features helps to secure your Microsoft SQL Server installations by reducing the SQL Server surface area.

The tables below list the default for each setting and provide links to F1 Help topics.

▣ Database Engine Feature Defaults

Setting	Default value	F1 Help topic
Ad hoc distributed queries	Disabled in new installations.	Surface Area Configuration for Features (Ad Hoc Remote Queries) - Database Engine
Common language runtime (CLR) integration	Disabled in new installations.	Surface Area Configuration for Features (CLR Integration) - Database Engine
Dedicated administrator connection (DAC)	Disabled in new installations.	Surface Area Configuration for Features (DAC) - Database Engine
Database Mail	Disabled in new installations.	Surface Area Configuration for Features (Database Mail) - Database Engine
Native XML Web services	Endpoints are not configured by default.	Surface Area Configuration for Features (Native XML Web Services) - Database Engine
OLE Automation stored procedures	Disabled in new installations.	Surface Area Configuration for Features (OLE Automation) - Database Engine
Service Broker	Endpoints are not configured by default.	Surface Area Configuration for Features (Service Broker) - Database Engine
SQL Mail	Disabled in new installations.	Surface Area Configuration for Features (SQL Mail) - Database Engine
Web Assistant stored procedures	Disabled in new installations.	Surface Area Configuration for Features (Web Assistant) - Database Engine
xp_cmdshell	Disabled in new installations.	Surface Area Configuration for Features (xp_cmdshell) - Database Engine

▣ Analysis Services Feature Defaults

Setting	Default value	F1 Help topic
Ad hoc data mining queries	Disabled in new installations.	Surface Area Configuration for Features (Ad Hoc Data Mining Queries) - Analysis Services
Anonymous connections	Disabled in new installations.	Surface Area Configuration for Features (Anonymous Connections) - Analysis Services
Linked objects	Disabled in new installations.	Surface Area Configuration for Features (Linked Objects) - Analysis Services
User-defined functions	Disabled in new installations.	Surface Area Configuration for Features (User-Defined Functions) - Analysis Services

▣ Reporting Services Feature Defaults

Feature	Default setting	F1 Help topic
Web service requests and HTTP access	Enabled in new installations.	Surface Area Configuration for Features (Web Service and HTTP Access) - Reporting Services
Scheduled events and report delivery	Enabled in new installations.	Surface Area Configuration for Features (Scheduled Events and Report Delivery) - Reporting Services
Windows Integrated Security	Enabled in new installations	Surface Area Configuration for Features (Windows Integrated Security) - Reporting Services

▣ **See Also**

Concepts

[SQL Server Surface Area Configuration](#)

[Configuring a SQL Server Installation](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Surface Area Configuration for Features (Ad Hoc Remote Queries) - Database Engine

 [See Also](#)



[Configuring a SQL Server Installation](#) > [SQL Server Surface Area Configuration](#) > [Surface Area Configuration for Features](#) >

Ad hoc distributed queries use the OPENROWSET and OPENDATASOURCE functions to connect to remote data sources using OLE DB. Use this dialog to configure support for ad hoc remote queries.

▣ Options

Enable OPENROWSET and OPENDATASOURCE support

If checked, the instance of the Database Engine supports the use of the OPENROWSET and OPENDATASOURCE functions to run remote queries without first defining a linked server. OPENROWSET and OPENDATASOURCE are designed for infrequent queries. If your applications do not use these functions, you should not enable them.

For more information about ad hoc queries, see **Identifying a Data Source Using an Ad Hoc Name** in SQL Server Books Online.

For more information about configuring server options, see **sp_configure (Transact-SQL)** in SQL Server Books Online.

▣ **See Also**

Concepts

[Surface Area Configuration for Features](#)

[Surface Area Configuration for Services and Connections](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Surface Area Configuration for Features (CLR Integration) - Database Engine

 [See
Also](#)



[Configuring a SQL Server Installation](#) > [SQL Server Surface Area Configuration](#) > [Surface Area Configuration for Features](#) >

Execution of user assemblies inside Microsoft SQL Server 2005 is disabled by default. Use this dialog to configure support for the common language runtime (CLR).

▣ Options

Enable CLR integration

If checked, you can use CLR-based user-defined functions, procedures, triggers, and types from within the Database Engine.

Even if this option is disabled, you can run data definition language (DDL) statements to create, alter, or drop user assemblies and CLR modules, but you cannot execute the CLR code.

For more information about CLR integration, see **Building Database Objects with Common Language Runtime (CLR) Integration** in SQL Server Books Online.

For more information about configuring server options, see **sp_configure (Transact-SQL)** in SQL Server Books Online.

▣ **See Also**

Concepts

[Surface Area Configuration for Features](#)

[Surface Area Configuration for Services and Connections](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Surface Area Configuration for Features (DAC) - Database Engine



[See Also](#)



[Configuring a SQL Server Installation](#) > [SQL Server Surface Area Configuration](#) > [Surface Area Configuration for Features](#) >

Updated: 5 December 2005

The dedicated administrator connection (DAC) allows an administrator to connect to a server to execute diagnostic functions or Transact-SQL statements. The DAC can be used when the Database Engine will not respond to regular connections. By default, this feature is only enabled for a local connection. Enable this option to use the DAC from a remote computer. On cluster configurations, you must enable this option to connect with the DAC.

▣ Options

Enable remote DAC

If checked, the DAC is enabled from remote computers.

For more information about the DAC, see **Using a Dedicated Administrator Connection** in SQL Server Books Online.

For more information about configuring server options, see **sp_configure (Transact-SQL)** in SQL Server Books Online.

▣ Change History

Release	History
5 December 2005	New content: <ul style="list-style-type: none">• Added clarification about using this option with cluster configurations.

▣ **See Also**

Concepts

[Surface Area Configuration for Features](#)

[Surface Area Configuration for Services and Connections](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Surface Area Configuration for Features (Database Mail) - Database Engine

 [See Also](#)



[Configuring a SQL Server Installation](#) > [SQL Server Surface Area Configuration](#) > [Surface Area Configuration for Features](#) >

Database Mail is a new component for sending e-mail messages from the Database Engine using SMTP. Enable Database Mail stored procedures only if you plan to configure and use Database Mail.

Disabling Database Mail prevents the Database Mail program from starting, but if Database Mail is running when the option is changed, Database Mail continues to run and process mail (without query or attachments) until it completes sending mail and is idle for the time configured using the **DatabaseMailExeMinimumLifeTime** option.

▣ Options

Enable Database Mail stored procedures

If checked, the extended stored procedures that support Database Mail are enabled. To use Database Mail, you must configure it. For more information, see **Database Mail** in Microsoft SQL Server Books Online.

For more information about configuring SQL Server options, see **sp_configure (Transact-SQL)** in SQL Server Books Online.

▣ **See Also**

Concepts

[Surface Area Configuration for Features](#)

[Surface Area Configuration for Services and Connections](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Surface Area Configuration for Features (Native XML Web Services) - Database Engine

 [See
Also](#)



[Configuring a SQL Server Installation](#) > [SQL Server Surface Area Configuration](#) > [Surface Area Configuration for Features](#) >

Native XML Web Services provide database access over HTTP using SOAP messages. To use this capability you must create and start HTTP endpoints. For more information, see **CREATE ENDPOINT (Transact-SQL)** and **Using Native XML Web Services in SQL Server 2005** in Microsoft SQL Server 2005 Books Online.

 **Note:**

HTTP SOAP endpoint support started on the Microsoft Windows XP SP2 and Microsoft Windows Server 2003 operating systems.

▣ Options

Endpoint Name

The name of the HTTP endpoint.

State

Displays the current state of the endpoint. The current state can be Started, Stopped, or Disabled.

To change the state, select the state from the State box and click **Apply**.

- If the state is **Started**, you can change the state to **Stopped**. Stopping the endpoint both stops and disables the endpoint.
- If the state is **Stopped** or **Disabled**, you can change the state to **Started**. Starting the endpoint both enables and starts the endpoint.

If you want to only disable the endpoint, use ALTER ENDPOINT (Transact-SQL).

▣ **See Also**

Concepts

[Surface Area Configuration for Features](#)

[Surface Area Configuration for Services and Connections](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Surface Area Configuration for Features (OLE Automation) - Database Engine

 [See Also](#)



[Configuring a SQL Server Installation](#) > [SQL Server Surface Area Configuration](#) > [Surface Area Configuration for Features](#) >

The OLE automation extended stored procedures allow Transact-SQL batches, stored procedures, and triggers to reference custom OLE Automation objects. Use this dialog to control whether any of the following OLE automation procedures below can be executed.

- **sp_OACreate**
- **sp_OADestroy**
- **sp_OAGetErrorInfo**
- **sp_OAGetProperty**
- **sp_OAMethod**
- **sp_OASetProperty**
- **sp_OAStop**

In new installations of Microsoft SQL Server, these stored procedures are disabled by default.

▣ Options

Enable OLE Automation

If checked, the OLE Automation stored procedures are enabled.

For more information about OLE Automation, see **OLE Automation Objects in Transact-SQL** in SQL Server Books Online.

For more information about configuring SQL Server options, see **sp_configure (Transact-SQL)** in SQL Server Books Online.

▣ **See Also**

Concepts

[Surface Area Configuration for Features](#)

[Surface Area Configuration for Services and Connections](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Surface Area Configuration for Features (Service Broker) - Database Engine

 [See Also](#)



[Configuring a SQL Server Installation](#) > [SQL Server Surface Area Configuration](#) > [Surface Area Configuration for Features](#) >

Microsoft SQL Server uses a Service Broker endpoint for Service Broker communication outside the instance of SQL Server.

An endpoint is a SQL Server object that represents the capability for SQL Server to communicate over the network. Each endpoint supports a specific type of communication. For example, an HTTP endpoint lets SQL Server process specific SOAP requests. A Service Broker endpoint configures SQL Server to send and receive Service Broker messages over the network.

Service Broker endpoints can be started, stopped, or disabled. You can use this dialog to start or stop and disable the Service Broker endpoint for an instance.

▣ Options

Service Broker endpoint

Displays the name of the Service Broker endpoint.

State

Displays the current state of the Service Broker endpoint.

To change the state, select the state from the State box and then click **Apply**.

- If the state is **Started**, you can change the state to **Stopped**. Stopping the endpoint both stops and disables the endpoint.
- If the state is **Stopped** or **Disabled**, you can change the state to **Started**. Starting the endpoint both enables and starts the endpoint.

If you want to only disable the endpoint, use **ALTER ENDPOINT (Transact-SQL)**.

▣ **See Also**

Concepts

[Surface Area Configuration for Features](#)

[Surface Area Configuration for Services and Connections](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Surface Area Configuration for Features (SQL Mail) - Database Engine

 [See Also](#)



[Configuring a SQL Server Installation](#) > [SQL Server Surface Area Configuration](#) > [Surface Area Configuration for Features](#) >

SQL Mail, which is deprecated in favor of Database Mail, is disabled by default. Using this dialog, you can enable SQL Mail for backward compatibility.

▣ Options

Enable SQL Mail stored procedures

If checked, the extended stored procedures that support SQL Mail are enabled.

For more information about SQL Mail, see **SQL Mail** in Microsoft SQL Server Books Online.

For more information about configuring server options, see **sp_configure (Transact-SQL)** in SQL Server Books Online.

▣ **See Also**

Concepts

[Surface Area Configuration for Features](#)

[Surface Area Configuration for Services and Connections](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Surface Area Configuration for Features (Web Assistant) - Database Engine

 [See Also](#)



[Configuring a SQL Server Installation](#) > [SQL Server Surface Area Configuration](#) > [Surface Area Configuration for Features](#) >

The Web Assistant stored procedures generate HTML files from Microsoft SQL Server data. In Microsoft SQL Server 2005, these stored procedures have been deprecated. Use this dialog to enable the procedures used by the Web Assistant.

Reporting Services provides more advanced options for producing Web-based reports from SQL Server data. For more information, see **SQL Server Reporting Services** in SQL Server Books Online.

▣ Options

Enable Web Assistant

If checked, Web Assistant stored procedures are enabled.

For more information about configuring server options, see **sp_configure (Transact-SQL)** in SQL Server Books Online.

▣ **See Also**

Concepts

[Surface Area Configuration for Features](#)

[Surface Area Configuration for Services and Connections](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Surface Area Configuration for Features (xp_cmdshell) - Database Engine

 [_See
Also](#)



[Configuring a SQL Server Installation](#) > [SQL Server Surface Area Configuration](#) > [Surface Area Configuration for Features](#) >

The **xp_cmdshell** extended stored procedure executes a command string as an operating-system command shell and returns any output as rows of text. Use the **xp_cmdshell** dialog in Surface Area Configuration to configure **xp_cmdshell** support.

▣ Options

Enable `xp_cmdshell`

If checked, the `xp_cmdshell` extended stored procedure is enabled. If unchecked, the `xp_cmdshell` extended stored procedure cannot be used.

For more information about `xp_cmdshell`, see `xp_cmdshell (Transact-SQL)` in Microsoft SQL Server Books Online.

For more information about configuring server options, see **Setting Server Configuration Options** in SQL Server Books Online.

▣ **See Also**

Concepts

[SQL Server Surface Area Configuration](#)

[Surface Area Configuration for Features](#)

[Surface Area Configuration for Services and Connections](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Surface Area Configuration for Features (Ad Hoc Data Mining Queries) - Analysis Services

 [See Also](#)



[Configuring a SQL Server Installation](#) > [SQL Server Surface Area Configuration](#) > [Surface Area Configuration for Features](#) >

The Data Mining Extensions (DMX) OPENROWSET function supports ad hoc queries using external providers. If your applications and scripts do not use this function, disable ad hoc data mining queries.

If you enable ad hoc data-mining queries, you can enable or disable the passing of OLEDB provider name and connection string when using OPENROWSET.

▣ Options

Enable Ad hoc data mining queries

If checked, the DMX OPENROWSET function is enabled.

For more information, see **OPENROWSET (DMX)** in Microsoft SQL Server Books Online.

For more information about configuring server options, see **sp_configure (Transact-SQL)** in SQL Server Books Online.

▣ **See Also**

Concepts

[Surface Area Configuration for Features](#)

[Surface Area Configuration for Services and Connections](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Surface Area Configuration for Features (Anonymous Connections) - Analysis Services

 [See Also](#)



[Configuring a SQL Server Installation](#) > [SQL Server Surface Area Configuration](#) > [Surface Area Configuration for Features](#) >

Anonymous connections allow unauthenticated users to establish connections with Microsoft SQL Server 2005 Analysis Services. Unless your applications require unauthenticated users to connect to the instance, disable anonymous connections.

By default, Analysis Services disallows unauthenticated users from establishing a connection to the instance. Enabling anonymous connections to the instance is strongly discouraged, but it can be enabled by using SQL Server Management Studio to change the value of the **RequireClientAuthentication** server configuration property to **0** or by using this dialog.

▣ Options

Enable anonymous connections

If checked, unauthenticated users can connect to Analysis Services.

For more information about securing connections to an Analysis Services instance, see **Securing Client Communication with an Analysis Services Instance** in SQL Server Books Online.

▣ **See Also**

Concepts

[Surface Area Configuration for Features](#)

[Surface Area Configuration for Services and Connections](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Surface Area Configuration for Features (Linked Objects) - Analysis Services

 [See Also](#)



[Configuring a SQL Server Installation](#) > [SQL Server Surface Area Configuration](#) > [Surface Area Configuration for Features](#) >

By default, Microsoft SQL Server 2005 Analysis Services disallows the creation of linked objects, such as linked dimensions and linked measure groups. In order to enable the linking of these objects both instances need to enable this feature. This is controlled by using SQL Server Management Studio to change the values of the **Link\ToOtherInstanceEnabled** and **Link\FromOtherInstanceEnabled** server configuration properties, or by using this dialog.

▣ Options

Enable links to other instances

If checked, this instance of Analysis Services can link to objects on other Analysis Services instances.

For more information about linked objects, see **Linked Dimensions (SSAS)** and **Linked Measure Groups (SSAS)** in SQL Server Books Online.

Enable links from other instances

If checked, other instances of Analysis Services can link to objects on this instance.

For more information about linked objects, see **Linked Dimensions (SSAS)** and **Linked Measure Groups (SSAS)** in SQL Server Books Online.

▣ **See Also**

Concepts

[Surface Area Configuration for Features](#)

[Surface Area Configuration for Services and Connections](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Surface Area Configuration for Features (User-Defined Functions) - Analysis Services

 [See Also](#)



[Configuring a SQL Server Installation](#) > [SQL Server Surface Area Configuration](#) > [Surface Area Configuration for Features](#) >

Analysis Services allows you to load assemblies containing user-defined functions. These functions can be based on the common language runtime (CLR) or can be Component Object Model (COM) objects. Objects developed using the CLR can be secured using the CLR security model. COM objects do not support this security model. Unless your applications require user-defined COM objects, disable the loading of these objects.

▣ Options

Enable loading of user-defined COM functions

If checked, Analysis Services will load user-defined COM functions.

For more information about configuring server options, see **sp_configure (Transact-SQL)** in Microsoft SQL Server Books Online.

▣ **See Also**

Concepts

[Surface Area Configuration for Features](#)

[Surface Area Configuration for Services and Connections](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Surface Area Configuration for Features (Scheduled Events and Report Delivery) - Reporting Services

 [See Also](#)



[Configuring a SQL Server Installation](#) > [SQL Server Surface Area Configuration](#) > [Surface Area Configuration for Features](#) >

In Reporting Services, you can schedule report delivery, report snapshots, and report cache expiration. If you do not require subscription functionality, or the ability to schedule reports to run on specific dates and times, you can disable scheduling and delivery processing. Note that if you disable this feature, reports can only be run on demand, and e-mail and shared folder delivery are not available.

▣ Options

Enable scheduled events and report delivery

If checked, scheduled events and report delivery are enabled.

For more information, see **Scheduling and Delivery Processor** in Microsoft SQL Server Books Online.

▣ **See Also**

Concepts

[Surface Area Configuration for Features](#)

[Surface Area Configuration for Services and Connections](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Surface Area Configuration for Features (Web Service and HTTP Access) - Reporting Services

 [See Also](#)



[Configuring a SQL Server Installation](#) > [SQL Server Surface Area Configuration](#) > [Surface Area Configuration for Features](#) >

You can configure the Report Server web service to ignore all Simple Object Access Protocol (SOAP) requests and URL access requests. Disable HTTP and Web service requests if you do not have client applications that use the Web service, and if you do not require Report Manager, Report Builder, or Microsoft SQL Server Management Studio for this Reporting Services installation.

▣ Options

Enable Web service and URL access

If checked, Reporting Services allows connections using SOAP and URL access.

For more information, see **Report Server Web Service and URL Access** in SQL Server Books Online.

▣ **See Also**

Concepts

[Surface Area Configuration for Features](#)

[Surface Area Configuration for Services and Connections](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Surface Area Configuration for Features (Windows Integrated Security) - Reporting Services

 [See Also](#)



[Configuring a SQL Server Installation](#) > [SQL Server Surface Area Configuration](#) > [Surface Area Configuration for Features](#) >

In Reporting Services, report data sources can use Windows integrated security to connect to external data sources. Using Windows integrated security, however, may allow access to data under a user's identity without the user's knowledge. To ensure that all report data source connections use an explicitly specified set of credentials, disable Windows integrated security for report data sources. For more information about why integrated security might put sensitive data at risk, see **Integrated Security and Elevated Permissions**.

When you disable Windows integrated security, you reduce the options for configuring report data source credentials. The reduced set of choices apply to defining data sources in Report Manager, Management Studio, and any third-party tool that you use to define data source connections for a report. After you disable Windows integrated security, you can only obtain credentials in the following ways:

- Prompted credentials. Users are prompted to type a user name and password to run the report. Credentials can be Windows domain credentials or database credentials.
- Stored credentials. The credentials used to run the report are stored securely in the report server database. The credentials can be Windows domain credentials or database credentials. Only one set of credentials for each data source can be stored. All users access the report data source using the same credentials.
- No credentials. This option can be used when the data source either does not require credentials or allows credentials to be passed on the connection string. Because credentials appear in plain text on the data source properties page, this option is not recommended. To use this option, you must configure a special-purpose account that is used for

unattended report execution. For more information, see [Configuring an Account for Unattended Report Processing](#).

 **Note:**

Disabling Windows integrated security does not remove the **Use Windows integrated security** option in Report Manager, Management Studio, and the report design tools. If a user chooses Windows integrated security on a report server that does not support it, an error will occur when the report is executed.

▣ Options

Enable Windows integrated security for report data source connections

Use Windows integrated security for report data sources.

For more information, see **Integrated Security and Elevated Permissions** and **Specifying Credential and Connection Information**.

▣ **See Also**

Concepts

[Surface Area Configuration for Features](#)

[Surface Area Configuration for Services and Connections](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Surface Area Configuration for Services and Connections



[See Also](#)



[Configuring a SQL Server Installation](#) > [SQL Server Surface Area Configuration](#) >

The Surface Area Configuration for Features tool provides a single interface for enabling or disabling Microsoft SQL Server 2005 services and network protocols used for remote connections. Disabling unused services and connection types helps to secure your SQL Server installations by reducing the SQL Server surface area.

The tables below list the default for each setting and provide links to F1 Help topics.

▣ Windows Services and Network Protocol Defaults

Service or protocol	Default setting	F1 Help topic
Database Engine service	Startup can be set to Automatic or Manual during installation.	Surface Area Configuration for Services and Connections (Service) - Database Engine
Database Engine remote connections	Remote connections are disabled in SQL Server 2005 Express, Evaluation, and Developer editions and are enabled in other editions.	Surface Area Configuration for Services and Connections (Remote Connections) - Database Engine
Analysis Services service	Startup can be set to Automatic or Manual during installation.	Surface Area Configuration for Services and Connections (Service) - Analysis Services
Analysis Services remote connections	Remote connections are disabled in SQL Server 2005 Express, Evaluation, and Developer editions and are enabled in other editions.	Surface Area Configuration for Services and Connections (Remote Connections) - Analysis Services
Reporting Services service	Startup can be set to Automatic or Manual during installation.	Surface Area Configuration for Services and Connections (Service) - Reporting Services
SQL Server Agent service	If installed, the startup type is set to Manual and the service is stopped.	Surface Area Configuration for Services and Connections (Service) - SQL Server Agent
Full-text Search service	If installed, startup is set to Manual and the service is stopped.	Surface Area Configuration for Services and Connections (Service) - Full-Text Search
Notification Services instance services	Services not installed during setup.	Surface Area Configuration for Services and Connections (Service) - Notification Services
Integration Services	If installed, the service is set to the Automatic startup mode.	Surface Area Configuration for Services and Connections (Service) - Integration

service		Services
SQL Server Browser service	<p>Startup is set to Automatic in the following conditions:</p> <ul style="list-style-type: none">• When there are named instances of the Database Engine or Analysis Services on the server• When installed on a cluster• When installed side-by-side with SQL Server 2000	Surface Area Configuration for Services and Connections (Service) - SQL Server Browser

▣ **See Also**

Concepts

[SQL Server Surface Area Configuration](#)

[Surface Area Configuration for Features](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Surface Area Configuration for Services and Connections (Service) - Database Engine

 [See Also](#)



[Configuring a SQL Server Installation](#) > [SQL Server Surface Area Configuration](#) > [Surface Area Configuration for Services and Connections](#) >

The Database Engine Windows service is installed only if an instance of the Database Engine is installed. Use this dialog to change service settings for the instance.

▣ Options

Service name

The name of the service.

Display name

The name that appears in the **Name** column in the Microsoft Windows Services panel.

Description

The service description, which also appears in the **Description** column of the Windows services panel.

Startup type

Use to configure how the service starts.

Automatic specifies that the service starts when the system starts.

Manual specifies that a user or dependent service can start the service, and that the service does not start when the system starts.

Disabled prevents the service from being started by the system, a user, or any dependent service.

Service status

Shows the status of the service.

Start button

Starts the service.

Stop button

Stops the service.

Pause button

Pauses the service.

Resume button

Resumes the paused service.

▣ **See Also**

Concepts

[Surface Area Configuration for Services and Connections](#)

[Surface Area Configuration for Features](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Surface Area Configuration for Services and Connections (Remote Connections) - Database Engine

 [See Also](#)



[Configuring a SQL Server Installation](#) > [SQL Server Surface Area Configuration](#) > [Surface Area Configuration for Services and Connections](#) >

By default, SQL Server 2005 Express, Evaluation, and Developer editions allow local client connections only. Enterprise, Standard, and Workgroup editions also listen for remote client connections over TCP/IP. Use the Remote Connections page to change the protocols on which SQL Server listens for incoming client connections. TCP/IP is preferred over named pipes because it requires fewer ports to be opened across the firewall.

This settings on the Remote Connections page affect incoming connections only, not outward server connections such as those used for distributed queries.

 **Note:**

You must restart the Database Engine instance's service to apply changes.

▣ Options

Local connections only

If selected, the Database Engine instance will only listen for shared memory connections on the local computer. Connections from remote computers will fail.

 **Note:**

The dedicated administrator connection port is configured separately. See [Surface Area Configuration for Features \(DAC\) - Database Engine](#).

Local and remote connections

If selected, the Database Engine instance will listen for connections from the local computer and from remote computers using the selected protocols.

Using TCP/IP protocol only

If selected, the Database Engine instance listens for remote connections using TCP/IP, but not named pipes.

Using Named Pipes protocol only

If selected, the Database Engine instance listens for remote connections using named pipes, but not TCP/IP.

Using both TCP/IP and Named Pipes protocols

If selected, the Database Engine instance listens for remote connections using TCP/IP and named pipes.

▣ **See Also**

Concepts

[Surface Area Configuration for Services and Connections](#)

[Surface Area Configuration for Features](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Surface Area Configuration for Services and Connections (Service) - Analysis Services

 [See Also](#)



[Configuring a SQL Server Installation](#) > [SQL Server Surface Area Configuration](#) > [Surface Area Configuration for Services and Connections](#) >

Analysis Services Windows service are installed with instances of Analysis Services. Use this dialog to change service settings.

▣ Options

Service name

The name of the service.

Display name

The name that appears in the **Name** column in the Microsoft Windows Services panel.

Description

The service description, which also appears in the **Description** column of the Windows services panel.

Startup type

Use to configure how the service starts.

Automatic specifies that the service starts when the system starts.

Manual specifies that a user or dependent service can start the service, and that the service does not start when the system starts.

Disabled prevents the service from being started by the system, a user, or any dependent service.

Service status

Shows the status of the service.

Start button

Starts the service.

Stop button

Stops the service.

Pause button

Pauses the service.

Resume button

Resumes the paused service.

▣ **See Also**

Concepts

[SQL Server Surface Area Configuration](#)

[Surface Area Configuration for Services and Connections](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Surface Area Configuration for Services and Connections (Remote Connections) - Analysis Services

 [See
Also](#)



[Configuring a SQL Server Installation](#) > [SQL Server Surface Area Configuration](#) > [Surface Area Configuration for Services and Connections](#) >

In Microsoft SQL Server 2005 Express Edition, Developer Edition, and Evaluation Edition, remote connections are disabled by default. In other editions, remote connections are enabled by default. Use this dialog to enable or disable remote connections.

Enabling remote connections for Analysis Services opens a TCP/IP port on the server. Enable remote connections only if you want to allow connections from remote computers.

 **Note:**

You must restart the Analysis Services service to apply protocol changes.

▣ Options

Local connections only

Select this option to disable remote connections.

If this option is selected, the Analysis Services instance listens for connections from the local computer only. Connections from remote computers fail.

Local and remote connections

Select this option to enable remote connections.

If this option is selected, the Analysis Services instance listens for connections from the local computer and from remote computers using TCP/IP.

▣ **See Also**

Concepts

[Surface Area Configuration for Services and Connections](#)

[Surface Area Configuration for Features](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Surface Area Configuration for Services and Connections (Service) - Reporting Services

 [See Also](#)



[Configuring a SQL Server Installation](#) > [SQL Server Surface Area Configuration](#) > [Surface Area Configuration for Services and Connections](#) >

The Reporting Services Windows service is installed only if Reporting Services is installed. Use this dialog to change service settings.

▣ Options

Service name

The name of the service.

Display name

The name that appears in the **Name** column in the Microsoft Windows Services panel.

Description

The service description, which also appears in the **Description** column of the Windows Services panel.

Startup type

Configure how the service starts.

Automatic specifies that the service starts when the system starts.

Manual specifies that a user or dependent service can start the service, and that the service does not start when the system starts.

Disabled prevents the service from being started by the system, a user, or any dependent service.

Service status

Show the status of the service.

Start button

Start the service.

Stop button

Stop the service.

Pause button

Pause the service.

Resume button

Resume the paused service.

▣ **See Also**

Concepts

[Surface Area Configuration for Services and Connections](#)

[Surface Area Configuration for Features](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Surface Area Configuration for Services and Connections (Service) - SQL Server Agent

 [See
Also](#)



[Configuring a SQL Server Installation](#) > [SQL Server Surface Area Configuration](#) > [Surface Area Configuration for Services and Connections](#) >

Microsoft SQL Server Agent Windows services are installed with instances of the Database Engine. By default, this service is disabled and stopped. If you want to use automated administration, you must enable this service. Use this dialog to change service settings. For more information about configuring the SQL Server Agent service account, see **Selecting an Account for SQL Server Agent Service** in Microsoft SQL Server 2005 Books Online.

▣ Options

Service name

The name of the service.

Display name

The name that appears in the **Name** column in the Windows Services panel.

Description

The service description, which also appears in the **Description** column of the Microsoft Windows services panel.

Startup type

Use to configure how the service starts.

Automatic specifies that the service starts when the system starts.

Manual specifies that a user or dependent service can start the service, and that the service does not start when the system starts.

Disabled prevents the service from being started by the system, a user, or any dependent service.

Service status

Shows the status of the service.

Start button

Starts the service.

Stop button

Stops the service.

Pause button

Pauses the service.

Resume button

Resumes the paused service.

▣ **See Also**

Concepts

[Surface Area Configuration for Services and Connections](#)

[Surface Area Configuration for Features](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Surface Area Configuration for Services and Connections (Service) - Full-Text Search

 [See Also](#)



[Configuring a SQL Server Installation](#) > [SQL Server Surface Area Configuration](#) > [Surface Area Configuration for Services and Connections](#) >

The Microsoft Full-Text Engine (MSFTE) for Microsoft SQL Server service is installed only if the Database Engine is installed. Use this dialog to change service settings.

▣ Options

Service name

The name of the service.

Display name

The name that appears in the **Name** column in the Microsoft Windows Services panel.

Description

The service description, which also appears in the **Description** column of the Windows services panel.

Startup type

Use to configure how the service starts.

Automatic specifies that the service starts when the system starts.

Manual specifies that a user or dependent service can start the service, and that the service does not start when the system starts.

Disabled prevents the service from being started by the system, a user, or any dependent service.

Service status

Shows the status of the service.

Start button

Starts the service.

Stop button

Stops the service.

Pause button

Pauses the service.

Resume button

Resumes the paused service.

▣ **See Also**

Concepts

[Surface Area Configuration for Services and Connections](#)

[Surface Area Configuration for Features](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Surface Area Configuration for Services and Connections (Service) - Notification Services

 [See
Also](#)



[Configuring a SQL Server Installation](#) > [SQL Server Surface Area Configuration](#) > [Surface Area Configuration for Services and Connections](#) >

You can install a Notification Services Windows service when you deploy an instance of Notification Services. The Microsoft Windows services are not installed when you install Notification Services. If you have installed one or more Notification Services Windows services on the server, use this dialog to change service settings.

▣ Options

Service name

The name of the service.

Display name

The name that appears in the **Name** column in the Windows Services panel.

Description

The service description, which also appears in the **Description** column of the Windows services panel.

Startup type

Use to configure how the service starts.

Automatic specifies that the service starts when the system starts.

Manual specifies that a user or dependent service can start the service, and that the service does not start when the system starts.

Disabled prevents the service from being started by the system, a user, or any dependent service.

Service status

Shows the status of the service.

Start button

Starts the service.

Stop button

Stops the service.

Pause button

Pauses the service.

Resume button

Resumes the paused service.

▣ **See Also**

Concepts

[Surface Area Configuration for Services and Connections](#)

[Surface Area Configuration for Features](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Surface Area Configuration for Services and Connections (Service) - Integration Services

 [See
Also](#)



[Configuring a SQL Server Installation](#) > [SQL Server Surface Area Configuration](#) > [Surface Area Configuration for Services and Connections](#) >

The Integration Services Windows service is installed only if Integration Services is installed. Use this dialog to change service settings.

▣ Options

Service name

The name of the service.

Display name

The name that appears in the **Name** column in the Microsoft Windows Services panel.

Description

The service description, which also appears in the **Description** column of the Windows services panel.

Startup type

Use to configure how the service starts.

Automatic specifies that the service starts when the system starts.

Manual specifies that a user or dependent service can start the service, and that the service does not start when the system starts.

Disabled prevents the service from being started by the system, a user, or any dependent service.

Service status

Shows the status of the service.

Start button

Starts the service.

Stop button

Stops the service.

Pause button

Pauses the service.

Resume button

Resumes the paused service.

▣ **See Also**

Concepts

[SQL Server Surface Area Configuration](#)

[Surface Area Configuration for Services and Connections](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Surface Area Configuration for Services and Connections (Service) - SQL Server Browser

 [See
Also](#)



[Configuring a SQL Server Installation](#) > [SQL Server Surface Area Configuration](#) > [Surface Area Configuration for Services and Connections](#) >

The SQL Server Browser service returns named pipe and TCP port information to client applications. Enabling this service is useful when a pipe name or port number of a named instance is unknown to client applications. If you disable this service, you may choose static ports for Database Engine named instances and then specify the port number in client connections. For Analysis Services named instances, it is best to enable the SQL Server Browser service.

▣ Options

Service name

The name of the service.

Display name

The name that appears in the **Name** column in the Windows Services panel.

Description

The service description, which also appears in the **Description** column of the Microsoft Windows services panel.

Startup type

Use to configure how the service starts.

Automatic specifies that the service starts when the system starts.

Manual specifies that a user or dependent service can start the service, and that the service does not start when the system starts.

Disabled prevents the service from being started by the system, a user, or any dependent service.

Service status

Shows the status of the service.

Start button

Starts the service.

Stop button

Stops the service.

Pause button

Pauses the service.

Resume button

Resumes the paused service.

▣ **See Also**

Concepts

[Surface Area Configuration for Services and Connections](#)

[Surface Area Configuration for Features](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

sac Utility



[See Also](#)



[Configuring a SQL Server Installation](#) > [SQL Server Surface Area Configuration](#) >

The **sac** utility imports and exports Microsoft SQL Server 2005 surface area settings. Using this utility, you can configure the surface area on one computer, and then apply the same settings to other computers.

The easiest way to use the **sac** utility is to use the SQL Server Surface Area Configuration graphical user interface to configure one computer, then use **sac** to export the settings of that computer to a file. You can then use **sac** to apply the settings for all SQL Server 2005 components to other SQL Server 2005 instances on the local computer or on remote computers.

This utility is located at x:\Program Files\Microsoft SQL Server\90\Shared.

▣ Syntax

```
sac {in | out} filename [-S computer_name]
    [-U SQL_login [-P SQL_password]]
    [-I instance_name ]
    [-DE] [-AS] [-RS] [-IS] [-NS] [-AG] [-BS] [-FT]
    [-F] [-N] [-T] [-O]
    [-H | -?]
```

▣ Parameters

in

Import the surface area settings from a file and configure the instance, specified by *instance_name*, using those settings.

out

Export the surface area configuration settings from an instance to the file specified by *filename*.

filename

The full path of the file used when importing or exporting the surface area settings.

-S *computer_name*

Specifies the name of a remote computer. If this argument is not specified, **sac** connects to the local computer.

-U *SQL_login*

Specifies the SQL Server Authentication login to use for the connection to the Database Engine. If not specified, **sac** uses Windows Authentication to connect to the Database Engine.

-P *SQL_password*

Specifies the password for *SQL_login*. If this argument is not specified, **sac** prompts for a password. If **-P** is specified at the end of the command without a value, **sac** uses a null password.

-I *instance_name*

Specifies the SQL Server instance to run **sac** against. If this option is not specified, **sac** connects to all SQL Server instances. For the default instance, the instance name is **MSSQLServer**.

-DE

Import or export Database Engine settings.

-AS

Import or export Analysis Services settings.

-RS

Import or export Reporting Services settings.

-IS

Import or export Integration Services settings.

-NS

Import or export Notification Services settings.

-AG

Import or export SQL Server Agent settings.

-BS

Import or export the SQL Server Browser service settings.

-FT

Import or export the Full-Text Search service settings.

-F

Import or export the state of the following features for the specified components:

Component	Features
	Ad hoc remote queries CLR integration

Database Engine	DAC Database Mail Native XML Web service OLE automation SQL Server Service Broker SQL Mail Web assistant xp_cmdshell
Analysis Services	Ad hoc data mining queries Anonymous connections Linked objects User-defined functions
Reporting Services	Scheduled events and report delivery Web service and HTTP access Windows integrated security

-N

Import or export the state of network protocols for remote connectivity for the specified components. Only the Database Engine and Analysis Services provide options for configuring remote connectivity.

-T

Import or export the state of the Windows services for the specified components.

-O

Specifies the name of the file that receives command-line output from **sac**.

-H | **-?**

Display command syntax. Other arguments are ignored.

▣ **Remarks**

The following rules apply to **sac**:

- Arguments are not case-sensitive.
- Precede command arguments with either a hyphen (-) or a forward slash (/). These symbols are equivalent.
- Include a space between the argument and its value.
- Enclose any value that contains special characters in quotation marks.

▣ Examples

The following examples show common usage scenarios for the **sac** utility:

A. Export All Default Instance Settings

The following example shows how to export all of the feature, connections, and service settings for the default SQL Server instance to the file server1.out. Because the example specifies a user name but no password, **sac** will prompt for a password.

 [Copy Code](#)

```
sac out server1.out -S server1 -U admin -I MSSQLSERVER
```

B. Import Feature Settings to Another Server

The following example imports only the feature settings from the file server1.out and applies them to the default SQL Server instance on server 2. Because this example does not provide a user name, **sac** uses Windows Authentication.

 [Copy Code](#)

```
sac in server1.out -S server2 -F
```

C. Export All Local Feature and Network Settings

The following example will export all feature and connection settings, but not service settings, for all SQL Server instances and components on the local server to the file server1.out.

 [Copy Code](#)

```
sac out server1.out -F -N
```

D. Import All Settings

The following example imports all settings available in the file server1.out to server2.

 [Copy Code](#)

```
sac in server1.out -S server2
```

E. Export All Database Engine Settings

The following example exports all settings for all Database Engine instances on the local server to the file server1.out.

 [Copy Code](#)

```
sac out server1.out -DE
```

F. Import Service State Settings

The following example imports only the service state settings from server1.out and applies the service settings to server2.

 [Copy Code](#)

```
sac in server1.out -S server2 -T
```

▣ Change History

Release	History
	Changed content: <ul style="list-style-type: none"><li data-bbox="500 499 1073 541">• Removed -AD switch from syntax.

▣ **See Also**

Concepts

[SQL Server Surface Area Configuration](#)

Other Resources

Command Prompt Utilities

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Reporting Services Configuration

 [See Also](#)



[Configuring a SQL Server Installation](#) >

Use the Reporting Services Configuration tool to configure a SQL Server 2005 Reporting Services installation. If you installed a report server using the files-only installation option, you must use this tool to configure the server so that it can be used. If you installed a report server in the default configuration installation option, you can use this tool to verify or modify the settings that were specified during setup. If you are upgrading from a previous version, you can use this tool to update the report server database to the new format.

Reporting Services Configuration can be used to configure a local or remote report server instance. You must have local system administrator permissions on the computer that hosts the report server you want to configure.

Note:

The Reporting Services configuration tool uses icons to indicate whether settings are configured. It is possible to configure settings that are not valid. You should always test a Reporting Services installation to verify it works as expected. Visual indicators for configuration status are not a substitute for tests that verify a successful deployment. For more information about how to verify a deployment, see [How to: Verify a Reporting Services Installation](#).

▣ In This Section

Term	Def
Configuring a Report Server for Internet Access	Describes different approaches for deploying reports on an Internet-facing Web server.
Configuring a Report Server Scale-Out Deployment	Explains how to configure multiple report server instances to use a shared report server database.
Configuring Reporting Services for SharePoint 3.0 Integration	Provides step-by-step instructions for configuring a report server to run within a larger deployment of a SharePoint product or technology.
Configuring Report Server Virtual Directories	Describes how to create the virtual directories used to access the report server and Report Manager.
Configuring Service Accounts and Passwords in Reporting Services	Provides recommendations and steps on how to modify service accounts and passwords.
Creating a Report Server Database	Describes how to create a report server database, required for storing server metadata and objects.
Configuring a Report Server Database Connection	Describes how to modify the connection string used by the report server to connect to the report server database.
Configuring a Report Server for E-Mail Delivery	Describes how to configure a report server to support e-mail report distribution.
Configuring an Account for Unattended Report Processing	Describes how to configure a user account to process reports in unattended mode.
Administering a Report Server Database	Describes administration tasks and provides information about database roles and permissions.
Administering the Report Server Web Service and Windows Service	Describes the purpose and function of each service, and how to make the services available.
Starting and Stopping the Report Server Windows Service	Explains how to start and stop the Report Server Windows service.

Managing Encryption Keys	Explains how to use and manage encryption keys that are used when storing sensitive data.
Reporting Services Configuration How-to Topics	Provides step-by-step instruction for common tasks.
Reporting Services Configuration F1 Help	Provides help topics for the pages in the Reporting Services Configuration tool.

▣ **In SQL Server 2005 Books Online**

The online help for the Reporting Services Configuration tool is a subset of the product documentation for SQL Server 2005 Reporting Services. Complete product documentation can be found in SQL Server 2005 Books Online.

When you use SQL Server Books Online, you have access to all of the topics provided in this online help system, plus additional content about deploying, administering, and securing a Reporting Services installation. For more information, see [How to: Access Books Online for SQL Server 2005](#).

▣ **See Also**

Concepts

[Installing SQL Server Reporting Services](#)

[Upgrading Reporting Services](#)

[Configuring a SQL Server Installation](#)

Other Resources

[Reporting Services Installation How-to Topics](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Configuring a Report Server for Internet Access



[See Also](#)



[Configuring a SQL Server Installation](#) > [Reporting Services Configuration](#) >

Updated: **14 April 2006**

Although SQL Server Reporting Services is not expressly designed for Internet report deployment scenarios, you can successfully place Reporting Services on an Internet-facing Web server to disseminate general information to the public at large or private corporate data to authorized and authenticated users. Depending on application and user requirements, you can choose any of the following approaches:

- Embedded reports hosted in a ReportViewer Web server control in a custom Internet application that you create and deploy.
- Report server deployment on an extranet, using Windows Authentication.
- Report server deployment on an extranet, using custom authentication.
- Report server configured to run as a back-end server in a larger deployment of a SharePoint product or technology, where you configure the front-end server for Internet deployment.

Note:

Although Report Manager was not designed as an Internet application, you can install just Report Manager on an Internet-facing Web server, and then install the report server and report server database behind the firewall. To install Report Manager on an Internet-facing Web server, you must select both Report Server and Report Manager as feature components. After setup is finished, you can use SQL Server Surface Area Configuration tool to disable the report server instance that installed with Report Manager. You can then configure Report Manager to use the second report server instance that is behind the firewall. You need to modify the **<ReportServerVirtualDirectory>** and **<ReportServerUrl>** settings in the RSWebApplication.config file to point Report Manager to the report server

instance.

▣ **Using ReportViewer Web Server Controls to Host Reports in an Internet Application**

To deploy reports in an Internet application, you can embed the ReportViewer Web server control in a custom Internet-ready application that you create and deploy. The ReportViewer Web server control is included in Visual Studio 2005 and can be distributed freely with your application. You can configure the control to display reports that run on a report server. The connection between your application and the report server is handled by the control, through the Web service programming interface. All authentication and authorization is handled by your application, which connects to the report server through a single, user-trusted connection.

📌 **Note:**

ReportViewer controls can be used independently of Reporting Services to host client report definition (.rdlc) files that you provide in your application. For more information, see **Comparing Reporting Services and ReportViewer Controls** in SQL Server Books Online.

▣ **Deploying Report Server and Report Manager in an Extranet**

To deploy a report server in an extranet scenario that supports connections from predefined Microsoft Active Directory accounts, you can use the default Windows Authentication security extension. Plan on configuring the server for Secure Sockets Layer (SSL) connections and Basic authentication in Internet Information Services (IIS).

To deploy a report server in an extranet scenario that supports connections from a single sign-on technology or a forms-based authentication model that stores user identity information in a database, you must create a custom authentication extension to replace the default Windows Authentication security extension. Forms-based authentication is typically used when you are opening up a report server to the public but only want authenticated users to view content. By default, SQL Server Reporting Services does not provide a forms-based authentication module. For more information about how to create one, see **Implementing a Security Extension** in SQL Server Books Online.

Caution:

Configuring a report server for Anonymous access is not recommended or encouraged unless you are also using custom authentication. If you do not use custom authentication, and you enable Anonymous access on a report server anyway, you will not be able to vary role assignments in a meaningful way. All users will access the report server under the Anonymous user account, and no one will have permission to administer the report server through Report Manager or Management Studio. Role assignments that you subsequently define will not vary for individual users; all users will have the same access to the same content and reports that require user-dependent data will not function.

Use the following guidelines to deploy a report server and Report Manager on the same computer:

1. Install Reporting Services on an Internet-facing Web server using the files-only installation mode. In SQL Server Installation Wizard, this is the **Install but do not configure** option.

2. Secure the connection between the Web server and the SQL Server Database Engine instance you plan to use to host the report server database. You can use IPsec to protect the connection.
3. Run the Reporting Services Configuration tool after Setup is complete to configure the report server:
 1. Specify the service accounts and create the report server database.
 2. If the SQL Server instance is in a different domain and Kerberos 5.0 is not enabled, use SQL Server authentication for the report server database connection.
4. Configure the report server and Report Manager virtual directories, choosing a Web site that you have configured for Internet access.

 **Note:**

In previous versions, you had to modify configuration settings when deploying a report server instance on an Internet-facing Web server. This step is no longer required as long as Report Manager and report server are installed together.

Configuring Proxy Settings in Web.config Files

In SQL Server 2005, Reporting Services includes a Web.config setting that allows Report Manager to bypass the proxy server when sending requests to a local report server that is installed on the same computer.

The Web.config setting is the System.NET **defaultProxy** network setting. By default, **defaultProxy** is disabled in the Web.config file for Report Manager. This is the recommended configuration when Report Manager and the report server are deployed together on the same computer.

If you upgraded from SQL Server 2000 Reporting Services, the Report Manager Web.config file does not include the **defaultProxy** configuration setting. You can add and set the **defaultProxy** setting to bypass the proxy server for installations where Report Manager and report server are running on the same computer. Copy the following configuration settings into the Report Manager Web.config file:

 [Copy Code](#)

```
<configuration>  
...
```

```
<system.net>  
  <defaultProxy enabled="false" />  
</system.net>  
</configuration>
```

For more information about these settings, see "Configuring Internet Applications" and "defaultProxy Element (Network Settings)" in the Microsoft .NET Framework Developer's Guide.

Verifying Internet Deployment

To verify that your report server connection is accessible, you should be able to view the report server folder namespace over an Internet connection by typing **http: (or https://<your-web-server-fully-qualified-domain-name>/reportserver**, where **/reportserver** is the default name of the report server virtual directory.

Note:

Deploying Reporting Services on an Internet-facing Web server requires careful evaluation on your part. Network and security software, network topology, and domain configuration can introduce variables that make it difficult to prescribe an exact set of steps for report server deployment. When evaluating a report server for Internet access, be sure that your test scenarios include all of the features you intend to use and that you always test using secure connections.

▣ **Running a Report Server in an Internet Deployment of a SharePoint Product or Technology**

If you have an instance of a SharePoint product or technology that is configured for connections over the Internet, you can use that instance as a front-end server for reports that run on a Reporting Services report server. The authentication provider and permissions you configure for a SharePoint farm are used to access report server content and operations. By using the SharePoint security features, you eliminate the need to use a custom authentication extension on the report server that you must create and deploy yourself. It also eliminates the need to set up the report server site for Internet connections. For more information about server integration, see [Reporting Services and SharePoint Technology Integration](#).

▣ **See Also**

Concepts

[Configuring Report Server Virtual Directories](#)

Other Resources

RSWebApplication Configuration File

Deploying Reporting Services

Integrated Security and Elevated Permissions

Configuring Report Server Security

[| © 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Configuring a Report Server Scale-Out Deployment



[See Also](#)



[Configuring a SQL Server Installation](#) > [Reporting Services Configuration](#) >

A scale-out deployment refers to an installation configuration that has multiple report server instances sharing a single report server database. This deployment model is used when you want to run Reporting Services in a server cluster. Configuring the report server nodes to use a shared database is a prerequisite to running the report servers in a cluster.

Requirements for a scale-out deployment include the following:

- The Reporting Services edition must be Enterprise, Developer, or Evaluation. Standard edition does not support a scale-out deployment. You can create a scale-out deployment using a combination of editions as long as the edition supports the scale-out feature.
- All of the report servers must be in the same domain or in a trusted domain.
- All of the report servers must be the same version (SQL Server 2005 or Service Pack 1).

A report server scale-out can include report server instances that are all configured to run in native mode, or instances that are all configured to run in SharePoint integrated mode. You cannot create a scale-out deployment that includes report server instances that run in different modes. For more information about server modes, see [Deployment Modes for Reporting Services](#).

Important:

To run a report server scale-out deployment on a server cluster, you must also have software and tools that you can use to create and manage a server cluster. Reporting Services does not provide this functionality. For more information about running Reporting Services in a server cluster, see **Planning a Reporting Services Deployment** in SQL Server Books Online.

▣ Deployment Steps for Report Server Scale-Out

Deployment steps include installation, configuration through the Reporting Services Configuration tool, and manual editing of the report server configuration files. If you are using Microsoft Application Center, you must exclude configuration files from replication operations.

Pre-installation Steps for an Application Center Deployment

If you are using Application Center to manage server clusters, you must define filters to exclude the report server configuration file (RSReportServer.config) from replication operations. Use the files-only installation option to install the report server instances you want to use. This allows you to set the filter prior to configuring the report server instances that will be included in the cluster.

1. To set the filter, open the Application Center 2000 console, right-click **Synchronizations**, and select **Properties**.
2. Define a filter that excludes RSReportServer.config. The default path is c:\Program Files\Microsoft SQL Server\MSSQLn\Reporting Services\ReportServer\RSReportServer.config.

Creating and Configuring Report Server Instances

You can install Reporting Services on nodes that are already part of a server cluster. Follow these steps to install and enable Reporting Services functionality:

1. Install and configure a report server scale-out deployment. For more information, see [How to: Configure a Report Server Scale-Out Deployment \(Reporting Services Configuration\)](#).
2. Run the SQL Server Surface Area Configuration tool to verify that Reporting Services is enabled. The Report Server Windows service must be enabled and running. The service is both enabled and running if you chose the **autostart** option in Setup. For more information, see [SQL Server Surface Area Configuration](#).
3. Configure view state validation by editing the Web.config file for Report Manager for every report server instance in the deployment. This step occurs after all of the nodes in the scale-out are installed and initialized, and

before you edit the Reporting Services configuration files. Once incoming requests are load-balanced across the nodes, view state validation must be configured correctly. Otherwise, you will see many exceptions due to failed state validation. For more information, see the [Configuring View State Validation](#) section in this topic.

4. If you are using Secure Sockets Layer (SSL), verify the SSL certificates are installed in the IIS application pools you are using, and that the virtual directories are configured to use SSL. Use the Virtual Directory pages in the Reporting Services Configuration tool to specify SSL connection for both Report Manager and the report server. Before continuing to the next step, verify that the report servers are configured correctly by opening Report Manager through an https:// connection. For more information about using SSL with a report server, see **Configuring a Report Server for Secure Sockets Layer (SSL) Connections**.
5. Back up all configuration files in case you need to roll back later. In subsequent steps, you will modify the configuration files. If you make a mistake, you can roll back to the copies that you create in this step. Configuration files that you must back up include RSReportServer.config, RSWebApplication.config, and the Web.config files for both the report server and Report Manager.
6. If you are using a virtual server as a single point of access to a Network Load Balanced (NLB) cluster, perform the following additional steps:
 1. Edit the Reporting Services configuration files to specify the virtual server name. For more information, see the [Specifying the Virtual Server Name in Reporting Services Configuration Files](#) section in this topic.
 2. In IIS, configure redirection to the URL of the virtual server. To configure redirection, set options on the Home Directory tab on Web Site Properties.
 3. Verify that you can access the scale-out deployment through the virtual server name (for example, https://MyReportPortal/reportserver and https://MyReportPortal/reports). You can check which node actually processes reports by looking at the report server log files or by checking the RS execution log (the execution log table contains a column called InstanceName which will show which instance processed a particular request). For more information, see **Reporting Services Log Files** in SQL Server Books Online.

Configuring View State Validation

To complete scale-out deployment, you must edit configuration files to enable view state for interactive HTML reports that are viewed in Report Manager.

In Microsoft .NET Framework 2.0, view state validation is enabled by default and uses ASP.NET process identity information to perform validation. Validation by process identity does not work in a cluster or scale-out scenario where the process identity varies for each node. As an alternative, you can generate a validation key to support view state validation and manually configure each report server node to use it.

The following steps include references to generating keys for the <machineKey> element. There are a variety of ways to generate key values, and the approaches for doing so are described in the .NET Framework documentation. For more information about configuring view state validation for a network of Web services, see the .NET Framework reference documentation for the <machineKey> element.

1. Generate a validation key. You can use the autogenerate functionality provided by the .NET Framework or you can create the key some other way. Do not use the AutoGenerate option when setting the **validationKey** attribute.
2. Generate a decryption key. For the decryption key, you can create an explicit value or set decryptionKey to **AutoGenerate**.
3. Open the Web.config file for Report Manager and set the <machineKey> element. You must specify the validation key, decryption key, and the type of encryption used for validation of data. The following example illustrates the entries you must provide (these example values are not valid; avoid copying these entries into your configuration files):

```
<machineKey  
validationKey="53ab8bv3ebc93bb826a7b3v37b903a9d5d4a0"  
decryptionKey="26165b97a9d5d4a0" validation="SHA1"/>
```
4. Repeat these steps for each report server in the scale-out deployment. Verify that all Web.Config files in the \Reporting Services\Report Manager folders contain identical <machineKey> elements in the <system.web> section.

Specifying the Virtual Server Name in Reporting Services Configuration files

If you configure the report server scale-out deployment to run on an NLB cluster, you must manually update report server URL settings in the configuration files to use the virtual server name. For more information about the Reporting Services configuration files, see **Reporting Services Configuration Files** in SQL Server Books Online.

1. Use a text editor to open the RSWebApplication.config file.
2. Set the **<ReportServerUrl>** to the virtual server name and remove the entry for **<ReportServerVirtualDirectory>**. This step ensures that all requests coming through Report Manager are load-balanced to the report servers that are running in the scale-out deployment. The following example illustrates the syntax you should use; it specifies the virtual server name and report server virtual directory:



```
<ReportServerUrl>https://MyReportPortal/reportserver</ReportServerUrl>  
<ReportServerVirtualDirectory></ReportServerVirtualDirectory>
```

3. Open the RSReportServer.config file.
4. Set the **<UrlRoot>** to the virtual server address. This step ensures that all hyperlinks in reports point back to the scale-out deployment and are load-balanced accordingly. This setting is also used to complete report delivery. The following example illustrates the syntax you should use:



```
<ReportServerUrl>https://MyReportPortal/reportserver</ReportServerUrl>
```

5. Repeat these steps for each report server in the scale out deployment.

If you start the Reporting Services Configuration tool after making these changes, the tool might change the **<ReportServerUrl>** settings to the default value. Always keep a backup copy of the configuration files in case you need to replace them with the version that contains the settings you want to use.

▣ **See Also**

Concepts

[Installing SQL Server Reporting Services](#)

Other Resources

[Reporting Services Component Overview](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Deployment Modes for Reporting Services



[See Also](#)



[Configuring a SQL Server Installation](#) > [Reporting Services Configuration](#) >

New: **12 December 2006**

SQL Server 2005 Reporting Services supports two modes of deployment for report server instances:

- Native mode, including native mode with SharePoint Web Parts.
- SharePoint integrated mode.

 **Note:**

SQL Server 2005 Service Pack 2 (SP2) introduces the concept of a mode of operation for a Reporting Services report server. Prior to this release, all report servers ran in what is now defined as native mode.

▣ **Native Mode**

In native mode, a report server is a stand-alone application server that provides all viewing, management, processing, and delivery of reports and report models. This is the default mode for report server instances.

All report server instances are installed in native mode or in no mode at all. If you install a report server in the default configuration, the report server is fully operational in native mode. If you install a report server by using the files-only option (this is the **Install but do not configure the server** option in the Report Server Installation Options page), the report server is not configured; the database flag that you subsequently set for that instance determines which server mode it runs under.

Native Mode with SharePoint Web Parts

Reporting Services includes two Web Parts that you can install and register on an instance of Windows SharePoint Services 2.0 or 3.0. From a SharePoint site, you can use the Web Parts to find and view reports that are stored and processed on a report server that runs in native mode. These Web parts were introduced in earlier releases of Reporting Services. For more information about how the native mode Web Parts compare with SharePoint integrated mode features, see [Comparing Levels of Integration Across Versions of Reporting Services and SharePoint Technologies](#).

▣ SharePoint Integrated Mode

After you install Reporting Services, you can configure a report server to run in SharePoint integrated mode. In this mode, a report server becomes part of a SharePoint Web application deployment. Users of Windows SharePoint Services 3.0 or Office SharePoint Server 2007 can store reports in SharePoint libraries and access them from the same SharePoint sites they use to access other business documents.

A report server is configured for SharePoint integrated mode if it has the following characteristics:

- The report server is running SQL Server 2005 SP 2, the target instance of the SharePoint technology is version 3.0, and both servers meet the requirements for running in this mode. For more information, see [Requirements for Running Reporting Services in SharePoint Integrated Mode](#).
- The report server database is created with an option that flags it for integrated operations. When you connect a report server instance to a report server database that runs in SharePoint integrated mode, the report server property **SharePointIntegrated** is set to **True** automatically. In this mode, the report server switches to the custom security extension and SOAP endpoint that are used when a report server is set for **SharePointIntegrated** mode.
- When the report server database is created with **SharePointIntegrated** mode set, the site hierarchy in Windows SharePoint Services becomes the content hierarchy for the report server. The report server accepts only fully qualified SharePoint URLs as paths. No relative paths are allowed. File extensions for each item are required. For more information about the hierarchy and URL requirements, see **Deploying Reports, Models, and Shared Data Sources to a SharePoint Site**.
- Both the report server and the SharePoint Web application are configured for connections to and from each other, and the report server is granted permissions to the SharePoint databases.

▣ **Switching Modes**

Report server modes are mutually exclusive. You can switch modes by repurposing an existing report server deployment, but there is no supported approach for migrating content between database types.

To switch modes, modify the report server connection to the report server database so that it points to a database that supports a specific mode of operation. For more information about how to configure an existing report server instance to run in SharePoint integrated mode, see [Configuring Reporting Services for SharePoint 3.0 Integration](#).

▣ **Deploying a SharePoint Integrated Report Server in a Multi-Lingual Environment**

Windows SharePoint Services and Office Server products are available in many more languages than SQL Server 2005. Currently, SQL Server 2005 is localized into these languages: Simplified Chinese, Traditional Chinese, French, German, Italian, Japanese, Korean, Russian, and Spanish.

When you configure a report server to run within a deployment of a SharePoint product or technology, you might see a combination of languages. The user interface, documentation, and messages will appear in the following languages:

- All application pages, tools, errors, warnings, and messages that originate from Reporting Services will appear in the language used by the Reporting Services instance in one of the SQL Server languages.
- Application pages that you open on a SharePoint site, the Report Viewer Web Part, and Report Builder in one of the 22 supported languages for the Reporting Services Add-in. To view the list of supported languages, go to [SQL Server Downloads](#) and find the download page for the Reporting Services Add-in.
- SharePoint sites, SharePoint Central Administration, online help, and messages are available in the 39 languages supported by Office Server products.

If the language of your SharePoint product or technology differs from the report server language, Reporting Services will try to use a language from the same language family that provides the closest match. If a close substitute is not available, the report server uses English.

For more information about language support in Reporting Services, see [International Considerations for Reporting Services](#) on msdn.microsoft.com or **International Considerations for Reporting Services** in SQL Server Books Online.

▣ **See Also**

Tasks

[How to: Create a Report Server Database for SharePoint Integrated Mode \(Reporting Services Configuration\)](#)

[How to: Switch Server Modes \(Reporting Services Configuration\)](#)

Concepts

[Requirements for Running Reporting Services in SharePoint Integrated Mode](#)

[Configuring Reporting Services for SharePoint 3.0 Integration](#)

[Security Overview for Reporting Services in SharePoint Integration Mode](#)

[Installing SQL Server Reporting Services](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Configuring Reporting Services for SharePoint 3.0 Integration



[See Also](#)



[Configuring a SQL Server Installation](#) > [Reporting Services Configuration](#) >

New: 12 December 2006

You can configure a deployment of SQL Server 2005 Reporting Services with Service Pack 2 (SP2) to work with a deployment of Microsoft Windows SharePoint Services 3.0 or Microsoft Office SharePoint Server 2007. You must perform configuration steps on both a report server and a deployment of the SharePoint product or technology before users can publish reporting documents to SharePoint libraries and get report viewing and management services on SharePoint sites. The configuration process establishes server connection information; therefore, it is important that you follow the steps in order.

This topic provides an overview of the configuration steps for each server, and then concludes with the tasks you must follow in the [How to Configure Server Integration](#) section. SQL Server Setup will not install a report server in SharePoint integrated mode by default. You must manually configure the report server for SharePoint integration mode after you have run Setup.

▣ **Overview of Report Server Configuration**

To configure a report server for SharePoint integration, you must have an existing stand-alone report server or a scale-out deployment. You must perform the following tasks on the report server:

- Apply SP2, if you have not done so already.
- Install the Windows SharePoint Services Web front-end on the report server computer. This step is only necessary if you deploy the report server and the SharePoint technology instance on separate computers. Installing the Web front-end allows you to join the report server computer to a SharePoint farm.
- Optionally, configure the Report Server Web service and Windows service to run under domain accounts. This step is only necessary for a distributed server deployment that puts the SharePoint databases on the same computer as the report server.
- Create a report server database in SharePoint integrated mode. Doing so sets server properties on the report server and activates a new SOAP report server management endpoint that is used for integrated operations. Use the Reporting Services Configuration tool to create the database.

If you are integrating a scale-out deployment, all report server instances that are part of the deployment must run in SharePoint integrated mode. You cannot have a combination of server modes in the same scale-out deployment.

About Service Account Configuration

A report server is implemented as a Web service and as a Windows service which run under built-in accounts or Windows user accounts. In SharePoint integrated mode, both services connect to the SharePoint configuration and content databases with write and execute permissions.

In most cases, the default service account settings are valid for a connection to a SharePoint database. However, if the instance of the SharePoint Web application is on one computer, and the report server instance and the SharePoint databases are on a different computer, you cannot use built-in machine accounts such as Network Services or Local System for the Report Server service accounts.

When SharePoint databases run on a remote computer, the SharePoint Web application explicitly denies database access to the machine accounts of a remote computer. If the report server is on the same computer as the SharePoint databases, and if either of the Reporting Services service accounts run under a built-in account, they will be denied access accordingly. Configuring the services to run as Windows user accounts ensures that the report server will not be denied database access because they are running under a prohibited account. For more information about the service account connection to a SharePoint technology instance, see [Security Overview for Reporting Services in SharePoint Integration Mode](#). For more information about configuring the service accounts, see [How to: Configure Service Accounts \(Reporting Services Configuration\)](#).

About the Report Server Database

A report server database provides internal storage for one or more report server instances. A report server database can support native mode operations or SharePoint integration mode, but not both. The contents in the database are mode-specific and are not interchangeable across server modes. If you create a report server database to support SharePoint integrated mode, you cannot automatically convert or migrate the database to run with a native mode report server instance later. A report server database is used in the following ways depending on the server mode:

- When created for native mode operations, the report server database is the sole repository for persistent data used by the report server.
- When created for SharePoint integrated mode, the report server database stores server properties, report execution snapshots, report history, subscription definitions, and schedules. It stores a secondary copy of reports, report models, shared data sources, and resources to improve processing performance on the server. Primary storage for report documents is in the SharePoint content databases. For more information, see [Storing and Synchronizing Report Server Content With SharePoint Databases](#).

Creating a report server database for SharePoint integrated mode is an important first step in configuring the report server for SharePoint integration. You can use the Reporting Services Configuration tool to create the report server database and configure the connection to the report server. Only the report server connects to the report server database. The Windows SharePoint Services or Office SharePoint Server instance never connects to or retrieves data from the report

server database.

▣ **Configuring SharePoint Deployments for Report Server Integration**

To configure integration on the server running the SharePoint product or technology, you must download and install the Reporting Services Add-in for SharePoint Technologies. The add-in installs program files and application pages, including pages that you open in Central Administration to set the report server URL and other integration settings. You can specify report server integration settings for a SharePoint farm or a stand-alone SharePoint Web application.

Note:

Only one report server can be added to a SharePoint farm. The report server can be a stand-alone report server installation, or it can be a scale-out deployment that is accessed through a single URL.

SharePoint products and technologies can be deployed as a farm or as a stand-alone server. A farm contains one or more virtual servers, where each virtual server is a SharePoint Web application. A report server integrates at the farm level. From the perspective of storage integration, there can only be one report server database for a farm. You can use SharePoint Central Administration to manage integration settings for all the servers in the farm.

Each SharePoint Web application in a farm can be configured to use different authentication providers. This does not affect report server integration in any way. The report server deployment will handle requests for each server in the farm regardless of the authentication provider it uses.

Each Web application has a default top-level site. You can set permissions differently on each top-level site to vary the availability of report server items and operations across all the servers in a farm.

▣ **How to Configure Server Integration**

Configuring the servers for integrated operations requires multiple tools and steps. The following list has all the tasks you must perform. These tasks include links to detailed instructions that you can use to complete a specific step. For an overview of how server integration works, see [Reporting Services and SharePoint Technology Integration](#).

Installation Order

Before you can configure integration, you must install Reporting Services and the SharePoint product or technology you want to use. You can install Reporting Services and the SharePoint product or technology in any order. However, the installation order and the deployment topology will determine whether you need to perform additional configuration steps in Internet Information Services (IIS) and when you can verify a successful installation.

For side-by-side deployments:

- If you install Reporting Services on a computer that already has a SharePoint technology instance, you will not be able to verify the Reporting Services installation until you have configured all integration settings.
- If you install a SharePoint technology instance on a computer that already has Reporting Services, the Report Server Web service will stop working until you edit the port assignment for the default Web site as described in the next section.

For distributed server deployments where the report server, the SharePoint Web application, and the server databases on different computers, you can use default settings that include the default Web site and port assignments. If you install the SharePoint Web application and the report server on different computers, but you put the SharePoint databases on the report server computer, you will run into connection errors unless you follow the guidelines about service account configuration described in the previous section. For instructions, see [How to: Configure Service Accounts \(Reporting Services Configuration\)](#).

Edit the Port Assignment for the Default Web Site

Installing the report server and the instance of the SharePoint product or technology side-by-side on the same computer requires that each Web application is assigned to different ports. If you chose the Basic Installation option or used default settings when installing the SharePoint product or technology, the **SharePoint Products and Technology Configuration Wizard** stopped the default Web site and assigned port 80 to the top-level site when configuring the SharePoint Web application. If you want to use the default Web site for a report server, you must choose a different port for the default Web site and then start the Web site.

1. Start IIS Manager.
2. Right-click **Properties** on the Default Web Site.
3. Specify a different port. By default, a SharePoint Web application uses port 80. To run the report server on the same computer under the default Web site, select a different unused port (for example, 8080).
4. Start the Default Web site. By default, this site is stopped when you create and configure the site collection for the SharePoint Web application. You must restart this site to access the report server.
5. If you already had a working installation of Reporting Services on the computer before you installed Windows SharePoint Services, verify that you can still access the report server by typing a URL that includes the port (for example, <http://example-server-name:8080/reportserver>). Although integration is not yet configured, the report server should still be operational as a stand-alone server.

Install Prerequisites on the Report Server

1. Apply SP2 to the report server.
2. If the report server and the SharePoint Web application are installed on separate computers, install and configure the Windows SharePoint Services object model on the report server computer. For instructions, see [How to: Install the Windows SharePoint Services Object Model on a Report Server Computer](#).

Configure a Report Server for SharePoint Integration

1. Start the Reporting Services Configuration tool and connect to the report server instance that you want to configure for SharePoint integration.

2. For a new installation only, on the Report Server Virtual Directory page, select the default Web site and enter a name for the virtual directory. Skip the page for configuring the Report Manager virtual directory. You cannot use Report Manager with a report server that is configured for SharePoint integrated mode. After the report server is configured, the Report Manager Virtual Directory page will be disabled.
3. On the Database Setup page, create a new report server database in SharePoint integrated mode by selecting the **Integrate with SharePoint Products and Technologies** option when you create the database. For instructions, see [How to: Create a Report Server Database for SharePoint Integrated Mode \(Reporting Services Configuration\)](#).
4. Back up the encryption key.
5. If you are working with a new installation of Reporting Services, you should also configure the unattended report processing account if you want to support unattended report execution. Optionally, configure the report server for e-mail delivery. For more information, see [Configuring an Account for Unattended Report Processing](#) and [Configuring a Report Server for E-Mail Delivery](#).
6. Close the Reporting Services Configuration tool.
7. If you used the default Web site configured for a specific port, edit the RSReportServer.config file to add the port to the **UrlRoot** configuration setting (for example, http://example-server-name:8080/reportserver). Reporting Services uses the **UrlRoot** configuration setting to construct links in e-mail messages that resolve to reports that are processed on a report server.

After you configure the report server, the next step is to install the Reporting Services Add-in and configure integration settings for the SharePoint Web application.

Install the Reporting Services Add-in and Configure Integration on the SharePoint Web Application

You must be a SharePoint farm administrator to install the Reporting Services Add-in.

1. Download and run the Setup program for the Reporting Services Add-in. For more information, see [Installing the Reporting Services Add-in](#).

2. In Administrator Tools, click **SharePoint 3.0 Central Administration**.

 **Important:**

To complete the steps below, you must be a member of the SharePoint Farm Administrator group and a Site Collection administrator.

3. Click **Application Management**. You should see a new section named Reporting Services. This section has links to pages used to specify integration settings. At a minimum, you must set the report server URL, select an authentication mode, and grant report server access to the SharePoint databases. For instructions, see [How to: Set Report Server Integration in SharePoint Central Administration](#).
4. Close Central Administration.

Verify the installation

Verify integration by starting the Reporting Services Configuration tool and opening the SharePoint Integration page to view server status. You should see a link that you can click to start SharePoint Central Administration.

You can also verify an installation by opening a browser window to the report server. The URL should be similar to `http://example-server-name:8081/reportsrvr`. This URL is to the proxy server endpoint. After you add report server items to a SharePoint library, you can open a browser window to this URL to view items on the site.

Finally, you should be able to open the SharePoint Web application and upload reports and other documents to a SharePoint library.

Add Report Server Items

1. Open the SharePoint Web application. By default, the URL is `http://<your-server-name>`.
2. Set permissions to grant user access to the SharePoint site. Security is required, but if you want to inherit existing permissions, you do not have to set permissions on specific items. For instructions, see **Managing Permissions and Security for Report Server Items on a SharePoint Site**.
3. Add Reporting Services content types if you want to create new shared data

sources, report models, and Report Builder reports. For instructions, see **How to: Add Report Server Content Types to a Library (Reporting Services in SharePoint Integrated Mode)**.

4. Create shared data sources to use with reports and models. For instructions, see **How to: Create and Manage Shared Data Sources (Reporting Services in SharePoint Integrated Mode)**.
5. Create shared schedules to use with subscriptions or unattended report processing. For instructions, see **How to: Create and Manage Shared Schedules (Reporting Services in SharePoint Integrated Mode)**.
6. Publish report definition files (.rdl), data source files (.rds), and report model files (.smdl) to a SharePoint library. You can also use the Upload command in a SharePoint library to upload .rdl and .smdl files, but not .rds files. Depending on your permissions, you can also create reports in Report Builder and save them to a library. If the reports include references to shared data sources or external files, you must update the references. For more information, see **Deploying Reports, Models, and Shared Data Sources to a SharePoint Site** and **How to: Upload Documents to a SharePoint Library (Reporting Services in SharePoint Integrated Mode)**.

When you upload reports, make sure that the data source properties for the report are correct. The data source must have a valid connection string to an external data source. The credential type must be valid for your network topology. Specifically, if Kerberos authentication is not enabled for your domain, you cannot use the Windows integrated security credential type. Instead, you should specify stored credentials. After you verify the data source properties are correct, you can click the report name to open the report. The report will open in the Report Viewer Web Part automatically.

If the report does not open, check permissions on the report and the data source properties that define a connection to the external data source that provides data to the report. On a site or farm, SharePoint permissions are used to control access to reports, report models, shared data sources, and the Report Builder tool used to create ad hoc reports. If these items are not visible to users, you should verify that permissions are set correctly.

A report server that runs in SharePoint integrated mode cannot be managed through Report Manager or SQL Server Management Studio. For a complete list of supported and unsupported features, see [Features Supported by Reporting](#)

[Services in SharePoint Integration Mode.](#)

▣ **See Also**

Tasks

[How to: Set Report Server Integration in SharePoint Central Administration](#)

[How to: Create a Report Server Database for SharePoint Integrated Mode \(Reporting Services Configuration\)](#)

[How to: Switch Server Modes \(Reporting Services Configuration\)](#)

Concepts

[Backing Up and Restoring Encryption Keys](#)

[Deployment Modes for Reporting Services](#)

[Security Overview for Reporting Services in SharePoint Integration Mode](#)

Other Resources

RSReportDesigner Configuration File

Managing Permissions and Security for Report Server Items on a SharePoint Site

Deploying Reports, Models, and Shared Data Sources to a SharePoint Site

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Reporting Services and SharePoint Technology Integration



[See Also](#)



[Configuring a SQL Server Installation](#) > [Reporting Services Configuration](#) > [Configuring Reporting Services for SharePoint 3.0 Integration](#) >

New: 12 December 2006

You can configure a report server to run within a deployment of a SharePoint product or technology and use the collaboration and centralized document management features of Windows SharePoint Services 3.0 or Office SharePoint Server 2007 with Reporting Services. Running a report server as part of a larger SharePoint deployment provides these levels of integration:

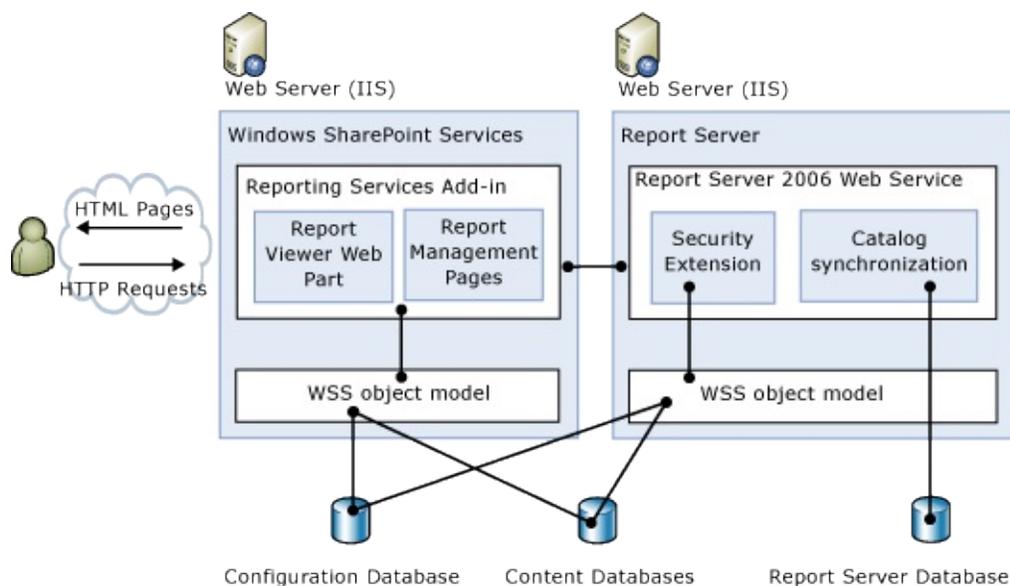
- Shared storage.
- Shared security.
- Same site access for all business documents, including reports, report models, and shared data sources.

▣ Server Integration Architecture

When you integrate a report server with an instance of a SharePoint product or technology, items and properties are stored in the SharePoint content databases. This provides a deeper level of integration between the server technologies that effects how content is stored, secured, and accessed.

Storing report items and properties in SharePoint content databases allows you to browse SharePoint libraries for report server content types, secure items using the same permission levels and authentication provider that controls access to other business documents hosted on a SharePoint site, use the collaboration and document management features to check reports in and out for modification, use alerts to find out if an item has changed, and embed or customize the Report Viewer Web part on pages and sites within the application. Depending on your permissions, you can view and sometimes create reports and report models on a SharePoint site.

The report server continues to provide all data processing, rendering, and delivery. It also supports all scheduled report processing for snapshots and report history. The following diagram shows how the server components work together:



When you open a report from a SharePoint site, the report server proxy server endpoint connects to a report server, creates a session, prepares the report for

processing, retrieves data, merges the report into the report layout, and displays it in the Report Viewer Web part. While the report is open, you can export it to different application formats, or interact with data by drilling into underlying numbers or clicking through to a related report. Export and report interaction operations are performed on the report server.

The report server synchronizes operations and data with Windows SharePoint Services and tracks information about the files it processes. When you modify properties or settings for any report server item, the change is stored in a SharePoint database and then copied to a report server database that provides internal storage to a report server.

▣ **Components That Provide Integration**

To combine the servers in a single deployment, you integrate an installation of Microsoft SQL Server 2005 Reporting Services with an instance of Windows SharePoint Services or Office SharePoint Server 2007.

Integration is provided through SQL Server 2005 Service Pack 2 and the SQL Server 2005 that reports Services Add-in for SharePoint Technologies. The Reporting Services Add-in is a freely distributable component that you can download from a Microsoft Web site and then install on a server that is running Windows SharePoint Services 3.0 or Office SharePoint Server 2007.

- On Windows SharePoint Services, the Reporting Services Add-in provides a proxy server endpoint, a Report Viewer Web part, and application pages so that you can view, store, and manage report server content on a SharePoint site or farm.
- On the report server, Service Pack 2 provides updated program files, a new SOAP endpoint, custom security and delivery extensions, a revised Reporting Services Configuration tool, scripts for granting access to SharePoint configuration and content databases, and a script for creating a report server database that supports SharePoint integrated mode. The report server must be configured to run in SharePoint integrated mode, dedicated exclusively to supporting report access and delivery through your SharePoint site.

After you install the Reporting Services Add-in on Windows SharePoint Services and configure the two servers for integration, you can upload or publish report server content types to a SharePoint library, and then view and manage those documents from a SharePoint site. Uploading or publishing report server content is an important first step; the Web Part and pages become available when you select report definitions (.rdl), report models (.smdl) and shared data sources (.rsds) on a SharePoint site.

▣ In This Section

The following topics in this section provide additional information about how the servers work together. For installation recommendations and configuration steps, see [Configuring Reporting Services for SharePoint 3.0 Integration](#).

[Features Supported by Reporting Services in SharePoint Integration Mode](#)

Summarizes feature availability for a report server that is integrated with SharePoint technologies.

[Storing and Synchronizing Report Server Content With SharePoint Databases](#)

Describes how report server integration is supported at the data layer. This overview provides an introduction to the data stores that are used and how content and properties are stored.

[Security Overview for Reporting Services in SharePoint Integration Mode](#)

Describes how report server integration is also implemented at the security layer. Access to report server content and operations is supported by the authentication and authorization functionality in SharePoint technologies.

[Viewing and Managing Report Server Items from a SharePoint Site](#)

Explains how report server integration extends to the Web application pages on SharePoint sites and in SharePoint Central Administration. With few exceptions, current users of Reporting Services will find that the functionality available on SharePoint sites is equivalent to Report Manager application pages.

▣ **See Also**

Concepts

[Deployment Modes for Reporting Services](#)

[Installing the Reporting Services Add-in](#)

[Requirements for Running Reporting Services in SharePoint Integrated Mode](#)

[Comparing Levels of Integration Across Versions of Reporting Services and SharePoint Technologies](#)

Other Resources

[Using the Report Viewer Web Part on a SharePoint Site](#)

[Deploying Reports, Models, and Data Sources on a SharePoint Site](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Features Supported by Reporting Services in SharePoint Integration Mode

 [See Also](#)



[Reporting Services Configuration](#) > [Configuring Reporting Services for SharePoint 3.0 Integration](#) > [Reporting Services and SharePoint Technology Integration](#) >

New: 12 December 2006

This topic provides a summary of the features that are available on a report server that is configured for integration with Windows SharePoint Services 3.0 or Office SharePoint Server 2007. To learn more about integration requirements and how the server technologies work together, see [Reporting Services and SharePoint Technology Integration](#) and [Requirements for Running Reporting Services in SharePoint Integrated Mode](#).

▣ Feature Summary

Configuring a report server to run in SharePoint integrated mode provides the following additional functionality that is only available when you deploy a report server in this mode:

- Use document management and collaboration features, including alerts and version control. A SharePoint site provides a unified portal for accessing and managing all documents in one place.
- Use SharePoint permissions and authentication providers to control access to reports, models, and other items.
- Use SharePoint deployment topologies to distribute reports over an Internet connection outside the firewall. A report server provides report and data processing services in the context of a larger SharePoint deployment that is configured for Internet access.
- Manage reports, models, data sources, schedules, and report history in custom application pages on a SharePoint site. You can set properties, define schedules and subscriptions, and create and manage report history on a SharePoint site the same way you create and manage them from other tools in SQL Server.
- View a report in a new customizable Report Viewer Web Part. The Web Part includes page navigation, search, print, and export features.
- Publish or upload reports, report models, resources, and shared data source files to a SharePoint library, including Report Center in Office SharePoint Server.

Use Report Designer, Model Designer, and Report Builder to create report server items. If you are using the Service Pack 2 version of these tools and if you configure the report server to run in SharePoint integrated mode, you can publish the newly created items directly to a SharePoint library. You can also use the Upload action on a SharePoint site to add any report definitions and report models to a SharePoint library.

Because the report server processes report definitions in the same way regardless of the server mode you use, the report data and layout is unaffected by server mode. Any report that you can run in a native mode report server can run on a report server that is configured for SharePoint integrated mode.

- Create ad hoc reports using report models and Report Builder. You can

generate models and start Report Builder from a SharePoint site. You can add, open, and save ad hoc reports to a SharePoint site. If you are using the Enterprise Edition, you can deploy ad hoc reports that support infinite clickthrough functionality.

- Subscribe to and deliver reports to a SharePoint library using a new SharePoint delivery extension. You can also deliver reports through e-mail or to a shared folder. The report server delivery extensions are used to deliver reports.
- In Office SharePoint Server , connect a Report Viewer Web Part to a Filter Web Part or a Library Web Part in dashboards or Web Part pages.
- Program against a new SOAP endpoint to create custom applications that integrate with a SharePoint site. You can also use the updated Windows Management Instrumentation (WMI) provider to programmatically configure a report server instance that runs in SharePoint integrated mode.

▣ **Unsupported SharePoint Features**

Not all features are available for integrated operations. On the SharePoint site, you cannot use SharePoint Outlook calendar integration or SharePoint scheduling for report server document files. Personalization is also not supported.

▣ **Unsupported Reporting Services Features**

The following features are not available on a report server that runs in SharePoint integrated mode:

- URL addressing is different in SharePoint integrated mode. SharePoint URLs are used to reference reports, report models, shared data sources, and resources. The report server folder hierarchy is not used.
- Reporting Services custom security extensions cannot be deployed or used on the report server. The report server includes a special-purpose security extension that is used whenever you configure a report server to run in SharePoint integrated mode. This security extension is an internal component, and it is required for integrated operations.
- Report Manager or Management Studio cannot be used to manage a report server instance that is configured for SharePoint integration.
- Data-driven subscriptions are not available. This applies to all editions.
- The rs.exe command line utility is not supported. The utility does not support the SOAP endpoint used for programmatic access to a report server that runs in SharePoint integrated mode.
- Linked reports are not supported.
- My Reports is not supported.
- Job management features that allow you to stop a long-running report process are not supported.
- Batching methods will not be supported.

▣ **See Also**

Reference

[Editions and Components of SQL Server 2005](#)

[Features Supported by the Editions of SQL Server 2005](#)

Concepts

[Reporting Services and SharePoint Technology Integration](#)

Other Resources

Introducing Reporting Services

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Viewing and Managing Report Server Items from a SharePoint Site



[See Also](#)



[Reporting Services Configuration](#) > [Configuring Reporting Services for SharePoint 3.0 Integration](#) > [Reporting Services and SharePoint Technology Integration](#) >

New: **12 December 2006**

When you configure a report server to run in SharePoint integrated mode, you can view and manage reports and other report server items from a SharePoint site.

Custom application pages are added to a SharePoint Web application when you install the Reporting Services Add-in on an instance of Windows SharePoint Services 3.0 or Office SharePoint Server . There are pages to set data source properties, report history, report processing options, schedules, subscriptions, report parameters, and create shared schedules. You can manage report server items on a SharePoint site the same way you create and manage them from other tools in SQL Server.

To access the application pages, select item-specific actions from a drop-down menu. Depending on the item and your permissions, you might also be able to create reports in Report Builder, generate models, and set model item security.

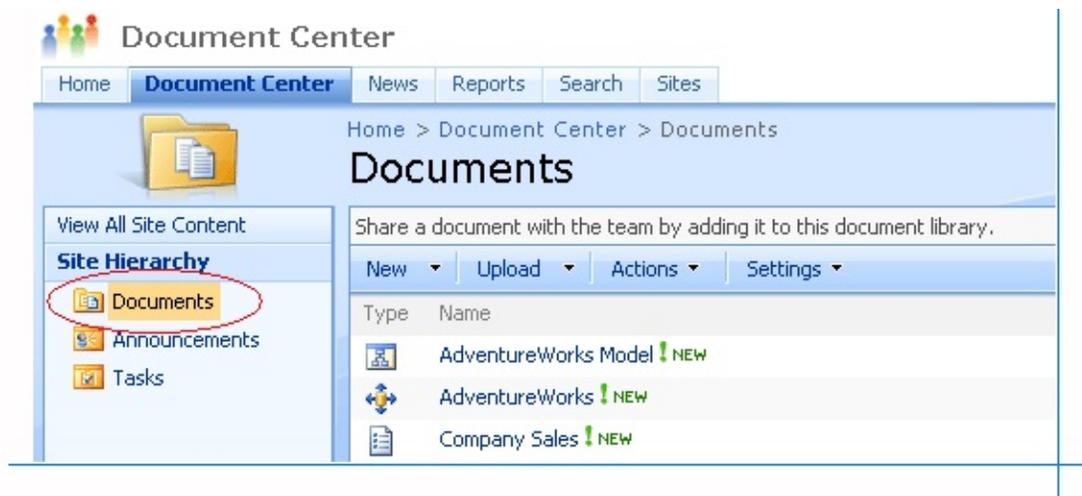
Note:

The Reporting Services Add-in adds report server integration pages to SharePoint Central Administration so that server administrators can configure a connection to a report server instance and update service account information for a SharePoint Web application. For more information, see [Configuring Reporting Services for SharePoint 3.0 Integration](#).

▣ Finding Report Server Items on a SharePoint Site

Before you can set properties, you must first be able to locate the item. Report server items are always stored in libraries or in a folder within a library. By default, document libraries for a site are accessed from the Quick Launch pane on the side of the main view area. The library icon and placement on the Quick Launch page can help you quickly identify a document library. A library is further identified by the **New**, **Upload**, and **Actions** items on the menu bar that are used to create and manage items in the library.

To distinguish report server items from other items on a SharePoint site, you can use the icon to visually identify an item, or pause the mouse cursor over the type and read the file extension. The following image shows a report model, a shared data source, and a report definition in the **Documents** library:



▣ **Viewing Reports**

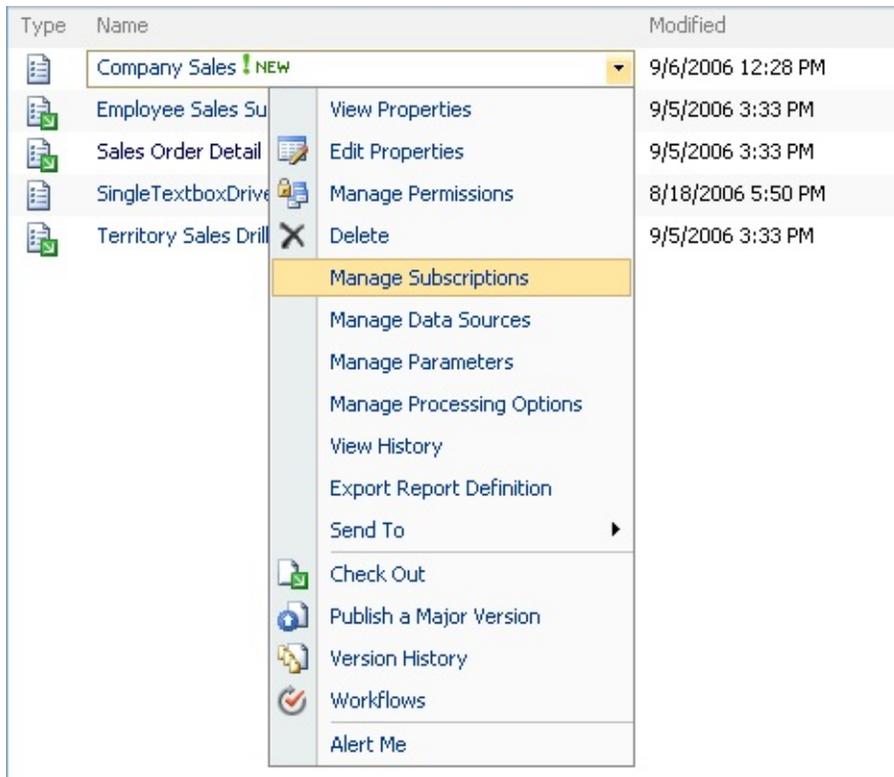
Report definitions (.rdl files) that you upload to a SharePoint library are viewed through a Report Viewer Web Part that is installed by the Reporting Services Add-in. An .rdl file association is defined automatically when you install the add-in. When you select a report, it opens automatically in the Web Part. After the report is open, you can use the report toolbar that is included in the Web Part to navigate pages, search, zoom, export, and print the report.

▣ **Managing Items Through Actions**

Management tasks are supported through actions on a drop-down menu for each item:

1. Open the SharePoint Web application or site that contains the content you want to manage.
2. In a library, find the item.
3. Click the item to select it.
4. Click the down arrow that appears when you select the item.
5. Choose an action from the drop-down menu. Several actions, such as **View Properties** and **Edit Properties**, are common to all items that are stored in a library.

Depending on your permissions, each item has common actions that are standard for items that are stored in a SharePoint library. **View Properties** and **Edit Properties** are examples of common actions. Custom actions provide item-specific management functionality. The following image shows the actions for a report definition. Examples of custom actions for a report definition include **Manage Subscriptions** and **Manage Processing Options**:



For more information about how to use these actions for each report server item, see **Report Server in SharePoint Integrated Mode How-to Topics** in SQL Server Books Online.

▣ **See Also**

Concepts

[Features Supported by Reporting Services in SharePoint Integration Mode](#)

[Reporting Services and SharePoint Technology Integration](#)

[Configuring Reporting Services for SharePoint 3.0 Integration](#)

Other Resources

Managing Permissions and Security for Report Server Items on a SharePoint Site

How to: Add the Report Viewer Web Part to a Page (Reporting Services in SharePoint Integrated Mode)

Using the Report Viewer Web Part on a SharePoint Site

How to: Upload Documents to a SharePoint Site (Reporting Services in SharePoint Integrated Mode)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Security Overview for Reporting Services in SharePoint Integration Mode

 [See Also](#)



[Reporting Services Configuration](#) > [Configuring Reporting Services for SharePoint 3.0 Integration](#) > [Reporting Services and SharePoint Technology Integration](#) >

New: **12 December 2006**

When you configure a report server to run in SharePoint integrated mode, the report server uses the authentication provider and permissions defined in the SharePoint Web application to control access to report server items and operations.

Permission to access items and operations is granted through SharePoint security policies that map a user or group account with a permission level, relative to an item. Conceptually, it is identical to how role assignments are used in a native mode report server deployment, where a role assignment maps a user or group account to a set of allowable tasks relative to an item. As is common with most role-based authorization models, SharePoint security provides permission inheritance to reduce the complexity and overhead of maintaining a large number of policies.

If you are deploying report server content types to a SharePoint site, you need to know the following:

- How connections and requests are made between the two servers. The authentication provider used in the SharePoint Web application determines whether you can subsequently use the Windows integrated security option to access external data sources used by a report.
- How to use default security to add, manage, and access report server items and operations. For more information, see **Managing Permissions and Security for Report Server Items on a SharePoint Site** and **Using Default Security and Permissions in a SharePoint Web Application to Access Report Server Items and Operations** in SQL Server Books Online.

- How to override default security by setting permissions on specific reports or operations. For more information, see **How to: Secure Items and Operations in SharePoint Integrated Mode (Reporting Services Configuration)** in SQL Server Books Online.

Before you can set permissions, you must configure each server for integration. For more information, see [Configuring Reporting Services for SharePoint 3.0 Integration](#).

▣ **Authentication Providers in SharePoint Technologies**

A SharePoint Web application can use Windows authentication or Forms authentication. A report server will process requests from either one. You can configure authentication in these combinations:

- Windows authentication with integrated security (Kerberos is enabled)
- Windows authentication, without impersonation or delegation
- Forms authentication

📌 **Note:**

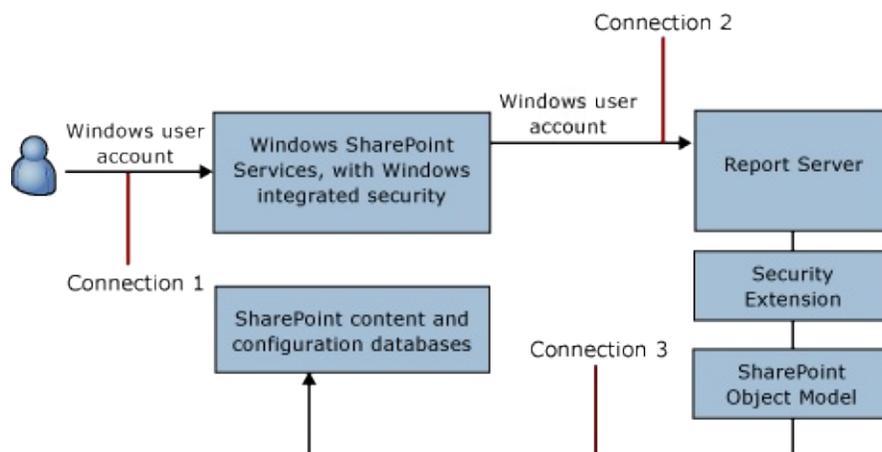
Both Reporting Services and SharePoint products and technologies support forms authentication. The implementation is different for each product group and they are not compatible. Reporting Services custom authentication extensions are not supported for report servers that run in SharePoint integration mode.

▣ Sending Requests to a Report Server

All requests for a report server item or operation must be a valid authenticated request. The authentication provider you are using determines how that request is processed.

Windows Integrated Security

If the SharePoint Web application is configured for Windows authentication and Kerberos is enabled, the connection from the SharePoint Web application to the report server is through the impersonated or delegated credentials of the current Windows user. The following diagram shows the connections when a report server is configured for SharePoint integration, and the SharePoint Web application uses Windows authentication with Kerberos enabled.



Connection 1

A user accesses a SharePoint site under the user token created when the user logged on to the network. It contains the user identity and group membership. The SharePoint Web application authenticates the user. The user requests a report server item or operation.

Connection 2

The SharePoint Web application sends the token and the request to the report server. The connection request is sent under the Windows identity of

the user. The report server validates whether the user has permission to access the report server.

Connection 3

If the validation is successful, the report server will verify that the user has permission to access the item or operation.

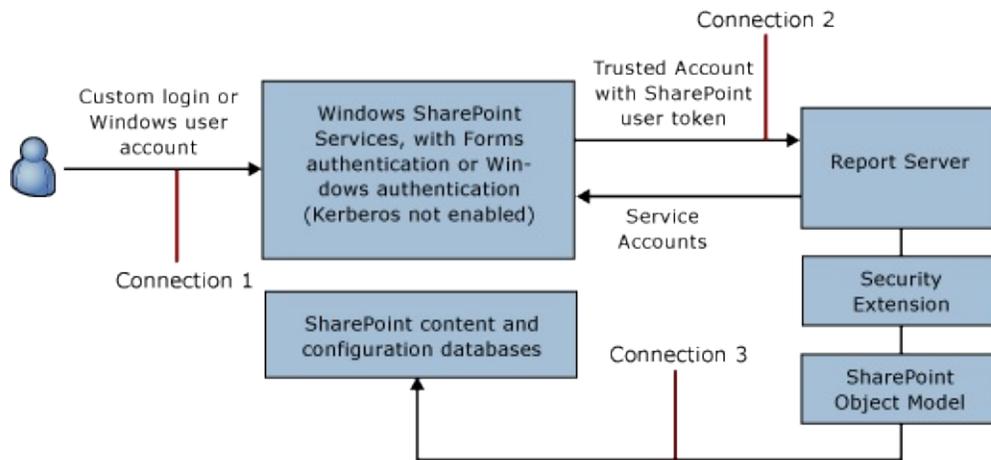
A report server uses an internal security extension to verify user permissions. The report server uses the extension to call in to the Windows SharePoint Services object model to check the permissions that are stored in the SharePoint content databases. You cannot configure or manage this security extension. It is an internal component used for report servers that run in SharePoint integrated mode.

If the user has permission to access the report or other item or operation, the request is serviced.

By using Windows integrated security, you can eliminate the classic "double-hop" issue wherein Windows credentials expire after a single connection. It can also expand the set of options that are available to you when you configure data source connections for reports and models. If the Windows user identity is used to connect to the report server, the report server can use that identity during report processing to retrieve data from external data sources. This means that when you set data source properties on a report, you can select the **Windows integrated security** option for the data source connection. For more information, see **Specifying Credential and Connection Information** in SQL Server Books Online.

Windows or Forms Authentication and Trusted Accounts

If the SharePoint Web application is configured for Forms authentication, the connection to the report server is sent across the network under a predefined trusted account that has permission to impersonate a SharePoint user on the report server. The same approach is used if the SharePoint Web application is configured for Windows authentication and Kerberos is not enabled. The following diagram shows the connections when trusted accounts and SharePoint user identities are used.



Connection 1

A user logs on to a SharePoint site. The SharePoint Web application authenticates the user. The SharePoint Web application translates the user identity to a SharePoint user identity (SPUser). A new user token is created for that user in the context of SPUser. It contains the user identity and group membership. The user requests a report server item or operation.

Connection 2

The SharePoint Web application impersonates the SharePoint user identity on the request to the report server. The connection request is sent that uses a trusted account. The account is the process identity of the SharePoint Web application.

The report server validates whether the connection request is from a trusted account by comparing it to account information that the report server retrieved from the SharePoint configuration databases when the report server started. On a report server, the trusted account is a Windows user with permission to impersonate SharePoint Web application. It is used only to impersonate SPUser. It is not allowed access to report server items and operations.

Connection 3

If the validation is successful, the report server will verify that SPUser has permission to access the item or operation.

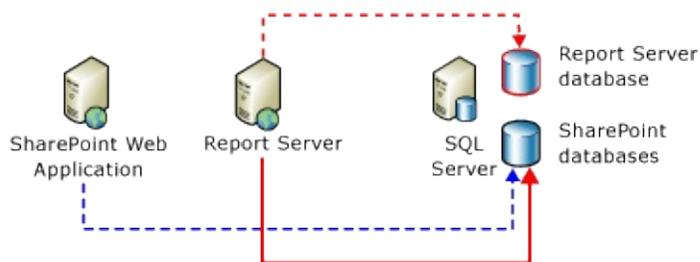
The report server uses an internal security extension to verify user permissions, call in to the Windows SharePoint Services object model, and check the permissions that are stored in the SharePoint content databases. If the user has permission to access the report or other item or operation, the request is serviced.

Account Expiration and Subscription Processing

When you create a subscription to a report, the report server will store the SPUser account information so that it can verify that the user has permission to view the report at the time of delivery. If SPUser has expired, the subscription will fail and return an **rsSharePointError** error. A farm-level property named **TokenTimeout** determines how long SPUser is valid.

□ How the Report Server Accesses SharePoint Content Databases

Both the SharePoint Web application and the report server connect to their respective databases to store application state and other data, but the report server must also connect to the SharePoint databases to store and retrieve items, properties, and configuration settings. The following diagram shows the server connections to the various databases.



A SharePoint Web application can use a local or remote database for internal storage. If the SharePoint databases are on remote computers, a domain account must be used for the connection.

A report server can use a local or remote database for internal storage. For either type, the database connection can be made by using a domain account, a SQL Server login, or a built-in account such as **Network Service** or **Local System**.

Report Server Connection to the SharePoint Databases

In Reporting Services, both the Web service and Windows service require access to SharePoint databases. The service accounts for both services run as trusted users in a SharePoint Web application, and are automatically granted permission to access the SharePoint databases.

The connection is managed internally; it is configured when you use SharePoint Central Administration to point a SharePoint Web application to a report server and set the trusted accounts. In contrast with the report server connection to its own databases, which you can set or modify using the Reporting Services Configuration tool, you cannot explicitly configure or manage the report server connection to the SharePoint databases.

Running a report server in SharePoint integrated mode introduces constraints on how you configure the service accounts in Reporting Services. Use these guidelines when configuring service accounts:

- Choose accounts that have network logon permissions if the Report Server service accounts must connect to the SharePoint databases on a remote computer.
- Avoid built-in accounts (such as **Local System** or **Network Service**) if the report server and the SharePoint databases are on one computer, and the SharePoint Web application is on a remote computer. When SharePoint databases run on a remote computer, the SharePoint Web application explicitly denies database access to built-in accounts that are defined on that remote computer. This means that all services that run under built-in accounts on that computer cannot connect to SharePoint databases.
- For all other topologies that place the servers and databases on the same computer or different computers, the Reporting Services service accounts can be configured as domain accounts or built-in accounts.

Errors Connecting to the SharePoint Databases

If the report server cannot access the SharePoint databases and there is a configuration error (for example, if the service accounts or passwords are not valid or if a local instance of the Windows SharePoint object model is not installed), an **rsServerConfigurationError** error occurs. For all other connection errors, the **rsSharePointError** error is returned, along with additional error information from the local Windows SharePoint Services instance.

▣ **See Also**

Concepts

[Storing and Synchronizing Report Server Content With SharePoint Databases](#)

Other Resources

Using Default Security and Permissions in a SharePoint Web Application to Access Report Server Items and Operations

Comparing SharePoint Groups to Role Definitions in Reporting Services

How to: Secure Items and Operations in SharePoint Integrated Mode (Reporting Services Configuration)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Storing and Synchronizing Report Server Content With SharePoint Databases

 [See Also](#)



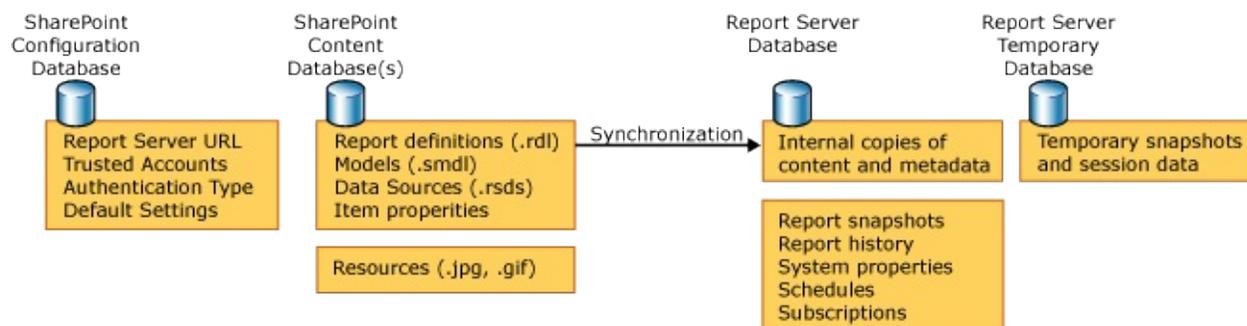
[Reporting Services Configuration](#) > [Configuring Reporting Services for SharePoint 3.0 Integration](#) > [Reporting Services and SharePoint Technology Integration](#) >

New: 12 December 2006

When you configure a report server to run in SharePoint integration mode, the report server uses the SharePoint configuration and content databases as well as its own internal databases to store content and metadata.

Both Reporting Services and Windows SharePoint Services are distributed server applications that allow you to run services and internal databases on separate computers. Each server stores different kinds of data. Multiple SQL Server relational databases provide the internal storage for both servers. Knowing which type of data is stored in each one can help you understand how the servers work together. It also provides background information that can help you make decisions about how to allocate disk space and schedule database backups.

The following diagram shows the complete set of databases used in a report server deployment that runs in SharePoint integrated mode.



▣ **SharePoint Configuration Database**

Every SharePoint Web application has a configuration database that stores application settings. When you set up an instance of Windows SharePoint Services or Office SharePoint Server to work with a report server, you specify configuration settings in SharePoint Central Administration. The settings that you specify are stored in this database. Settings include the URL of the report server, account information for the Report Server Web service and Windows service, information about the authentication provider used on the server, and site-level settings that limit or enable report history and logging. For more information, see [Configuring Reporting Services for SharePoint 3.0 Integration](#).

▣ **SharePoint Content Database**

Each SharePoint Web application has one or more content databases that store the documents and properties that are accessed and managed through the server. For a report server that is configured for SharePoint integrated mode, the SharePoint content database provides the primary storage for published reports, report models, shared data sources, resources, properties, and permissions.

▣ **Report Server Database and the Report Server Temporary Database**

Each report server instance or scale-out deployment uses a single report server database to store persistent data. Temporary data is isolated in a secondary database. This enables you to create different backup schedules and set different growth properties for each one. There is one temporary database for each report server database. It stores session data and temporary snapshots that are created only for subscription processing, interactive reporting, or report caching as a performance enhancement.

The data that is kept in a report server database includes schedules, subscriptions, and snapshots for report history or report execution. The report server database also maintains internal copies of the content files that are stored in the SharePoint content database. Storing internal copies improves performance by minimizing the number of times a file has to be sent to the report server for processing. Synchronization and verification checks ensure that the reports, models, and data sources are the same.

✍ **Important:**

A report server database can support native mode operations or SharePoint integration mode, but never both. If you create a report server database to support SharePoint integrated mode, you cannot convert or migrate the database to run with a native mode report server instance later. The report server database contains mode-specific metadata (specifically, item location and permission information) that is not compatible in the other mode.

▣ **Database Connections, Permissions, and Logins**

A user who configures a report server or an instance of Windows SharePoint Services or Office SharePoint Server must be a local administrator or a member of the built-in Administrators group on the computer.

Both the Report Server Web service and Windows service require a database connection to the SharePoint content database, but not to the configuration database. Report server integration settings are stored in the configuration databases using the login and permissions that are created for users of the Central Administration tool when Windows SharePoint Services is installed.

In contrast, both the Report Server Web service and Windows service create, update, and delete items and properties in the SharePoint content database. The connection information, login, and database permissions are configured when you specify the trusted accounts in Central Administration.

The Windows SharePoint Services or Office SharePoint Server instance does not add or retrieve data from the report server databases. All requests are directed to the report server, which in turn accesses the report server databases using its own connection settings.

Permission to view an item or perform an operation on a report server is managed by the Windows SharePoint Services or Office SharePoint Server instance. Authentication and security checks occur before the request is sent to the report server. Each request is accompanied by a security token that includes permission information that determines whether the request is serviced or denied.

▣ Database Maintenance

The report server performs routine maintenance to remove orphaned report snapshots, report history, schedules, and subscriptions from the report server database that are deleted on the SharePoint site.

At daily intervals, the report server verifies that items stored in the report server database are associated with a report that exists in a SharePoint content database. To modify the frequency of the clean up process, add the **DailyCleanupMinuteOfDay** configuration setting in the RSReportServer.config file. The clean up process runs in the background and uses the schedule information that is defined in the configuration files. You cannot run the process on demand.

If the instance of Windows SharePoint Services or Office SharePoint Server is not available (for example, if the server is offline), no cleanup operation occurs. If you restore a report server database that has old data that does not correspond to the items currently stored in the SharePoint database, the items will be removed the next time that the clean up process runs.

The clean up process only deletes content; it does not synchronize or update the report server items that are stored in the SharePoint content database. By default, the clean up process runs at 2:00 a.m. each day.

To specify a different time, you must add **DailyCleanupMinuteOfDay** to the RSReportServer.config file and set a different value. For example:

 [Copy Code](#)

```
<Add Key=" DailyCleanupMinuteOfDay " Value="120" />
```

The setting is not in RSReportServer.config file by default. To add it, copy the entry and place it under the **Configuration** element alongside the other **Add** elements in that section. The value is minutes starting at 12:00 A.M., with a minimum value of 30 and a maximum of 1380 (23 hours).

▣ Synchronization and Verification Operations

To ensure that the copies kept for internal processing on the report server are the same as the original items in the SharePoint library, the report server performs synchronization and item verification steps before processing a request.

Synchronization is a process that creates, updates, or deletes an internal copy of an item that is stored in the report server database. The report server retrieves items and properties from a SharePoint library and stores them in the report server database. Timestamp information is the basis of comparison. The timestamp is stored in UTC format. If the report server and the SharePoint site or farm are in different time zones, the timestamp is converted to local server time as it is stored.

Verification is a step that determines whether synchronization is necessary. If the timestamp for the original item and the internal copy is the same, no synchronization occurs.

How items are synchronized

Synchronization is performed by the report server. It is a background process that is triggered automatically whenever an item is created, updated, or retrieved. You cannot manage, configure, or explicitly control synchronization operations. Synchronization is scoped to specific requests to create or update an item. There is no global synchronization operation that aligns all the items and properties at the same time. To manually synchronize a report, report model, or shared data source, you must run it.

Internal copies of an item are created or updated when the item does not exist in the report server database or the item in the report server database is older than the corresponding item in the SharePoint library. An internal copy is created in response to the following events:

- Request for a report or report model. This creates an internal copy of a report, report model, and any shared data sources that are referenced by either item.
- Publish a report or report model from Report Designer or Model Designer

to a SharePoint library. After the item is added to the SharePoint content database, a copy of the report or model is created and added to the report server database. If you also publish a shared data source along with the report model, an internal copy of the shared data source will not be created until it is used.

Using the **Upload** action on a SharePoint site will delay the creation of an internal copy in the report server database. For any report server item that you upload, an internal copy is created when the item is requested.

No internal copies are ever made of external image files used in a report, or of schedules, snapshots, or subscriptions that are stored exclusively in the report server database.

About Configuration Settings and Permissions

Although configuration settings must be defined correctly on both servers, they are excluded from synchronization processes. If you change the report server service accounts, the URLs, authentication type, or the server execution mode of the report server so that it runs in native mode instead of SharePoint integrated mode, you must explicitly update server configuration settings in SharePoint Central Administration or Reporting Services Configuration tool.

Permissions that are set on a SharePoint site or farm are used by the report server. There is no synchronization of permissions between a SharePoint site or farm and a report server.

▣ **Server Availability and Synchronization Failures**

Synchronization of items, properties, and operations requires that all computers and connections that are part of a deployment are available and operational. Synchronization will not occur when the following conditions are true:

- Servers or backend databases are taken offline or are otherwise unavailable.
- Connection information for any of the servers or backend databases becomes invalid. For example, if you change a URL incorrectly or you change a service account or password.
- Connection information is modified without taking servers offline first, disrupting the timing and flow of synchronized operations.

For the cases where synchronization cannot occur because the servers cannot connect to each other or their backend databases, the report server will reconcile out-of-sync content or metadata when the connection is restored and when a specific report, report model, shared data source, subscription is requested.

The **rsItemNotFound** error will occur if synchronization fails. This means that the item was not found in a SharePoint library.

▣ **See Also**

Tasks

[How to: Create a Report Server Database for SharePoint Integrated Mode \(Reporting Services Configuration\)](#)

Concepts

[Configuring a Report Server Database Connection](#)

[Configuring Reporting Services for SharePoint 3.0 Integration](#)

Other Resources

Report Server Database

RSReportServer Configuration File

Report Server Database

Deploying Reports, Models, and Shared Data Sources to a SharePoint Site

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Comparing Levels of Integration Across Versions of Reporting Services and SharePoint Technologies

 [See Also](#)



[Reporting Services Configuration](#) > [Configuring Reporting Services for SharePoint 3.0 Integration](#) > [Reporting Services and SharePoint Technology Integration](#) >

New: 12 December 2006

A report server configured to run in SharePoint integrated mode within a larger deployment of a SharePoint 3.0 product or technology provides a deeper level of integration than previously available. This new integration capability is introduced in SQL Server 2005 Service Pack 2 (SP2) and requires the 3.0 release of SharePoint products or technologies.

Earlier releases of Reporting Services also provided integration support for SharePoint products and technologies. Specifically, SQL Server 2000 Reporting Services SP2, SQL Server 2005, and SQL Server 2005 SP1 provided two Web Parts (Report Explorer and Report Viewer) that you could register and use in a SharePoint Web application. The Web Parts were designed for SharePoint version 2.0 technologies. The Web Parts continue to be available in SP2 and can be used in a deployment of a SharePoint 3.0 product or technology. The Web Parts have not been updated to include new features or capabilities; if you are currently using those Web Parts, they will continue to provide the same level of functionality as before.

The following table compares newer and older levels of integration. If you are planning a new deployment, or evaluating the new integration features for your organization, the table can help you weigh the advantages and disadvantages of each implementation.

Comparison point	SharePoint integrated mode	SharePoint 2.0 Web Parts
		SQL Server 2000 Reporting

<p>Product and version requirements</p>	<p>SQL Server 2005 SP2 only. Windows SharePoint Services 3.0 or Office SharePoint Server 2007.</p>	<p>Services Service Pack 2 and carried forward to SQL Server 2005. Windows SharePoint Services 2.0 and SharePoint Portal Server. The Web Parts are backward compatible on Windows SharePoint Services 3.0 and Office SharePoint Server 2007.</p>
<p>Key features</p>	<p>Unified portal for accessing and managing report server content.</p> <p>The front-end server is a SharePoint Web application; Reporting Services becomes a back-end server that provides processing and rendering capability behind a firewall.</p> <p>Authentication providers and permissions defined for the SharePoint Web application are used to control access to report server operations and content.</p> <p>The Report Viewer Web Part displays reports. It can be connected to other Web Parts and customized to fit the overall page design.</p> <p>Ad hoc reporting is included in the integrated feature set. You can generate models, secure models, and link reports to entities in a model from a</p>	<p>The Report Explorer Web Part shows a list of reports that are stored on a report server.</p> <p>The Report Viewer Web Part displays reports.</p> <p>The Web Parts are provided as-is with limited support for customization.</p> <p>The report server provides all storage, processing, and security for items managed by the server. Report access and content management are done through the report server tools.</p>

	<p>SharePoint Web application. You can create and edit reports using Report Builder from a SharePoint Web application.</p> <p>You can publish reports, models, and data sources to a SharePoint library.</p>	
Installation and configuration	<p>Download and install the Reporting Services Add-in on each Web front-end in a SharePoint farm.</p> <p>Configure integration by using SharePoint Central Administration and the Reporting Services Configuration tool.</p>	<p>Add Reporting Services virtual directories to the Windows SharePoint Services list of exclusions.</p> <p>Run a .cab file to install the Web Parts.</p> <p>Modify the Web.config file and manually copy files.</p>
Mode requirements	<p>Requires a report server that runs in SharePoint integrated mode.</p>	<p>Can be used on a native mode report server.</p>
Limitations	<p>No support for data-driven subscriptions, linked reports, Report Manager, or Management Studio.</p> <p>Programmatically, there is no support for batch operations or job management.</p>	<p>Requires that you configure and maintain separate security policies in the SharePoint Web application and the report server.</p> <p>Requires separate tools for managing content and operations on each server.</p> <p>Scheduled operations, data sources, report history, data processing and subscriptions must be managed using report</p>

server tools.

▣ **See Also**

Concepts

[Reporting Services and SharePoint Technology Integration](#)

[Requirements for Running Reporting Services in SharePoint Integrated Mode](#)

[Features Supported by Reporting Services in SharePoint Integration Mode](#)

[Storing and Synchronizing Report Server Content With SharePoint Databases](#)

[Security Overview for Reporting Services in SharePoint Integration Mode](#)

[Viewing and Managing Report Server Items from a SharePoint Site](#)

[Deployment Modes for Reporting Services](#)

[Configuring Reporting Services for SharePoint 3.0 Integration](#)

Other Resources

Deploying Reporting Services and Windows SharePoint Services 2.0 Side-by-Side

Viewing Reports with SharePoint 2.0 Web Parts

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Requirements for Running Reporting Services in SharePoint Integrated Mode

 [See Also](#)



[Configuring a SQL Server Installation](#) > [Reporting Services Configuration](#) > [Configuring Reporting Services for SharePoint 3.0 Integration](#) >

New: **12 December 2006**

You can integrate Microsoft SQL Server 2005 Reporting Services with Windows SharePoint Services or Office SharePoint Server by configuring a report server to run in SharePoint integrated mode and by installing a Reporting Services Add-in that adds infrastructure and application pages to a SharePoint Web application.

The following versions and editions can be used to support integrated operations:

- SQL Server 2005 Reporting Services with Service Pack 2. You cannot integrate a report server instance that is running earlier versions of the software.
Supported editions include Evaluation, Standard, Developer, and Enterprise. There is no support for this feature in the Workgroup Edition or in SQL Server 2005 Express with Advanced Services.
- Windows SharePoint Services 3.0 or Microsoft Office SharePoint Server 2007. If you are using Office SharePoint Server, you must have the Office Server Premium or the Office Server Standard Edition.
- Reporting Services Add-in for SharePoint Technologies. The add-in must be installed on the instance of the SharePoint technology you are using. There is only one version of the Reporting Services Add-in.

If your computer does not meet the requirements described in this topic, you can still access reports from a SharePoint site using Web Parts that shipped in earlier versions of Reporting Services. For more information, see [Comparing Levels of Integration Across Versions of Reporting Services and SharePoint Technologies](#).

▣ **Hardware and Software Requirements**

You must have 2 gigabytes of RAM on the computer that hosts the deployment of the SharePoint product or technology that will be integrated with a report server.

The computer must also satisfy the hardware and software requirements for each server technology. To review the requirements for SQL Server Reporting Services, see [Hardware and Software Requirements for Installing SQL Server 2005](#). For more information about requirements for SharePoint products and technologies, search for "Determine Hardware and Software Requirements" on the Microsoft Web site.

▣ **Permission Requirements**

You must be a member of the local administrators group to run Setup. You must run Setup on the local computer; you cannot run Setup remotely to apply SP2 or install the Reporting Services Add-in.

▣ **Web Site Requirements**

If you are running a side-by-side deployment of both server products on the same Web server, you must create a custom Web site for the report server or configure the default Web site to run under a different port. The Web site requirements are addressed in the configuration steps. For more information, see [Configuring Reporting Services for SharePoint 3.0 Integration](#).

▣ **Authentication Provider Requirements**

Anonymous access cannot be enabled on the SharePoint Web application. If Anonymous access is enabled, you will be able to configure integration settings but users will get an error when running a report.

All other authentication providers and options are supported. If you are configuring integration between a report server and a SharePoint farm, each SharePoint Web application in the farm can be configured to use different authentication providers.

▣ **Report Server Requirements**

In addition to SP2 and edition requirements, a report server instance must have a local installation of the Windows SharePoint Services object model. To determine whether the object model is present, Setup checks the registry for HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Shared Tools\Web Server Extensions\12.0 with **Sharepoint** set to **Installed**.

The Windows SharePoint Services object model is used to join the report server to a SharePoint farm. The object model stores a pointer to the configuration database of the SharePoint technology instance that the report server connects to. All connections from the report server to the SharePoint technology instance are made through the object model. For more information about server connections, see [Security Overview for Reporting Services in SharePoint Integration Mode](#). For more information about how to install the object model, see [How to: Install the Windows SharePoint Services Object Model on a Report Server Computer](#).

▣ **Requirements for the Reporting Services Add-in**

The Reporting Services Add-in is installed through a Setup program that you run at a command prompt on a server that has an instance of the SharePoint product or technology you want to use. Setup performs a prerequisite check before it installs the Reporting Services Add-in to verify the edition, version, and platform requirements.

Editions and Versions

The SharePoint product or technology must be Windows SharePoint Services 3.0 or Office SharePoint Server 2007. You cannot install the Reporting Services Add-in on a pre-release version of the software. You can use the Premium or Standard edition.

Language Availability for Multi-Lingual Deployment Scenarios

SQL Server 2005 and SharePoint products and technologies are localized into different languages. If you integrate a report server with a SharePoint product or technology instance that is localized into a language the report server does not support, you will get a combination of languages. For more information about language support, see [Deployment Modes for Reporting Services](#).

SQL Server Relational Databases

Both Reporting Services and Windows SharePoint Services use SQL Server relational databases for internal storage. Windows SharePoint Services installs SQL Server 2005 Express for its database. Reporting Services cannot use SQL Server Express for its database; it requires that you install the Evaluation, Developer, Standard, or Enterprise Edition of the SQL Server Database Engine.

If you want to install Reporting Services and a SharePoint technology instance on the same computer, you can run SQL Server Express and another edition of SQL Server 2005 side-by-side on the same computer. For more information, see [Installing Multiple Instances of Reporting Services](#).

Reporting Services has edition and version requirements for the report server database. For more information, see [Creating a Report Server Database](#).

Operating System Requirements

When you install the Reporting Services Add-in, Setup checks whether the computer is running the x86 or x64 version of Microsoft Windows Server 2003 or later.

▣ **See Also**

Concepts

[Configuring Reporting Services for SharePoint 3.0 Integration](#)

[Comparing Levels of Integration Across Versions of Reporting Services and SharePoint Technologies](#)

[Installing the Reporting Services Add-in](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Configuring Report Server Virtual Directories



[See Also](#)



[Configuring a SQL Server Installation](#) > [Reporting Services Configuration](#) >

Updated: 12 December 2006

In Reporting Services, URLs are used to access the report server, Report Manager, Report Builder, and individual reports. This topic describes the URLs that are used to access the report server and Report Manager, provides guidelines on how to create the virtual directories for each application, and includes a summary of the settings that are stored in the configuration files.



Note:

This topic does not describe URL access to Report Builder or to specific reports that are stored on the report server. For more information about URL access to these items, see **Starting Report Builder** and **Using a URL to Access Report Server Items** in SQL Server Books Online.

▣ **About URLs and Virtual Directories for a Report Server in SharePoint Integrated Mode**

If a report server is configured to run within a larger deployment of a SharePoint product or technology, URL and virtual directory construction will be affected in the following ways:

- All items on a SharePoint site are referenced through fully qualified URLs that include a file name extension used to distinguish between different types of report server items. You must specify fully qualified URLs when referencing shared data sources and models in reports, and when you specify a target server and folders for publish operations to a report server.
- For side-by-side deployments where the report server and the SharePoint technology instance are installed on the same computer, you cannot use `http://localhost/reportserver`. If `http://localhost` is used to access the SharePoint Web application, you must use a non-default Web site or a unique port assignment to access a report server.
- The URL and virtual directory for Report Manager cannot be configured. If you do configure it, it will no longer work after you deploy a report server in SharePoint integrated mode. Report Manager is not supported in this mode.

For more information about deployment modes and URL requirements for publishing items, see [Deployment Modes for Reporting Services](#) and **Deploying Reports, Models, and Shared Data Sources to a SharePoint Site**.

▣ About URLs and Virtual Directories for a Native Mode Report Server

In SQL Server 2005 Reporting Services, the report server and Report Manager are accessed through URLs that include dedicated virtual directories configured specifically for each component.

- The Report Manager virtual directory is used to access Report Manager. By default, the virtual directory name for Report Manager is **reports**.
- The report server virtual directory provides programmatic access to the Report Server Web service SOAP endpoints. It is also the root node of the report server folder namespace that provides URL addressing for reports, resources, models, shared data sources, and folders stored in the report server. By default, the virtual directory name for the report server is **reportserver**.

The virtual directories are part of the complete URL that is used to access Report Manager and the report server. A complete URL includes the following parts.

http:// or https://

The URL starts with a prefix. The https prefix is used if the server is configured for Secure Sockets Layer (SSL).

Http:// and https:// are the most common prefixes, but other prefixes are supported. For more information about URL construction, see **URL Access Syntax** in SQL Server Books Online.

A server name

This can be the computer name on the network. It can be **localhost** if you are accessing it locally. If the computer is accessible over an Internet connection, it must be a fully-qualified domain name.

A Web site

The Web site is an IIS construct that identifies which application receives the request.

This can be the default Web site. If you are using a custom Web site, it might be an IP address, a host header, or a port number. For more information about different ways to identify a Web site, see the [Hosting Multiple Web Sites on a Single Server](#) topic in the Internet Information Services (IIS) product documentation.

A virtual directory

The virtual directory is the only part of the URL that is configured through Reporting Services.

To create the virtual directory, you use the Reporting Services Configuration tool. If you are installing a default configuration, Setup will create the virtual directory for you.

When Setup creates the virtual directory, it uses the instance name if one exists. For example, if you install SQL Server 2005 Express Edition with Advanced Services in the default configuration, Setup will use `reportserver$SQLEXPRESS` for the virtual directory name.

Note:

URLs provide end-user or programmatic access to a report server. For more information about connections to a report server, see **Connections and Accounts in a Reporting Services Deployment** in SQL Server Books Online.

▣ Examples of Reporting Services URLs

The following list shows some examples of what a report server URL might look like:

- <http://localhost/reportserver>
- [http://localhost/reportserver\\$SQLEXPRESS](http://localhost/reportserver$SQLEXPRESS)
- <http://sales01/reportserver>
- <http://sales01:8080/reportserver>
- <https://sales.adventure-works.com/reportserver>
- <https://www.adventure-works.com:8080/reportserver01>

URLs that you use to access Report Manager share a similar format and are typically created under the same Web site that hosts the report server. The only difference is the virtual directory name (in this case, it is **reports** but you can configure it to use whatever name you want):

- <http://localhost/reports>
- [http://localhost/reports\\$SQLEXPRESS](http://localhost/reports$SQLEXPRESS)
- <http://sales01/reports>
- <http://sales01:8080/reports>
- <https://sales.adventure-works.com/reports>
- <https://www.adventure-works.com:8080/reports>

▣ Guidelines for Virtual Directory Configuration

Both the report server and Report Manager virtual directories must be created under an existing Web site. You can use the default Web site or a custom Web site. You can use the same Web site for both virtual directories. There are no special requirements on the Web site. All security, ASP.NET, paths, permissions, and other settings are configured on the virtual directories.

When you create the virtual directory, the Report Server Web service identity is configured automatically. The version of Internet Information Services (IIS) that you are using determines the Web service identity:

- In IIS 6.0, the application pool that contains the Web site determines the Web service identity. By default, it is **NetworkService**.
- In IIS 5.0, the Web service identity is always the ASP.NET process identity. By default, it is *computername\ASPNET*.

The Reporting Services virtual directories are configured to use ASP.NET 2.0 (this is the version that is installed with SQL Server). If you are running earlier versions of ASP.NET on your Web server, make sure that the different versions of ASP.NET do not conflict. The two major versions of the ASP.NET cannot run in the same process. In IIS 6.0, you can run different versions of ASP.NET side-by-side if you use separate application pools for each version. In IIS 5.0, each version of ASP.NET runs as a separate process automatically.

The report server virtual directories are configured for intranet access. Anonymous access is disabled by default. In most cases, report server URLs use network computer names. If you want to configure Reporting Services for Internet connections, you might need to modify the report server configuration files. For more information, see the section in this topic called "Configuration Settings for Specifying the URLs Used in Reporting Services" and [Configuring a Report Server for Internet Access](#).

To create and configure virtual directories, use the Reporting Services Configuration tool.

To run the report server and Report Manager under a custom Web site, see [How to: Configure Reporting Services to Use a Non-Default Web Site \(Reporting](#)

[Services Configuration](#)).

To specify an application pool for the Report Server Web service, use the Web Service Identity page in the Reporting Services Configuration tool.

▣ Custom Configuration in IIS

As with all virtual directories, you can further customize the report server and Report Manager virtual directories through Microsoft Internet Information Services (IIS). Further customization should only be required if you are deploying a custom security extension or if you are using a non-default Web site. If you customize the settings and break the deployment as a result, you can use the Reporting Services Configuration tool to reset the virtual directories to use the default configuration.

Using a Custom Port Number

To configure the report server or Report Manager to use a port other than port 80, you must edit the configuration files.

1. Open RSReportServer.config in a text editor.
2. Append the port number to the **UrlRoot** setting in the rsreportserver.config file. For example, if **Urlroot** is set to **http://sales01/report server**, set it to **http://sales01:8080/reportserver** instead.
3. Open RSWebApplication.config in a text editor.
4. Set **ReportServerUrl** to the same URL you specified in **UrlRoot**.
5. Delete the value (but not the tags) for **ReportServerVirtualDirectory**.
6. Save both files.

If you are using a custom port for a site that is configured for SSL, additional steps are required. For more information about how to use SSL on a port other than 443, see **Configuring a Report Server for Secure Sockets Layer (SSL) Connections**.

Redirection to the Report Manager Virtual Directory

Through IIS, you can use Report Manager as the default home page for the Web server. To configure redirection, use the following settings:

1. Open IIS Manager.
2. Right-click the Web site that hosts Report Manager and select **Properties**.
3. Click the **Home Directory** tab.

4. Click **A redirection to a URL**.
5. In **Redirect to:**, type **/reports**. Delete "http://" if it is in the text box.
6. Select **A directory below URL entered** if you are using IIS 6.0. If you are using IIS 5.0, the option to select is **A directory below this one**.
7. Click **OK**.

▣ Configuration Settings for Specifying the URLs Used in Reporting Services

Reporting Services configuration files include settings that have a URL as a value. The following table provides a concise summary of all the URLs that can be specified in a Reporting Services configuration file. You can use the descriptions to see how the URLs compare. To view an example of how some of these settings are used together, see **RSWebApplication Configuration File**.

Setting	Description
ReportServerVirtualDirectory	<p>Specifies the SOAP endpoint of the report server that is used by Report Manager. Valid values include the report server virtual directory name. For example, if the report server URL is http://adventure-works.com/reportserver, the value for the ReportServerVirtualDirectory setting is reportserver. Report Manager uses localhost to connect to the report server. If localhost is not enabled, use ReportServerUrl instead.</p> <p>This setting is defined in the RSWebApplication.config file. This value is created during Setup or through the Reporting Services Configuration tool when you configure the Report Manager virtual directory. You can also edit the configuration file to modify the value.</p> <p>This setting requires that Report Manager and the report server be configured to use the same Web site. If they use different Web sites, delete the value for ReportServerVirtualDirectory and use ReportServerUrl instead.</p>
	Provides an alternate way to specify the SOAP endpoint of the report server that is used by

ReportServerUrl

Report Manager. You can use this setting instead of **ReportServerVirtualDirectory** when Report Manager and the report server virtual directories are hosted on separate servers or Web sites.

To specify this setting, you must edit the `RSWebApplication.config` file manually. Valid values include `http://<server>/<reportserver>` or `https://<server>/<reportserver>` if you are using SSL certificates. If Report Manager is installed separately on an Internet-facing Web server, this value must be a fully qualified domain name that can be used to access a report server instance behind the firewall. This value can never be set to localhost.

If a Secure Sockets Layer (SSL) certificate is installed, this value must be the name of the server that is registered for that certificate. If you get the error "The underlying connection was closed: Could not establish trust relationship for the SSL/TLS secure channel", set **ReportServerUrl** to the fully-qualified domain name of the server for which the SSL certificate was issued. For example, if the certificate is registered to **https://adventure-works.com.onlinesales**, the report server URL would be **https://adventure-works.com.onlinesales/reportserver**.

Used by delivery extensions to compose the URL used to access items stored on the report server. For example, if you are using the Report Server E-mail delivery extension, the **UrlRoot** value is used to create a hyperlink to the report that is being distributed.

This value is also used to resolve links in a rendered report that is generated through

UrlRoot

unattended report processing.

This setting is specified in the RSReportServer.config file.

This value is specified during setup or through the Reporting Services Configuration tool if you installed the report server in a files-only installation. The value must be a valid URL address to a report server. If the report server will be accessed over the Internet, set it to a value that is accessible over an Internet connection.

▣ Change History

Release	History
12 December 2006	New content: <ul style="list-style-type: none">• About URLs and Virtual Directories for a Report Server in SharePoint Integrated Mode
17 July 2006	New content: <ul style="list-style-type: none">• Added examples of how to specify configuration settings. Changed content: <ul style="list-style-type: none">• ReportServerURL and ReportServerVirtualDirectory
14 April 2006	New content: <ul style="list-style-type: none">• Added examples of how to specify URLs for the report server and Report Manager.

▣ **See Also**

Tasks

[How to: Configure Reporting Services to Use a Non-Default Web Site \(Reporting Services Configuration\)](#)

Concepts

[Report Server Virtual Directory \(Reporting Services Configuration\)](#)

Other Resources

Configuring Reporting Services Components

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Configuring Service Accounts and Passwords in Reporting Services



[See Also](#)



[Configuring a SQL Server Installation](#) > [Reporting Services Configuration](#) >

Updated: **12 December 2006**

Reporting Services uses a Windows service and Web service to support server operations. This topic describes the default accounts used to run the services, how the accounts are configured at first, and how to specify a different account or password. For more information about each service, see [Administering the Report Server Web Service and Windows Service](#).

▣ Default Accounts and Initial Configuration

The Report Server Windows service can run under a built-in account or domain user account. The Windows service account is always configured during Setup. SQL Server Setup provides options for selecting a domain user account or the built-in Local System account when you specify options in the [Service Account](#) page. You must select one of these account types for Setup to continue.

📌 Note:

Setup does not provide options for specifying all possible service account choices. For example, it does not allow you to specify **NetworkService**. If you want to use **NetworkService**, you can use the Reporting Services Configuration tool to modify the service account properties after Setup is finished.

The Report Server Web service account is always the ASP.NET worker process identity. The account information for the Web service can be set during or after installation. It can be set during installation if you select the default configuration installation option. Otherwise, it is set after installation when you specify the report server virtual directory in the Reporting Services Configuration tool. Either way, the initial settings for the Report Server Web service are always the default values as determined by ASP.NET and the version of Microsoft Internet Information Services (IIS) that you are using:

- In IIS 6.0 on Microsoft Windows Server 2003, the ASP.NET worker process runs under the security identity of the application pool that contains it. You can have multiple ASP.NET worker processes, where each one is contained in a separate application pool that has its own security identity. By default, the security identity is **NetworkService**. **NetworkService** is the security identity of the default application pool, which provides settings that are inherited by any new application pool you subsequently create. When a report server is configured, the Report Server Web service is assigned to a dedicated application pool that is created for it when you specify the report server virtual directory. Because this application pool inherits the security identity of the default application pool, the account used to run the Report Server Web service is typically **NetworkService**.
- In IIS 5.0 on Windows 2000 Server or IIS 5.01 on Windows XP, there is

one ASP.NET worker process account for all ASP.NET applications that run on the computer. By default, ASP.NET runs under its own account as *computername\ASPNET*. To use a different account, you must configure ASP.NET to run under that account. There are no options in Reporting Services Configuration tool to set the ASP.NET account. You must modify the **<processModel>** element in the *Machine.config* file if you want to use a custom account for all ASP.NET applications that run on the server.

To view the account information for both services, use the Reporting Services Configuration tool. The tool includes the Web Service Identity page and the Windows Service Identity page that show service account information.

▣ **Changing the Service Accounts and Passwords for a SharePoint Integrated Report Server**

If you are running a report server in SharePoint integrated mode, and you subsequently change the accounts that each server runs under, you must update the service account information that is stored in the SharePoint configuration database. To do this, start Central Administration and re-select the report server instance in the **Manage integration settings** page. Repeating the integration steps will update the service account information accordingly. For more information, see [How to: Set Report Server Integration in SharePoint Central Administration](#).

▣ Changing the Service Accounts and Passwords

You can modify the service accounts, passwords, or both. For instructions on how to specify an account after you have decided which one to use, see [How to: Configure Service Accounts \(Reporting Services Configuration\)](#).

When you choose a new account, a login and database permissions will be created for the new account. Specifically, the account will be added to the **RSExecRole**. Accounts that were added previously are not removed from this role; you will need to remove accounts that are no longer in use. For more information, see [Administering a Report Server Database](#).

Choosing a Different Account

You can configure the Report Server Web service and Windows service accounts to run under non-default values. There is no single best approach for choosing an account type. Each account has advantages and disadvantages that you must consider. If you are deploying Reporting Services on a production server, best practices suggest that you configure the accounts to run under a user account that is used by a single service or application. The following guidelines and links in this section can help you decide on an approach that is best for your deployment.

Services	Online Resources
Windows service	Windows Service Identity (Reporting Services Configuration) Setting Up Windows Service Accounts in SQL Server Books Online. The Services and Service Accounts Security Planning Guide on MSDN.
Web service	Web Service Identity - Windows Server 2003 (Reporting Services Configuration)

Changing a Password before it Expires

To reset the password, use the Reporting Services Configuration tool and follow

these instructions: [How to: Configure Service Accounts \(Reporting Services Configuration\)](#).

 **Note:**

If the service account password for the Database Engine expires, the **rsReportServerDatabaseUnavailable** error occurs when you try to connect to the report server. Resetting the password resolves this error. To view the complete text of the error message, see **Troubleshooting Server and Database Connection Problems**.

Changing an Expired Password for the Report Server Windows Service

If the Report Server Windows service runs under a domain account and the password expires, the service will be unavailable until you specify a new password. To reset the password, click the **Start** menu, point to **Control Panel**, point to **Administrator Tools**, and click **Services**. Right-click **SQL Server Reporting Services**, select **Properties**, click **Log On**, and type the new password. After you update the password, start the Reporting Services Configuration tool and update the password in the Windows Service Identity page. This additional step is necessary to update the account information that is stored internally by the report server.

Dependencies on the Report Server Windows Service Identity

If you change the Report Server Windows service account, this can affect report server operations. For that reason, it is important to always use the Reporting Services Configuration tool when changing a service account. The Reporting Services Configuration tool performs the following additional steps to ensure the report server remains available:

- Automatically updates the encryption key to include the profile information of the new account. Because encryption is performed only by the Report Server Windows service, the keys must be updated when you reset the Windows service.

 **Note:**

If the report server is part of the scale-out deployment, only the report

server that you are updating is affected. The encryption keys for other report servers in the deployment are unaffected by the service account change.

- Automatically updates the login permissions on the SQL Server Database Engine instance used to host the report server database. If you are using the service accounts to connect to the database, Reporting Services granted SQL Server login permissions to the service accounts when you initially configured the connection. If you reset the Windows service account, the connection information must be updated.
- Automatically adds the new accounts to the report server group created on the local computer. This group is specified in the access control lists (ACLs) that secure Reporting Services files.

▣ Change History

Release	History
12 December 2006	Changed content: <ul style="list-style-type: none">• NetworkService security identity.• Changing service accounts and passwords.
14 April 2006	Changed content: <ul style="list-style-type: none">• Windows Server 2003 security identity information.

▣ **See Also**

Tasks

[How to: Configure Service Accounts \(Reporting Services Configuration\)](#)

[How to: Start Reporting Services Configuration](#)

Reference

[Web Service Identity - Windows Server 2003 \(Reporting Services Configuration\)](#)

[Windows Service Identity \(Reporting Services Configuration\)](#)

[Setting Up Windows Service Accounts](#)

Concepts

[Changing Passwords and User Accounts](#)

[Configuring Report Server Virtual Directories](#)

[Administering the Report Server Web Service and Windows Service](#)

[Starting and Stopping the Report Server Windows Service](#)

Other Resources

Deploying Reporting Services

Accounts in a Reporting Services Deployment

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Creating a Report Server Database



[See Also](#)



[Configuring a SQL Server Installation](#) > [Reporting Services Configuration](#) >

Updated: 12 December 2006

Reporting Services uses two SQL Server relational databases to store report server metadata and objects. One database is used for primary storage, and the second one stores temporary data. The databases are created together and bound by name. By default, the databases are named **reportserver** and **reportservertempdb**. Collectively, the two databases are referred to as the "report server database" or "report server catalog". You can use either SQL Server 2000 or SQL Server 2005 to host the databases.

Important:

Do not write applications that run queries against the report server database. The report server database is not a public schema. The table structure might change from one release to the next. If you write an application that requires access to the report server database, always use the Reporting Services APIs to access the report server database.

▣ **Server Modes and the Report Server Database**

When you create a report server database, you must know in advance whether the report server will be deployed in native mode or SharePoint integrated mode. The contents of the report server database will vary depending on the mode you use. Knowing how the report server will run will determine how you set options when you create the database.

Because native mode is the default, there is no option to select for this mode when you create the database. However, if you deploy a report server in SharePoint integrated mode, you must select the **Create the report server database in SharePoint integrated mode** option when creating the database.

If you do not select this option, the report server database will only support native mode report server operations. For more information, see [How to: Create a Report Server Database for SharePoint Integrated Mode \(Reporting Services Configuration\)](#).

▣ **Ways to Create the Report Server Database**

You can create the report server database in the following ways:

- Automatically through Setup, if you choose the default configuration installation option. In the SQL Server Installation Wizard, this is the **Install the default configuration** option in the Report Server Installation Options page. If you chose the **Install but do not configure** option, you must use the Reporting Services Configuration tool to create the database.
- Manually through the Reporting Services Configuration tool. For instructions, see [How to: Create a Report Server Database \(Reporting Services Configuration\)](#).

You can create a report server database on a local or remote SQL Server Database Engine instance. Creating the report server database on a remote computer requires that you configure the connection to use a domain user account or a service account that has network access. If you decide to use a remote SQL Server instance, consider carefully which credentials the report server should use to connect to the SQL Server instance. For more information, see [Configuring a Report Server Database Connection](#).

✍ **Important:**

Report Server and the SQL Server instance hosting the report server database can be in different domains. For Internet deployment, it is common practice to use a server that is behind a firewall. If you are configuring a report server for Internet access, use SQL Server credentials to connect to the instance of SQL Server that is behind the firewall and use IPSEC to secure the connection.

▣ **Creating Report Server Databases in Different Editions of SQL Server**

When creating a report server database, be aware that not all editions of SQL Server 2005 can be used to host the database. The following table shows you which editions of the Database Engine you can use for specific editions of Reporting Services.

For this edition of SQL Server 2005 Reporting Services	Use this edition of the Database Engine instance to host the database
SQL Server 2005 Express Edition with Advanced Services	SQL Server 2005 Express Edition with Advanced Services (local only).
Workgroup Edition	Workgroup Edition (local only)
Evaluation Edition	Standard, Evaluation, Enterprise Editions (local or remote)
Developer Edition	Standard, Developer, Enterprise Editions (local or remote)
Standard Edition	Standard, Enterprise Editions (local or remote)
Enterprise Edition	Standard, Enterprise Editions (local or remote)

▣ **Creating Report Server Databases on Upgraded Servers**

In some cases, you might encounter an exception when you try to create a report server database on an instance of SQL Server that has been upgraded from version 7.0. Upgrading does not set the database compatibility level for the master database. As a result, an exception occurs in the Reporting Services Configuration tool when you set up the database. For more information about how to create the database on an upgraded server, see [How to: Configure a Report Server Database on a Database Engine Upgraded from SQL Server 7.0 \(Reporting Services Configuration\)](#).

▣ Change History

Release	History
12 December 2006	New content: <ul style="list-style-type: none"><li data-bbox="592 499 1339 541">• Server Modes and the Report Server Database
14 April 2006	New content: <ul style="list-style-type: none"><li data-bbox="592 667 1339 751">• Creating Report Server Databases in Different Editions

▣ **See Also**

Tasks

[How to: Create a Report Server Database \(Reporting Services Configuration\)](#)

Concepts

[Deployment Modes for Reporting Services](#)

[Configuring Reporting Services for SharePoint 3.0 Integration](#)

[Administering a Report Server Database](#)

Other Resources

Report Server Database

Introducing Reporting Services Programming

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Configuring a Report Server Database Connection



[See Also](#)



[Configuring a SQL Server Installation](#) > [Reporting Services Configuration](#) >

Each report server instance requires a connection to the report server database that stores reports, report models, shared data sources, resources, and metadata managed by the server. The initial connection can be created during a report server installation if you are installing the default configuration. In most cases, you will use the Reporting Services Configuration tool to configure the connection after Setup is complete. You can modify the connection at any time to change the account type or reset credentials. For step-by-step instructions on how to create the database and configure the connection, see [How to: Create a Report Server Database \(Reporting Services Configuration\)](#).

You must configure a report server database connection in the following circumstances:

- Configuring a report server for first use.
- Configuring a report server to use a different report server database.
- Changing the user account or password that is used for the database connection. You only need to update the database connection when the account information is stored in the RSReportServer.config file. If you are using service accounts for the connection (which use Windows integrated security as the credential type), the password is not stored, eliminating the need to update the connection information. For more information about changing accounts, see [Configuring Service Accounts and Passwords in Reporting Services](#).
- Configuring a report server scale-out deployment. Configuring a scale-out deployment requires that you create multiple connections to a report server database. For more information about how to perform this multi-step operation, see [How to: Configure a Report Server Scale-Out Deployment \(Reporting Services Configuration\)](#).

Report server access to a report server database depends on credentials and connection information, and on encryption keys that are valid for the report

server instance that uses that database. Having valid encryption keys is necessary for storing and retrieving sensitive data. Encryption keys are created automatically when you configure the database for the first time. After the keys are created, you must update them if you change the Report Server Windows service identity. For more information about working with encryption keys, see [Managing Encryption Keys](#).

▣ Defining a Report Server Database Connection

To configure the connection, you must use the Reporting Services Configuration tool or the rsconfig command line utility. A report server requires the following connection information:

- Name of the SQL Server instance hosting the report server database. You can use a local or remote SQL Server 2000 or SQL Server 2005 Database Engine instance. If you are using SQL Server 2005 Express with Advanced Services, you must choose the local Database Engine instance.
- Name of the report server database. When creating a connection for the first time, you can create a new report server database or choose an existing database. For more information, see [Creating a Report Server Database](#).
- Credential type. You can use the service accounts, a Windows domain account, or a SQL Server database login.
- User name and password (required only if you are using Windows domain account or a SQL Server login).

The credentials that you provide must be granted access to the report server database. If you use the Reporting Services Configuration tool, this step is performed automatically. For more information about the permissions required to access the database, see the "Database Permissions" section in this topic.

Storing Database Connection Information

Reporting Services stores and encrypts the connection information in the following RSreportserver.config settings. You must use the Reporting Services Configuration tool or rsconfig utility to create encrypted values for these settings.

Not all of the values are set for every type of connection. If you configure the connection using the default values (that is, using the service accounts to make the connection), **<LogonUser>**, **<LogonDomain>**, and **<LogonCred>** will be empty, as follows:

```
<Dsn></Dsn>
```

 [Copy Code](#)

```
<ConnectionType></ConnectionType>  
<LogonUser></LogonUser>  
<LogonDomain></LogonDomain>  
<LogonCred></LogonCred>
```

If you configure the connection to use a specific Windows account or database login, you must remember to update the values that are stored if you subsequently change the account or login.

Choosing a Credential Type

There are three types of credentials that can be used in a connection to a report server database:

- Windows integrated security using the Report Server Web service and Windows service accounts.
- A Windows user account. If the report server and the report server database are installed on the same computer, you can use a local account. Otherwise, you must use a domain account.
- A SQL Server login.

Note:

A custom authentication extension cannot be used to connect to a report server database. Custom authentication extensions are used only to authenticate a principal to a report server. They have no effect on connections to the report server database or to external data sources that provide content to reports.

Using Service Accounts and Integrated Security

You can use Windows integrated security to connect through the Report Server Web service and Windows service accounts. Both service accounts are granted login rights to the report server database. This is the default credential type chosen by Setup if you install Reporting Services in the default configuration.

Service accounts are trusted accounts that provide a low-maintenance approach to managing a report server database connection. Because service accounts use Windows integrated security to make the connection, the credentials do not have to be stored. If you subsequently change the service account password without or

identity (for example, switching from a built-in account to a domain account), you must update the access rights for the account on the SQL Server instance that hosts the report server database. The Reporting Services Configuration tool will perform these steps for you. For more information, see [Configuring Service Accounts and Passwords in Reporting Services](#).

If you configure the database connection to use the service accounts, the accounts must have network permissions if the report server database is on a remote computer. Do not use the service account if the report server database is on a different domain, behind a firewall, or if you are using workgroup security instead of domain security. Use a SQL Server database user account instead.

Using a Domain User Account

You can specify that a single account is used to connect to the report server database. Both the Report Server Web service and Windows service will use this account to connect. If you use a local or domain account, you must update the report server database connection every time you change the password or the account. Always use the Reporting Services Configuration tool to update the connection.

Using a SQL Server Login

You can specify a single SQL Server login to connect to the report server database. If you use SQL Server authentication and the report server database is on a remote computer, use IPSEC to secure the transmission of data between the servers. If you use a SQL Server login, both the Report Server Web service and Windows service will use this account to connect. If you use a database login, you must update the report server database connection every time you change the password or the account.

Database Permissions

Accounts used to connect to the report server database are granted the following roles:

- **public** and **RSExecRole** roles for the **ReportServer** database.

- **RSExecRole** role for the **master**, **msdb**, and **ReportServerTempDB** databases.

When you use the Reporting Services Configuration tool to create or modify the connection, these permissions are granted automatically. If you use the rsconfig utility, and you are specifying a different account for the connection, you must update the SQL Server login for that new account. You can create script files in the Reporting Services Configuration tool that will update the SQL Server login for the report server.

Verifying the Database Name

Use the Reporting Services Configuration tool to find out which report server database is used by a particular report server instance. To find the name, connect to the report server instance and open the Database Setup page.

▣ **Using a Different Report Server Database or Moving a Report Server Database**

You can configure a report server instance to use a different report server database by changing the connection information. A common case for switching databases is when you deploy a production report server. Switching from a test report server database to a production report server database is typically how production servers are rolled out. You can also move a report server database to another computer. For more information, see **Moving a Report Server Database to Another Computer** and [Migrating Reporting Services](#).

▣ **Configuring Multiple Reports Servers to Use the Same Report Server Database**

You can configure multiple report servers to use the same report server database. Typically, this deployment configuration is considered a scale-out deployment model that you implement when you want to run multiple report servers in a server cluster. However, you can also use this configuration to test the installation and settings of a new report server instance to compare it against an existing report server that works the way you want it to. For more information, see [Configuring a Report Server Scale-Out Deployment](#).

▣ **See Also**

Concepts

[Creating a Report Server Database](#)

[Reporting Services Configuration How-to Topics](#)

[Configuring Service Accounts and Passwords in Reporting Services](#)

Other Resources

Connecting to a Data Source

Configuring Reporting Services Components

RSReportServer Configuration File

rsconfig Utility

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Configuring an Account for Unattended Report Processing



[See Also](#)



[Configuring a SQL Server Installation](#) > [Reporting Services Configuration](#) >

Updated: 12 December 2006

SQL Server 2005 Reporting Services provides a special account that is used for unattended report processing and for sending connection requests across the network. The account is used in the following ways:

- Send connection requests over the network for reports that use database authentication. For more information, see **Specifying Credential and Connection Information** in SQL Server Books Online.
- Retrieve external image files that are used in report. If you want to use an image file and the file cannot be accessed through Anonymous access, you can configure the unattended report processing account and grant the account permission to access the file. For more information about the unattended execution account is used for retrieving images, see **Adding an Image to a Report** in SQL Server Books Online.
- Connect to external report data sources that do not require or use authentication. Reports that do not use credentials to access the external data source are those reports that specify the **Credentials are not required** option in the Data Source property page.

Unattended report processing refers to any report execution process that is triggered by an event (either a schedule-driven event or data refresh event) rather than a user request. The report server uses the unattended report processing account to log on to the computer that hosts the external data source. This account is necessary because the credentials of the Report Server Windows service account or Web service account are never used to connect to other computers.

Important:

Configuring the account is optional. However, if you do not configure it, you will limit your options for connecting to some data sources, and you

might not be able to retrieve image files from remote computers. If you do configure the account, you must keep it up to date. Specifically, if you allow a password to expire or the account information is changed in Active Directory, you will encounter the following error the next time a report is processed: "Logon failed (rsLogonFailed) Logon failure: unknown user name or bad password." Proper maintenance of the unattended report processing account is essential, even if you never retrieve external images or send connection requests to external computers. If you configure the account but then find that you are not using it, you can delete it to avoid routine account maintenance tasks.

To configure the account or update account information, run the Reporting Services Configuration tool or **rsconfig -e** to set values for the following settings in the RSreportserver.config file on a local or remote report server instance:

```
<UnattendedExecutionAccount>
  <UserName></UserName>
  <Password></Password>
  <Domain></Domain>
</UnattendedExecutionAccount>
```

 [Copy Code](#)

The values are encrypted. Once you set the values, you cannot decrypt them to view the values in plain text. If you mistype the values or forget the values you specified, you must use the Reporting Services Configuration tool or run **rsconfig -e** to start over.

▣ **How to Configure the Account**

The account must be a domain user account. To server its intended purpose, this account should be different than the one used to run the Report Server Windows service or Web service. Be sure to use an account that has minimum permissions (read-only access is sufficient) and limited access to just those computers that provide data sources and resources to the report server.

To create the account, you can use the Reporting Services Configuration tool or the **rsconfig** utility. The easiest way to configure the unattended execution account is to run the Reporting Services Configuration tool and specify credentials in the Execution Account page. For more information, see [How to: Start Reporting Services Configuration](#).

Configuring the Unattended Report Processing Account.

1. Start the Reporting Services Configuration tool and connect to the report server instance you want to configure.
2. On the Execution Account page, select **Specify an execution account**.
3. Type the account and password, retype the password, and then click **Apply**.

Maintaining the Unattended Report Processing Account

Once you define the account, you must ensure that the account and password are kept up to date. You can use the Reporting Services Configuration tool to update the configuration settings that store information about this account.

1. Start the Reporting Services Configuration tool and connect to the report server instance you want to configure.
2. On the Execution Account page, verify that **Specify an execution account** is selected.
3. Type the new account or password, retype the password, and then click **Apply**.

Deleting the Unattended Report Processing Account

If you are not using the account, you can delete it to avoid routine account maintenance tasks.

1. Start the Reporting Services Configuration tool and connect to the report server instance you want to configure.
2. On the Execution Account page, clear **Specify an execution account**.
3. Click **Apply**.

The account information is removed from the RSReportServer.config file.

Using the RSConfig Utility to Configure and Maintain the Unattended Reporting Processing Account

You can also use the **rsconfig** utility. To specify the account, use the **-e** argument of **rsconfig**. Specifying the **-e** argument for **rsconfig** directs the utility to write the account information to the configuration file. You do not need to specify a path to RSreportserver.config. Follow these steps to configure the account.

1. Create or select a domain account that has access to computers and servers that provide data or services to a report server. You should use an account that has reduced permissions (for example, read-only permissions).
2. Open a command prompt: On the **Start** menu, click **Run**, type **cmd**, and then click **OK**.
3. Type the following command to configure the account on a local report server instance:
rsconfig -e -u < domain / username > -p < password >

rsconfig -e supports additional arguments. For more information about syntax and to view command examples, see **rsconfig Utility** in SQL Server Books Online.

Change History

Release	History
	New content: <ul style="list-style-type: none">• Important restrictions regarding the unattended

12 December
2006

account.

Changed content:

- How to configure the account.

▣ **See Also**

Other Resources

Accounts in a Reporting Services Deployment

Configuring Reporting Services Components

Data Sources Properties Page

RSReportServer Configuration File

rsconfig Utility

Specifying Credential and Connection Information

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Configuring a Report Server for E-Mail Delivery



[See Also](#)



[Configuring a SQL Server Installation](#) > [Reporting Services Configuration](#) >

Updated: **12 December 2006**

SQL Server 2005 Reporting Services includes an e-mail delivery extension so that you can distribute reports through e-mail.

The report server e-mail delivery extension is not configured by default. You must use the Reporting Services Configuration tool to minimally configure the extension. To set advanced properties, you must edit the RSReportServer.config file. If you cannot configure the report server to use this extension, you can deliver reports to a shared folder instead. For more information, see **File Share Delivery in Reporting Services**.

▣ Configuration Requirements

- Report server e-mail delivery is implemented on Collaboration Data Objects (CDO) and requires a local or remote Simple Mail Transport Protocol (SMTP) server or SMTP forwarder.
- The Report Server Windows service account must have permission on the SMTP server to send mail.
- The Report Server E-mail delivery extension uses UTF-8 encoding in e-mail attachments. You cannot modify the encoding; the HTML rendering extension only supports UTF-8.

📌 Note:

The default e-mail delivery extension does not provide support for digitally signing or encrypting outgoing mail messages.

▣ **Configuring a Report Server for Local or Remote SMTP Service**

You can use a local SMTP service or a remote SMTP server or forwarder to support e-mail delivery. If you have access to an existing remote SMTP server, you should consider using it. If there is no SMTP server available or if you subsequently encounter report delivery errors that can be attributed to computer connection failures, you should switch to using a local SMTP service. Details about how to configure a report server for local or remote service are provided further on in this topic.

▣ Setting Configuration Options for E-Mail Delivery

Before you can use Report Server e-mail delivery, you must set configuration values that provide information about which SMTP server to use.

To configure a report server for e-mail delivery, do the following:

- Use the Reporting Services Configuration tool if you are specifying just an SMTP server and a user account that has permission to send e-mail. These are the minimum settings that are required for configuring the Report Server e-mail delivery extension.
- (Optionally) Use a text editor to specify additional settings in the RSreportserver.config and RSwebapplication.config files. Together, these two configuration files include all of the configuration settings for Report Server e-mail delivery. Specifying additional settings in these files is required if you are using a local SMTP server or if you are restricting e-mail delivery to specific hosts. For more information about finding and modifying configuration files, see **Reporting Services Configuration Files** in SQL Server Books Online.

The following table describes the e-mail delivery settings in each file:

File	Description
RSwebapplication.config	Specifies the default delivery extension for all subscriptions that are created and managed through Report Manager. The DefaultDeliveryExtension setting in RSwebapplication.config specifies that the default delivery extension is report server e-mail.
RSreportserver.config	Stores all settings that relate to the SMTP service.

 **Note:**

Report server e-mail settings are based on CDO. If you want more detail about specific settings, you can refer to the CDO production documentation.

▣ Example Report Server E-Mail Configuration

The following example illustrates the settings in the RSreportserver.config file for a remote SMTP server. To read about the setting descriptions and valid values, see **RSReportServer Configuration File** or the CDO product documentation.

 [Copy Code](#)

```
<RSEmailDPConfiguration>
  <SMTPServer>mySMTPServer.Adventure-Works.com</SMTPServer>
  <SMTPServerPort></SMTPServerPort>
  <SMTPAccountName></SMTPAccountName>
  <SMTPConnectionTimeout></SMTPConnectionTimeout>
  <SMTPServerPickupDirectory></SMTPServerPickupDirectory>
  <SMTPUseSSL></SMTPUseSSL>
  <SendUsing>2</SendUsing>
  <SMTPAuthenticate></SMTPAuthenticate>
  <From>my-rs-email-account@Adventure-Works.com</From>
  <EmbeddedRenderFormats>
    <RenderingExtension>MHTML</RenderingExtension>
  </EmbeddedRenderFormats>
  <PrivilegedUserRenderFormats></PrivilegedUserRenderFormats>
  <ExcludedRenderFormats>
    <RenderingExtension>HTMLLOWC</RenderingExtension>
    <RenderingExtension>NULL</RenderingExtension>
  </ExcludedRenderFormats>
  <SendEmailToUserAlias>True</SendEmailToUserAlias>
  <DefaultHostName></DefaultHostName>
  <PermittedHosts>
    <HostName>Adventure-Works.com</HostName>
    <HostName>hotmail.com</HostName>
  </PermittedHosts>
</RSEmailDPConfiguration>
```

▣ **Configuration Options for Setting the To: Field in a Message**

User-defined subscriptions that are created according to the permissions granted by the **Manage individual subscriptions** task contain a pre-set user name that is based on the domain user account. When the user creates the subscription, the recipient name in the **To:** field is self-addressed using the domain user account of the person creating the subscription.

If you are using an SMTP server or forwarder that uses e-mail accounts that are different from the domain user account, the report delivery will fail when the SMTP server tries to deliver the report to that user.

To workaroud this issue, you can modify configuration settings that allow users to enter a name in the **To:** field:

1. Open RSReportServer.config with a text editor.
2. Set **SendEmailToUserAlias** to **False**.
3. Set **DefaultHostName** to the Domain Name System (DNS) name or IP address of the SMTP server or forwarder.
4. Save the file.

▣ Configuration Options for Remote SMTP Service

The connection between the report server and an SMTP server or forwarder is determined by the following configuration settings:

- **SendUsing** specifies a method for sending messages. You can choose between a network SMTP service or a local SMTP service pickup directory. To use a remote SMTP service, this value must be set to 2 in the `RSReportServer.config` file.
- **SMTPServer** specifies the remote SMTP server or forwarder. This value is a required value if you are using a remote SMTP server or forwarder.
- **From** sets the value that appears in the **From:** line of an e-mail message. This value is a required value if you are using a remote SMTP server or forwarder.

Other values that are used for remote SMTP service include the following (note that you do not need to specify unless you want to override the default values).

- **SMTPServerPort** is configured for port 25.
- **SMTPAuthenticate** specifies how the report server connects to the remote SMTP server. The default value is 0 (or no authentication). In this case, the connection is made through Anonymous access. Depending on your domain configuration, the report server and the SMTP server may need to be members of the same domain.
To send e-mail to restricted distribution lists (for example, distribution lists that accept incoming messages only from authenticated accounts), set **SMTPAuthenticate** to 2.

▣ Configuration Options for Local SMTP Service

Configuring a local SMTP service is useful if you are testing or troubleshooting report server e-mail delivery. The local SMTP service is not enabled by default. For instructions on how to enable it, see [How to: Configure a Report Server for E-mail Delivery \(Reporting Services Configuration\)](#).

The connection between the report server and a local SMTP server or forwarder is determined by the following configuration settings:

- **SendUsing** is set to **1**.
- **SMTPServerPickupDirectory** is set to a folder on the local drive.
 - ▣ **Note:**
Be sure that you do not set **SMTPServer** if you are using a local SMTP server.
- **From** sets the value that appears in the **From:** line of an e-mail message. This value is required.

Change History

Release	History
12 December 2006	Changed content: <ul style="list-style-type: none">• Configuration options for local SMTP service.
17 July 2006	New content: <ul style="list-style-type: none">• Configuration Requirements• Configuration Options for Setting the To: Field in a Message

▣ **See Also**

Tasks

[How to: Start Reporting Services Configuration](#)

[How to: Configure a Report Server for E-mail Delivery \(Reporting Services Configuration\)](#)

Other Resources

Distributing Reports Through Subscriptions

E-Mail Delivery in Reporting Services

Reporting Services Configuration Files

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Administering a Report Server Database



[See Also](#)



[Configuring a SQL Server Installation](#) > [Reporting Services Configuration](#) >

Updated: 12 December 2006

A Reporting Services deployment uses two SQL Server relational databases for internal storage. By default, the databases are named **ReportServer** and **ReportServerTempdb**. **ReportServerTempdb** is created with the primary report server database and is used to store temporary data, session information, and cached reports.

In Reporting Services, database administration tasks include backing up and restoring the report server databases and managing the encryption keys that are used to encrypt and decrypt sensitive data.

To administer the report server databases, SQL Server provides a variety of tools.

- To back up or restore the report server database, move a report server database, or recover a report server database, you can use SQL Server Management Studio, the Transact-SQL commands, or the database command prompt utilities. For instructions, see **Moving a Report Server Database to Another Computer** in SQL Server Books Online.
- To copy existing database content to another report server database, you can attach a copy of a report server database and use it with a different report server instance. Or, you can create and run a script that uses SOAP calls to recreate report server content in a new database. You can use the **rs** utility to run the script.
- To manage connections between the report server and report server database, and to find out which database is used for a particular report server instance, you can use Database Setup page in the Reporting Services Configuration tool. To learn more about the report server connection to the report server database, see [Configuring a Report Server Database Connection](#).

- To gather metrics about report server performance and activity, see **Querying and Reporting on Report Execution Log Data** in SQL Server Books Online.

▣ SQL Server Login and Database Permissions

The report server databases are used internally by the report server. Connections to either database are made by the Report Server Web service and Report Server Windows service.

You can use the Reporting Services Configuration tool to specify the connection. You can use the credentials of the service accounts, or the credentials of domain user account, or a SQL Server login. The account you choose for the connection must have a SQL Server login and must have the **Public** and **RSExecRole** roles for the report server databases.

The login and permissions are created for you automatically when you use the Reporting Services Configuration tool to configure the database connection, upgrade the report server database, or run equivalent scripts on the SQL Server instance that hosts the database.

The **RSExecRole** provides permissions for accessing the database tables and for executing stored procedures. The **RSExecRole** is created in **master** and **msdb** when you create the report server database. Starting in SQL Server 2005 Service Pack 1, the **RSExecRole** is a member of the **db_owner** role for the report server databases. The additional permissions allow the report server to update the schema in subsequent releases, thus simplifying the upgrade process.

▣ **Removing Unused Accounts in RSExecRole**

If you configure the Report Server Web service or Windows service to run under a different account, the new account will be added to the **RSExecRole** automatically. However, any accounts that were previously added and that are no longer used must be removed manually. To remove the accounts, use Management Studio to do the following:

1. Connect to the Database Engine instance that hosts the report server database.
2. Expand the **Databases** folder, expand the **ReportServer** folder, expand the **Security** folder, expand **Roles**, expand **Database Roles**, and then double-click **RSExecRole**.
3. Select the account that is no longer used.
4. Click **Remove**.

▣ Naming Conventions for the Report Server Databases

When creating the primary database, the name of the database must follow the rules specified for **Identifiers**. The temporary database name always uses the same name as the primary report server database but with a **Tempdb** suffix. You cannot choose a different name for the temporary database.

Renaming a report server database is not supported because the report server databases are considered internal components. Renaming the report server databases causes errors to occur. Specifically, if you rename the primary database, an error message explains that the database names are out of sync. If you rename the **ReportServerTempdb** database, the following internal error occurs later when you run reports:

"An internal error occurred on the report server. See the error log for more details. (rsInternalError)

Invalid object name 'ReportServerTempDB.dbo.PersistedStream'."

This error occurs because the **ReportServerTempdb** name is stored internally and used by stored procedures to perform internal operations. Renaming the temporary database will prevent the stored procedures from working properly.

▣ **About Database Versions**

In Reporting Services, explicit information about the database version is not available. However, because database versions are always synchronized to product versions, you can use product version information to tell when the database version has changed. Product version information for Reporting Services is indicated through file version information that appears in the log files, in the headers of all SOAP calls, and when you connect to the report server URL (for example, when you open a browser to <http://localhost/reportserver>).

▣ Change History

Release	History
12 December 2006	New content: <ul style="list-style-type: none"><li data-bbox="527 499 1250 541">• Removing Unused Accounts in RSExecRole
14 April 2006	New content: <ul style="list-style-type: none"><li data-bbox="527 667 1433 751">• Naming Conventions and versioning information for the Report Server Databases

▣ **See Also**

Concepts

[Creating a Report Server Database](#)

[Managing Encryption Keys](#)

Other Resources

**Backup and Restore Operations for a Reporting Services Installation
Report Server Database**

Deploying Reporting Services

Administering Reporting Services

Report Server Database Requirements

Storing Encrypted Data in a Report Server Database

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Administering the Report Server Web Service and Windows Service



[See Also](#)



[Configuring a SQL Server Installation](#) > [Reporting Services Configuration](#) >

A report server runs as a Windows service and as a Web service. The services work together and support different aspects of report server functionality:

- The Report Server Windows service performs initialization, reversible encryption, database maintenance tasks, and all scheduling and delivery. The service runs in the background. It performs end-to-end processing for reports that run on a schedule (specifically, it creates report snapshots and runs subscription reports).
Because it performs all encryption operations, the Report Server Windows service must be running whenever you specify or use encrypted values. Specifying stored credentials, running a report that uses stored credentials, and publishing a report to a report server (data source information is encrypted) are all operations that require the Report Server Windows service.
- The Report Server Web service performs end-to-end processing for reports that run on demand. It also provides the primary programmatic interface for applications that integrate with a report server. Report Manager, Report Builder, and SQL Server Management Studio are examples of applications that require the Report Server Web service.

In most cases, you will want to run the services together so that you can use all of the functionality provided in Reporting Services. However, if the deployment model you are implementing has very narrow requirements, you can run just the Report Server Windows service if all report processing is configured as scheduled operations. Similarly, you can run just the Report Server Web service if you only want interactive, on-demand reporting.

To make either service unavailable, run the SQL Server Surface Area Configuration Tool and select the **Surface Area Configuration for Features** option. You cannot turn off the Report Server Windows service completely; it

provides initialization and encryption functionality that is required for server operations. However, you can turn off scheduled and event processing.

The Report Server Windows service is registered and configured during Setup. It runs under whatever account you specify. We recommend a least-privilege domain user account that has permission to logon to the network or **NetworkService**, but you can use a local account if the report server database runs on the same computer as the report server. The Report Server Web service runs under the ASP.NET identity by default. If you modify service account settings, be sure to use the Reporting Services Configuration tool so that dependent settings are updated with the new values. For more information about account configuration, see **Connections and Accounts in a Reporting Services Deployment** and [Configuring Service Accounts and Passwords in Reporting Services](#).

▣ **SQL Server Agent Service Requirements**

The Report Server Windows service requires the SQL Server Agent service. SQL Server Agent service must run under a domain account if you configured the report server to connect to SQL Server using a domain account and Windows authentication (as opposed to a SQL Server login or Service Account). When the report server runs as a domain user, the report server creates SQL Server Agent jobs that are owned by that domain account. Before SQL Server Agent can route a task to the Scheduling and Delivery Processor, SQL Server Agent must have permission to access job information for jobs owned by a domain account. If SQL Server Agent happens to run as a local user account, the service will not have permission to access domain account information, and report subscription and delivery will subsequently fail.

▣ See Also

Tasks

[How to: Configure Service Accounts \(Reporting Services Configuration\)](#)

[How to: Start Reporting Services Configuration](#)

Concepts

[Configuring Service Accounts and Passwords in Reporting Services](#)

[Starting and Stopping the Report Server Windows Service](#)

[Initializing a Report Server](#)

[Managing Encryption Keys](#)

Other Resources

Scheduling and Delivery Processor

Specifying Credential and Connection Information

Deploying Reporting Services

Administering Reporting Services

Configuring SQL Server Agent

About SQL Server Agent

SQL Server Agent

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Starting and Stopping the Report Server Windows Service



[See Also](#)



[Configuring a SQL Server Installation](#) > [Reporting Services Configuration](#) >

You can stop the Report Server Windows service to stop many report server operations. When you stop the Report Server Windows service, the following results occur:

- All report processing that is triggered from scheduled events (for example, subscriptions, report snapshot refreshes, and cache expiration) stops.
- Encryption and decryption operations are suspended. These operations include initializing servers and accessing encrypted credentials and connection strings.
- Access to the Remote Procedure Calls (RPC) endpoint used by the Reporting Services Configuration tool, **rskeymgmt**, and the report server Windows Management Instrumentation (WMI) provider is disabled.
- Report server database maintenance and cleanup operations stop.

Stopping the Report Server Windows service does not affect the Web service. Web-based access to the report server and the SOAP endpoint continue to be available while the Windows service is stopped. If you want to stop the Web service, you can use Internet Information Services (IIS) Manager to stop the Web site that hosts the report server.

You can use a variety of tools to start or stop the Report Server Windows service, including the Reporting Services Configuration tool, SQL Server Configuration Manager, SQL Server Surface Area Configuration tool, and the Services tool provided in Microsoft Windows.

If you are doing more than starting or stopping the service, such as changing the service account, you must use the Reporting Services Configuration tool. Using other tools to change the service account can break your Reporting Services installation. For more information, see [Configuring Service Accounts and Passwords in Reporting Services](#).

You cannot pause and resume the service. There are no start parameters. Although there are no explicit dependencies, SQL Server Agent must be running if you support any scheduled operations on the server.

▣ **Using the Reporting Services Configuration Tool**

- To stop or start the service, run Reporting Services Configuration tool and connect to the report server. On the Report Server Status page, click **Stop** or **Start**.

▣ **Using Services in Administrative Tools**

- To stop or start the service, open Administrative Tools, open Services, right-click **SQL Server Reporting Services (MSSQLSERVER)**, and click **Stop** or **Restart**. If you are running multiple instance or if the report server is running as a named instance, verify that the instance name in parentheses corresponds to the report server instance you want to stop or restart.

▣ **Using the SQL Server Configuration Manager**

- To stop or start the service, run SQL Server Configuration Manager, select SQL Server 2005 Services, right-click **Report Server**, and click **Stop** or **Restart**.

▣ **Using the SQL Server Surface Area Configuration Tool**

The SQL Server Surface Area Configuration tool provides a means of starting and stopping the service as a convenience so that you can stop the service before you make it unavailable in the Reporting Services installation.

You can also use this tool to make specific features unavailable. You can disable the Scheduling and Delivery processor and Web service requests. Enabling and disabling features affects configuration settings in the RSReportServer.config file. Specifically, if you select the IsWindowsServiceEnabled option, you set the following values in RSReportServer.config to **true**: IsSchedulingService, IsNotificationService, IsEventService. The IsWindowsServiceEnabled option in the SQL Server Surface Area Configuration tool is a union of all three configuration settings. Manually editing the values in the configuration file can undermine or reverse the selection you made in the tool.

- To stop or start the service, run SQL Server Surface Area Configuration tool, click **Surface Area Configuration for Services and Connections**, expand Reporting Services, and click **Stop** or **Start**.

▣ **About the Report Server Web Service**

Starting, pausing, resuming, and stopping a service is not generally part of Web service management. ASP.NET starts and stops the Web service as part of managing the service. In most cases, you do not have to start or stop the Web service yourself. If you want to restart the Report Server Web service, you must restart Microsoft Internet Information Services (IIS). If you are running IIS 6.0, you can stop and start the application pool.

To verify whether the Web service is running, you can ping the service by typing **http://localhost/reportserver** in a browser URL address.

▣ **See Also**

Concepts

[Starting and Stopping the Report Server Windows Service](#)

[Administering the Report Server Web Service and Windows Service](#)

Other Resources

Scheduling and Delivery Processor

Deploying Reporting Services

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Managing Encryption Keys

 [See Also](#)



[Configuring a SQL Server Installation](#) > [Reporting Services Configuration](#) >

Reporting Services uses encryption keys to secure credentials and connection information that is stored in a report server database. In Reporting Services, encryption keys include a combination of public, private, and symmetric keys that are used to protect sensitive data. The symmetric key is created during report server initialization when you install or configure the report server, and it is used by the report server to encrypt sensitive data that is stored in the report server. Public and private keys are created by the operating system, and they are used to protect the symmetric key. A public and private key pair is created for each report server instance that stores sensitive data in a report server database.

Managing the encryption keys consists of creating a backup copy, and knowing when and how to restore, delete, or change the keys. If you migrate a report server installation or configure a scale-out deployment, you must have a backup copy of the symmetric key so that you can apply it to the new installation.

To manage symmetric keys, you can use the Reporting Services Configuration tool or the **rskeymgmt** utility. The tools included in Reporting Services are used to manage the symmetric key only (the public and private keys are managed by the operating system). Both the Reporting Services Configuration tool and the **rskeymgmt** utility support the following tasks:

- Back up a copy of the symmetric key so that you can use it to recover a report server installation or as part of a planned migration.
- Restore a previously saved symmetric key to a report server database, allowing a new report server instance to access existing data that it did not originally encrypt.
- Delete the encrypted data in a report server database in the unlikely event that you can no longer access encrypted data.
- Re-create symmetric keys and re-encrypt data in the unlikely event that the symmetric key is compromised. As a security best practice, you should recreate the symmetric key periodically (for example, every few

months) to protect the report server database from cyber attacks that attempt to decipher the key.

- Add or remove a report server instance from a report server scale-out deployment where multiple report servers share both a single report server database and the symmetric key that provides reversible encryption for that database.

▣ In This Section

Topic	Description
Initializing a Report Server	Explains how encryption keys are created.
Backing Up and Restoring Encryption Keys	Explains how to back up encryption keys and restore them to recover or migrate a report server installation.
Deleting and Re-creating Encryption Keys	Explains how you can replace a symmetric key with a new version, and how to start over if symmetric keys cannot be validated.
Adding and Removing Encryption Keys for Scale-Out Deployment	Explains how to add and remove encryption keys to control which report servers are part of a scale-out deployment.

▣ **See Also**

Concepts

[Administering a Report Server Database](#)

Other Resources

Storing Encrypted Data in a Report Server Database
Reporting Services Configuration Tool
rskeymgmt Utility

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Initializing a Report Server



[See Also](#)



[Configuring a SQL Server Installation](#) > [Reporting Services Configuration](#) > [Managing Encryption Keys](#) >

In Reporting Services, an initialized server is one that can encrypt and decrypt data in a report server database. Initialization is a requirement for report server operation. Initialization occurs when the Report Server Windows service is started for the first time. It also occurs when you join the report server to the existing deployment, or when you manually recreate the keys as part of the recovery process. For more information about how and why encryption keys are used, see [Managing Encryption Keys](#) and **Storing Encrypted Data in a Report Server Database**.

Encryption keys are based partly on the profile information of the Report Server Windows service. If you change the user identity used to run the Report Server Windows service, you must update the keys accordingly. If you are using the Reporting Services Configuration tool to change the identity, this step is handled for you automatically.

If initialization fails for some reason, the report server returns an **RSReportServerNotActivated** error in response to user and service requests. In this case, you may need to troubleshoot the system or server configuration. For more information, see **Troubleshooting Initialization and Encryption Key Errors**.

▣ Overview of the Initialization Process

The initialization process creates and stores a symmetric key used for encryption. The symmetric key is created by the Microsoft Windows Cryptographic Services and subsequently used by the Report Server Windows service to encrypt and decrypt data. The symmetric key is itself encrypted with an asymmetric key.

In a Reporting Services installation, only the Report Server Windows service can perform encryption operations. When the Report Server Web service requires encryption or decryption, it calls the Report Server Windows service to perform the operation.

The following steps describe the initialization process:

1. At initial start up, the Report Server Windows service reads the RSReportServer.config file to get the installation identifier and database connection information.
2. The Report Server Windows service requests a public key from Cryptographic Services. Windows creates a private and public key and sends the public key to the Report Server Windows service.
3. The Report Server Windows service connects to the report server database and stores the installation identifier and public key values.
4. The Report Server Windows service calls into Cryptographic Services again, this time to request a symmetric key. Windows creates the symmetric key.
5. The Report Server Windows service connects to the report server database again, and adds the symmetric key to the public key and installation identifier values that were stored in step 3. Before storing it, the Report Server Windows service uses its public key to encrypt the symmetric key. Once the symmetric key is stored, the report server is considered initialized and available to use.

▣ **Initializing a Report Server for Scale-out Deployment**

Reporting Services supports a scale-out deployment model that shares a single report server database among multiple report server instances. To join a scale-out deployment, a report server must create and store its copy of the symmetric key in the shared database. Although a single symmetric key is used by servers that use the database, each report server has its copy of the key. Each copy varies in that it is uniquely encrypted using the public key its owner.

The first set of steps for initializing a report server for scale-out deployment are identical to the first three steps that describe initialization for a single server and database combination.

The initialization process for a scale out deployment differs in how the report server gets the symmetric key. When the first server is initialized, it gets the symmetric key from Windows. When the second server is initialized during configuration for scale-out deployment, it gets the symmetric key from the Report Server Windows service that is already initialized. The first report server instance uses the public key of the second instance to create an encrypted copy of the symmetric key for the second report server instance. The symmetric key is never exposed as plain text at any point in this process.

▣ **How to Initialize a Report Server**

- To initialize a report server, use the Reporting Services Configuration tool. Initialization occurs automatically when you create and configure the report server database. For more information, see [Configuring a Report Server Database Connection](#).
- To initialize a report server for scale-out deployment, you can use the Initialization page in the Reporting Services Configuration tool or the **RSKeymgmt** utility. To follow step-by-step instructions, see [How to: Configure a Report Server Scale-Out Deployment \(Reporting Services Configuration\)](#).

📌 **Note:**

RSKeymgmt is a console application that you run from a command line on a computer that hosts a report server instance that is already part of a scale-out deployment. When you run the utility, you specify arguments to select a remote report server instance that you want to initialize.

A report server will be initialized only if there is a match between the installation identifier and the public key. If the match succeeds, a symmetric key is created that permits reversible encryption. If the match fails, the report server is disabled, in which case you may be required to apply a backup key or delete the encrypted data if a backup key is unavailable or not valid. For more information about encryption keys used by a report server, see [Managing Encryption Keys](#).

📌 **Note:**

You can also use the Reporting Services Windows Management Instrumentation (WMI) provider to initialize a report server programmatically. For more information, see **Reporting Services WMI Provider**.

▣ **How to Confirm a Report Server Initialization**

To confirm report server initialization, ping the Report Server Web service by typing **http://<servername>/reportserver** in the command window. If the **RSReportServerNotActivated** error occurs, the initialization failed.

▣ **See Also**

Other Resources

Backup and Restore Operations for a Reporting Services Installation

Troubleshooting Initialization and Encryption Key Errors

Deploying Reporting Services

Administering Reporting Services

Storing Encrypted Data in a Report Server Database

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Backing Up and Restoring Encryption Keys



[See Also](#)



[Configuring a SQL Server Installation](#) > [Reporting Services Configuration](#) > [Managing Encryption Keys](#) >

An important part of report server configuration is creating a backup copy of the symmetric key used for encrypting sensitive information. A backup copy of the key is required for many routine operations, and enables you to reuse an existing report server database in a new installation. It is necessary to restore the backup copy of the encryption key when any of the following events occur:

- Changing the Report Server Windows service account name or resetting the password. When you use the Reporting Services Configuration tool, backing up the key is part of a service account name change operation.

 **Note:**

Resetting the password is not the same as changing the password. A password reset requires permission to overwrite account information on the domain controller. Password resets are performed by a system administrator when you forget or do not know a particular password. Only password resets require symmetric key restoration. Periodically changing an account password does not require you to reset the symmetric key.

- Renaming the computer or instance that hosts the report server (a report server instance is based on a SQL Server instance name).
- Migrating a report server installation or configuring a report server to use a different report server database.
- Recovering a report server installation due to hardware failure.

You only need to back up one copy of the symmetric key. There is a one-to-one correspondence between a report server database and a symmetric key. Although you only need to back up one copy, you might need to restore the key multiple times if you are running multiple report servers in a scale-out deployment model. Each report server instance will need its copy of the symmetric key to lock and

unlock data in the report server database.

▣ Backing Up the Encryption Keys to Disk

Backing up the symmetric key is a process that writes the key to a file that you specify, and then scrambles the key using a password that you provide. The symmetric key can never be stored in an unencrypted state so you must provide a password to scramble the key when you save it to disk. After the file is created, you must store it in a secure location and remember the password that is used to unlock the file. To backup the symmetric key, you can use either the Reporting Services Configuration tool or the **rskeymgmt** utility.

How to back up encryption keys (Reporting Services Configuration Tool)

1. Start the Reporting Services Configuration tool, and then connect to the report server instance you want to configure.
2. Click **Encryption Keys**, and then click **Back Up**.
3. Type a strong password.
4. Specify a file to contain the stored key. Reporting Services appends a .snk file extension to the file. Consider storing the file on a diskette so that it is separate from the report server.
5. Click **OK**.

How to back up encryption keys (rskeymgmt)

1. Insert a diskette into the floppy disk drive if you want to store the password-protected file separately from the report server.
2. Run **rskeymgmt.exe** locally on the computer that hosts the report server. You must use the **-e** extract argument to copy the key, provide a file name, and specify a password. The following example illustrates the arguments you must specify:

 [Copy Code](#)

```
rskeymgmt -e -f a:\rsdbkey.snk -p<password>
```

3. Store the diskette in a secure location.

▣ Restoring the Encryption Keys

Restoring the symmetric key overwrites the existing symmetric key that is stored in the report server database. Restoring an encryption key replaces an unusable key with a copy that you previously saved to disk. Restoring encryption keys results in the following actions:

- The symmetric key is opened from the password protected backup file.
- The symmetric key is encrypted using the public key of the Report Server Windows service.
- The encrypted symmetric key is stored in the report server database.
- The previously stored symmetric key data (for example, key information that was already in the report server database from a previous deployment) is deleted.

To restore the encryption key, you must have a copy of the encryption key on file. You must also know the password that unlocks the stored copy. If you have the key and the password, you can run the Reporting Services Configuration tool or **rskeymgmt** utility to restore the key. The symmetric key must be the same one that locks and unlocks encrypted data currently stored in the report server database. If you restore a copy that is not valid, the report server cannot access the encrypted data currently stored in the report server database. If this occurs, you might need to delete all encrypted values if you cannot restore a valid key. If for some reason you cannot restore the encryption key (for example, if you do not have a backup copy), you must delete the existing key and encrypted content. For more information, see [Deleting and Re-creating Encryption Keys](#). For more information about creating symmetric keys, see [Initializing a Report Server](#).

How to restore encryption keys (Reporting Services Configuration Tool)

1. Start the Reporting Services Configuration tool, and then connect to the report server instance you want to configure.
2. On the Encryption Keys page, click **Restore**.
3. Select the .snk file that contains the back up copy.
4. Type the password that unlocks the file.

5. Click **OK**.

How to restore encryption keys (rskeymgmt)

1. Insert the diskette that contains the backup copy of the encryption key.
2. Run **rskeymgmt.exe** locally on the computer that hosts the report server. Use the **-a** argument to restore the keys. You must provide a fully-qualified file name and specify a password. The following example illustrates the arguments you must specify:

 [Copy Code](#)

```
rskeymgmt -a -f a:\rsdbkey.snk -p<password>
```

▣ **See Also**

Concepts

[Managing Encryption Keys](#)

Other Resources

Reporting Services Configuration Tool

rskeymgmt Utility

Backup and Restore Operations for a Reporting Services Installation

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Deleting and Re-creating Encryption Keys



[See Also](#)



[Configuring a SQL Server Installation](#) > [Reporting Services Configuration](#) > [Managing Encryption Keys](#) >

Deleting and re-creating encryption keys are activities that fall outside of routine encryption key maintenance. You perform these tasks in response to a specific threat to your report server, or as a last resort when you can no longer access a report server database.

- Re-create the symmetric key when you believe the existing symmetric key is compromised. You can also re-create the key on a regular basis as a security best practice.
- Delete existing encryption keys and unusable encrypted content when you cannot restore the symmetric key.

▣ Re-creating Encryption Keys

If you have evidence that the symmetric key is known to unauthorized users, or if your report server has been under attack and you want to reset the symmetric key as a precaution, you can re-create the symmetric key. When you re-create the symmetric key, all encrypted values will be re-encrypted using the new value. If you are running multiple report servers in a scale-out deployment, all copies of the symmetric key will be updated to the new value. The report server uses the public keys available to it to update the symmetric key for each server in the deployment.

You can only re-create the symmetric key when the report server is in a working state. Re-creating the encryption keys and re-encrypting content disrupts server operations. You must take the server offline while re-encryption is underway. There should be no requests made to the report server during re-encryption.

You can use the Reporting Services Configuration tool or the **rskeymgmt** utility to reset the symmetric key and encrypted data. For more information about how the symmetric key is created, see [Initializing a Report Server](#).

How to re-create encryption keys (Reporting Services Configuration Tool)

1. Use the SQL Server Surface Area Configuration tool to disable the Report Server Web service. If you are recreating encryption keys for a report server scale-out deployment, disable the Report Server Web service on all instances in the deployment.
 1. Start the SQL Server Surface Area Configuration tool and click **Surface Area Configuration for Features**.
 2. Select Reporting Services.
 3. For **Web Service and HTTP Access**, clear the checkbox **Enable Web Service and HTTP Access**.
2. Start the Reporting Services Configuration tool, and then connect to the report server instance you want to configure.
3. On the Encryption Keys page, click **Change**. Click **OK**.
4. Restart the Report Server Windows service. If you are recreating encryption keys for a scale-out deployment, restart the service on all instances.

5. Run the SQL Server Surface Area Configuration tool to enable Report Server Web service operations. Do this for all instances if you are working with a scale out deployment.

How to re-create encryption keys (rskeymgmt)

1. Disable Web service and HTTP access. Use the instructions in the previous procedure to stop Web service operations.
2. Run **rskeymgmt.exe** locally on the computer that hosts the report server. Use the **-s** argument to reset the symmetric key. No other arguments are required:

 [Copy Code](#)

```
rskeymgmt -s
```

3. Restart the Windows service and enable Web service operations.

▣ **Deleting Unusable Encrypted Content**

If for some reason you cannot restore the encryption key, the report server will never be able to decrypt and use any data that is encrypted with that key. To return the report server to a working state, you must delete the encrypted values that are currently stored in the report server database and then manually re-specify the values you need.

Deleting the encryption keys removes all symmetric key information from the report server database and deletes any encrypted content. All unencrypted data is left intact; only encrypted content is removed. When you delete the encryption keys, the report server re-initializes itself automatically by adding a new symmetric key. The following occurs when you delete encrypted content:

- Connection strings in shared data sources are deleted. Users who run reports get the error "The ConnectionString property has not been initialized."
- Stored credentials are deleted. Reports and shared data sources are reconfigured to use prompted credentials.
- Reports that are based on models (and require shared data sources configured with stored or no credentials) will not run.
- Subscriptions are deactivated.

Once you delete encrypted content, you cannot recover it. You must re-specify connection strings and stored credentials, and you must activate subscriptions.

You can use the Reporting Services Configuration tool or the **rskeymgmt** utility to remove the values.

How to delete encryption keys (Reporting Services Configuration Tool)

1. Start the Reporting Services Configuration tool, and then connect to the report server instance you want to configure.
2. Click **Encryption Keys**, and then click **Delete**. Click **OK**.
3. Restart the Report Server Windows service. For a scale-out deployment, do this on all report server instances.

How to delete encryption keys (rskeymgt)

1. Run **rskeymgt.exe** locally on the computer that hosts the report server. You must use the **-d** apply argument. The following example illustrates the argument you must specify:

 [Copy Code](#)

```
rskeymgt -d
```

2. Restart the Report Server Windows service. For a scale out deployment, do this on all report server instances.

How to re-specify encrypted values

1. For each shared data source, you must retype the connection string.
2. For each report and shared data source that uses stored credentials, you must retype the user name and password, and then save. For more information, see **Specifying Credential and Connection Information**.
3. For each data-driven subscription, open each subscription and retype the credentials to the subscription database.
4. For subscriptions that use encrypted data (this includes the File Share delivery extension and any third-party delivery extension that uses encryption), open each subscription and retype credentials. Subscriptions that use Report Server e-mail delivery do not use encrypted data and are unaffected by the key change.

▣ **See Also**

Concepts

[Managing Encryption Keys](#)

Other Resources

Reporting Services Configuration Tool

rskeymgmt Utility

Storing Encrypted Data in a Report Server Database

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Adding and Removing Encryption Keys for Scale-Out Deployment



[See Also](#)



[Configuring a SQL Server Installation](#) > [Reporting Services Configuration](#) > [Managing Encryption Keys](#) >

You can run Reporting Services in a scale-out deployment model by configuring multiple report servers to use a shared report server database. Membership in a scale-out deployment is based on whether the report server stores an encryption key in the report server database. You can control scale-out deployment membership by adding and removing encryption keys for specific report server instances. If you are removing nodes from the deployment, you can remove them in any order. If you are adding nodes to a deployment, you must join any new instances from a report server that is already part of the deployment.

▣ **Using the Reporting Services Configuration Tool to Configure Scale-Out Deployment**

The easiest way to configure a scale-out deployment is to use the Reporting Services Configuration tool. For more information and step-by-step instructions, see [Configuring a Report Server Scale-Out Deployment](#) and [How to: Configure a Report Server Scale-Out Deployment \(Reporting Services Configuration\)](#).

▣ Using Rskeymgmt to Configure Scale-Out Deployment

Use the **rskeymgmt** utility to initialize a report server instance to use a shared report server database. Adding a report server to a scale-out deployment requires that you initialize the report server. Initialization requires administrator permissions. You must have administrator credentials for the remote computer that hosts the report server you are joining to the deployment.

How to join a report server to a scale-out deployment (rskeymgmt)

1. Run **rskeymgmt.exe** locally on the computer that hosts a report server that is already a member of the report server scale-out deployment.
2. Use the **-j** argument to join a report server to the report server database. Use the **-m** and **-n** arguments to specify the remote report server instance you want to add to the deployment. Use the **-u** and **-v** arguments to specify an administrator account on the remote computer. If you are creating a scale-out deployment using multiple report server instances on the same computer, the syntax to use is slightly different. For more information about the syntax you should use, see **rskeymgmt Utility**.

The following example illustrates the arguments you must specify if you are joining a remote report server to a scale-out deployment (you can omit credentials if you have administrator permissions on the remote computer):

```
rskeymgmt -j -m <remotecomputer> -n <namedreportserverinstance>
```

How to remove a report server from a scale-out deployment (rskeymgmt)

1. Open the `rsreportserver.config` file of the report server you want to remove and find the installation ID. By default, this file is located at Program Files\Microsoft SQL Server\MSSQL.n\Reporting Services\ReportServer). If you installed a single instance, there will only be one `rsreportserver.config` file on the computer. If multiple instances of Reporting Services are installed, use the Server Status page in the Reporting

Services Configuration tool to find the instance identifier (for example, MSSQL.2) for the report server that you want to remove. The name of the folder that stores the program files for the report server instance will be based on the instance identifier (for example, Program Files\Microsoft SQL Server\MSSQL.2).

2. Run **rskeymgmt.exe**. You can run it on any report server that is part of the report server scale-out deployment.
3. Use the **-r** argument to release the report server instance from the scale-out deployment. The following example illustrates the arguments you must specify:

 [Copy Code](#)

```
rskeymgmt -r <installation ID>
```

▣ **See Also**

Concepts

[Managing Encryption Keys](#)

[Initializing a Report Server](#)

Other Resources

Reporting Services Configuration Tool
rskeymgmt Utility

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Reporting Services Configuration How-to Topics



[See Also](#)



[Configuring a SQL Server Installation](#) > [Reporting Services Configuration](#) >

This section contains procedures for configuring a report server instance using the Reporting Services Configuration tool.

▣ In This Section

Topic	Description
How to: Start Reporting Services Configuration	Provides steps for starting the Reporting Services Configuration tool.
How to: Configure Service Accounts (Reporting Services Configuration)	Explains how to specify account and password information for the Report Server Web service and Windows service.
How to: Configure Reporting Services to Use a Non-Default Web Site (Reporting Services Configuration)	Provides steps for choosing a Web site that is not the default Web site. These instructions are for users who might not be familiar with Web site creation.
How to: Create a Report Server Database (Reporting Services Configuration)	Provides steps for creating a report server database. This step is required for deploying a Reporting Services installation.
How to: Create a Report Server Database for SharePoint Integrated Mode (Reporting Services Configuration)	Explains how to create a report server database for a report server that runs in SharePoint integrated mode.
How to: Set Report Server Integration in SharePoint Central Administration	Provides instructions for configuring integration in SharePoint Central Administration.
How to: Switch Server Modes (Reporting Services Configuration)	Explains how to remove a report server instance from a deployment of a SharePoint technology or instance so that you can run it as a native mode report server.
How to: Configure a Report Server Scale-Out Deployment (Reporting Services Configuration)	Provides steps for configuring multiple report servers to share a report server database.
How to: Configure a Report Server Database on a Database Engine Upgraded from SQL Server 7.0 (Reporting Services Configuration)	Explains how to set the database compatibility level to avoid exceptions when creating a report server database on an upgraded Database Engine instance.
How to: Configure a Report Server	

for E-mail Delivery (Reporting Services Configuration)	Provides steps for configuring a report server for e-mail distribution.
How to: Enable Remote Errors (Reporting Services Configuration)	Explains how to set server properties on a report server to return additional information about error conditions that occur on remote servers.

▣ **See Also**

Other Resources

Reporting Services How-to Topics

Configuring Reporting Services Components

Reporting Services Configuration Tool

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Start Reporting Services Configuration



[See Also](#)



[Configuring a SQL Server Installation](#) > [Reporting Services Configuration](#) > [Reporting Services Configuration How-to Topics](#) >

Reporting Services Configuration tool is used to deploy a Reporting Services installation. You can run it on a computer that has a local report server instance. You can also use it to configure remote report server instances to which you have administrator access.

To start Reporting Services Configuration

1. Click **Start**, point to **Programs**, point to **Microsoft SQL Server 2005**, and then point to **Configuration Tools**.
2. Click **Reporting Services Configuration**. The Report Server Installation Instance Selection dialog box appears so that you can select the report server instance you want to configure.
3. In **Machine Name**, specify the name of the computer on which the report server instance is installed. The name of the local computer is specified by default, but you can also type the name of a remote SQL Server instance.

If you specify a remote computer, click **Find** to establish a connection. The report server must be configured for remote administration in advance. For more information, see **Configuring a Remote Report Server Instance** in SQL Server Books Online.

4. In **Instance Name**, choose the SQL Server 2005 Reporting Services instance that you want to configure. Only SQL Server 2005 report server instances appear in the list. You cannot configure earlier versions of Reporting Services.

5. Click **Connect**.

▣ **See Also**

Concepts

[Reporting Services Configuration How-to Topics](#)

Other Resources

Configuring Reporting Services Components

Reporting Services Configuration Tool

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Configure Service Accounts (Reporting Services Configuration)



[See Also](#)



[Configuring a SQL Server Installation](#) > [Reporting Services Configuration](#) > [Reporting Services Configuration How-to Topics](#) >

To specify the service accounts in a Reporting Services installation, use the Reporting Services Configuration tool so that other settings that depend on the service identity can be updated concurrently.

By default, the Report Server Web service runs under **Network Service** in Internet Information Services (IIS) 6.0 and under the ASP.NET account in IIS 5.0. You can modify the Web service to run under a different account. If you use a domain account, additional steps might be required before the service is fully operational.

The Report Server Windows service runs under an account that you specify during Setup, but you can change it to run under a different account, or update a password if the previous one expires.

Important:

When you modify the Report Server Windows service identity settings, each step in the service account update is recorded in the tasks panel at the bottom of the page. The Report Server Windows service will be stopped and restarted during the account update. Errors that might occur include login failures (for example, if you mistype an account or password). In the unlikely event that the symmetric key is not restored, you can perform this step yourself by clicking **Restore** in the Encryption Keys page. To verify that the account has been reset properly, open a report that uses stored credentials. If you cannot restore a backup copy of the key, you must delete the encrypted content. For more information, see [Deleting and Re-creating Encryption Keys](#) and [Backing Up and Restoring Encryption Keys](#).

Some of the steps for configuring the Web service accounts are from other articles published on the MSDN Web site. For more information about how to configure Web service accounts, see [How to create a service account for an ASP on MSDN](#).

You must be a domain administrator to create domain accounts.

☐ **To configure the Report Server Windows service**

1. Start the Reporting Services Configuration tool and connect to the report server.
2. On the Windows Service Identity page, specify the new account and password. Click **Apply**.
3. When prompted to backup the symmetric key, type a password and file name for the symmetric key backup, and click **OK**.

☐ **To configure the Report Server Web service to use a domain account in IIS 6.0**

1. Create a new domain user account by using the Active Directory Users and Computers tool in Control Panel. Give the account a descriptive name. Clear the **User must change password at next logon** option and select **Password never expires**. Assign a strong password for the account. Strong passwords should include at least seven characters and have a mixture of uppercase and lowercase letters, numbers, and other characters such as *, ?, or \$.
2. Assign ASP.NET permissions to the new account by running the following command from a command window:

```
aspnet_regiis -ga DomainName\AccountName
```

3. Start IIS Manager.

4. In Application Pools, right-click **Report Server** and click **Properties**.
5. Click **Identity**.
6. Enter the domain user account.
7. Start the Reporting Services Configuration tool and connect to the report server.
8. Open the Web Service Identity page to select the application pool you just configured.
9. Click **Apply**.

If the network is configured for Kerberos authentication, the domain administrator might need to create a Service Principal Name (SPN) that registers the domain account for the Web site. Otherwise, you might encounter HTTP 401 access denied errors. Be sure that the domain account that you register is the same one used for the application pool. For more information, see [Configuring Constrained Delegation for Kerberos \(IIS 6.0\)](#) on the Microsoft TechNet Web site.

☐ **To configure the Report Server Web service to use a domain account in IIS 5.0**

1. Create a new domain user account by using the Active Directory Users and Computers tool in Control Panel. Give the account a descriptive name. Clear the **User must change password at next logon** options and select **Password never expires**. Assign a strong password for the account. Strong passwords should include at least seven characters and have a mixture of uppercase and lowercase letters, numbers, and other characters such as *, ?, or \$.
2. Store an encrypted copy of the user account and password in the registry. In a subsequent step, you will reference this registry entry in Machine.config file.

1. Download the aspnet_setreg.exe utility from [How to use the ASP.NET utility to encrypt credentials and session state connection strings](#) on www.support.microsoft.com.
2. Run the following command from the command line, replacing the domain, user name, and password with values that are valid for the account you created in step 1.

```
aspnet_setreg -k:Software\MyASPNetApp\Identity -u:"domair
```

3. Open Machine.config using Microsoft Visual Studio or Notepad. Machine.config is located in the following folder: C:\WINNT\Microsoft.NET\Framework\v1.1.4322\CONFIG. Modify the configuration file to reference the registry key you just created by typing the following:

1. Find the `<identity impersonate="false" userName="" password=""/>` element.
2. Replace it with the following entries:

```
<identity impersonate="true"  
userName="registry:HKLM\Software\MyASPNetApp\Identity\ASF  
password="registry:HKLM\Software\MyASPNetApp\Identity\ASF
```

4. Run the Registry Editor to grant Read permissions to the account:
 1. Browse to My Computer/HKEY_LOCAL_MACHINE/SOFTWARE/MyASPNetA
 2. Right-click **ASPNET_SETREG** and click **Permissions**.
 3. Click **Add**.
 4. Enter the user account you created in step 1 and click **OK**.
 5. Select the account, click **Full Control** and click **OK**.
5. Assign file permissions to the ASP.NET temporary files folder:
 1. Browse to 'C:\WINNT\Microsoft.NET\Framework\V1.0.4322\Temporary ASP.NET Files'.
 2. Right-click **Temporary ASP.NET Files**, click **Properties**.
 3. Click **Security**, click **Add**, enter the account, and then select the account so that you can specify permissions on it.

4. Click **Write**, and click **OK**.
6. Add the account to the report server Windows security groups:
 1. Point to **Start**, point to **Administrator Tools**, and click **Computer Management**.
 2. Open **Local Users and Groups** and select **Groups**.
 3. Right-click
SQLServer2005ReportingServicesWebServiceUser\$<computer_name>
 4. Click **Add to group**.
 5. Click **Add**.
 6. Enter the account and click **OK**.

 **Note:**

Many of the steps in this procedure are from "Building Secure ASP.NET Applications: Authentication, Authorization, and Secure Communication" on MSDN. For more information about the steps, search for that document on MSDN. For more information about how to use the aspnet_setreg.exe utility, see [How to use the ASP](#) on www.support.microsoft.com.

▣ **See Also**

Tasks

[How to: Start Reporting Services Configuration](#)

Reference

[Windows Service Identity \(Reporting Services Configuration\)](#)

[Web Service Identity - Windows Server 2003 \(Reporting Services Configuration\)](#)

[Web Service Identity \(Reporting Services Configuration\)](#)

Concepts

[Configuring Service Accounts and Passwords in Reporting Services](#)
[Administering the Report Server Web Service and Windows Service](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Configure Reporting Services to Use a Non-Default Web Site (Reporting Services Configuration)

 [_See Also](#)



[Configuring a SQL Server Installation](#) > [Reporting Services Configuration](#) > [Reporting Services Configuration How-to Topics](#) >

Updated: **14 April 2006**

You can create virtual directories for the report server and Report Manager under a custom Web site. To use a non-default Web site, you must create the site, select it when you create the virtual directories, update the URLs in the Reporting Services configuration files, and then test your installation to make sure the report server is available. How you specify the URLs and test your configuration will vary depending whether you identify the Web site through a unique IP address, port number, host header name, or a combination.

Depending on how your Web site is configured, you might encounter one or more HTTP 401 errors when verifying the URL:

- "HTTP 401.1 - Unauthorized: Logon failed" can occur if you are running Windows XP Service Pack 2 or Windows Server 2003 Service Pack, and the fully qualified domain name of the Web site is different from the local computer name. There are several workarounds for resolving this error. For more information about the error and possible workarounds, go to <http://support.microsoft.com/kb/896861>.
- "HTTP 401.1 - Unauthorized: Access is denied due to invalid credentials" can occur if the application pool is a local account or domain user account, and the Web site is configured to use integrated security. Because the report server virtual directories use integrated security by default, you can expect to encounter this error when you access the report server. For more information about the error and possible workarounds, go to <http://support.microsoft.com/kb/871179>.

You can run multiple report server instances on the same computer without having to create custom Web sites for each instance. For more information, see

[Installing Multiple Instances of Reporting Services.](#)

▣ **To create the virtual directories in a new custom Web site**

1. Create a new Web site in IIS Manager. Creating a custom Web site requires that you either configure the Web site to use a specific IP address or host header name. For instructions on how to create a Web site, see the [Hosting Multiple Web Sites on a Single Server](#) topic in the Internet Information Services (IIS) product documentation.
2. Verify that the Web site is accessible and that you do not encounter authentication errors when you access the site.
3. Start the Reporting Services Configuration tool and connect to the report server that will use the new Web site.
4. On the Report Server Virtual Directory page, click **New**.
5. Select the Web site you just created, and click **OK**.
6. If you are configuring the Report Server for Secure Sockets Layer (SSL) connections, select the **Require Secure Sockets Layer (SSL)** checkbox.
7. Click **Apply** to create the report server virtual directory.
8. On the Report Manager Virtual Directory page, click **New**.
9. Select the Web site you just created. Click **OK**.
10. Click **Apply** to create the Report Manager virtual directory.
11. Verify that you can access the report server by testing the URL. To do this, type the URL in a browser window. If you get HTTP 401 errors, review the Microsoft Knowledge Base articles noted at the beginning of this topic.

A fully qualified report server URL includes the prefix, server name,

and virtual directory:

If you assigned a host header to the custom Web site, use the following syntax: `http://<hostheader>/reportserver`.

If you assigned an IP address to the custom Web site, make sure the IP address resolves to a host name, and then use the following syntax: `http://<IP address>/reportserver`.

If you assigned a unique port number to the custom site, append the port number to the Web server name: `http://<servername>:<portnumber>/reportserver`.

12. (Optional for some Web site configurations). Edit the `RSWebApplication.config` file to update the report server URL that Report Manager uses to connect to the report server.

You can omit these steps if the custom Web site is mapped to a specific IP address, and the host name resolves to the IP address. You can also omit this step if the Web site uses SSL and port 443, and you selected the SSL option when creating the report server virtual directory.

You must perform these steps if the custom site uses host headers or a custom port number:

1. Open the `RSWebApplication.config` file.
2. Delete the entry in **ReportServerVirtualDirectory**. Remove just the value; do not delete the tags.
3. In **ReportServerUrl**, type a fully qualified name to the report server instance. It should be the same value that you use previously to confirm the report server URL.

The settings should look similar to the following example:



```
<ReportServerUrl>http://myhostheader/reportserver</ReportServerUrl>  
<ReportServerVirtualDirectory></ReportServerVirtualDirectory>
```

If you are configuring multiple report server instances, repeat this step for each instance. For more information about this configuration file,

see **RSWebApplication Configuration File** in SQL Server Books Online.

13. Verify that you can access Report Manager by testing the URL. For example, if the report server URL is `http://<hostheader>/reportserver`, the Report Manager URL is most likely `http://<hostheader>/reports`.

If you get the error "The request failed with HTTP status 400: Bad Request", the URL that you specified in `RSWebApplication.config` is not valid.

14. If you are using report server e-mail delivery, edit the `RSReportServer.config` file and specify the **UrlRoot** configuration setting. The value should be the fully qualified report server URL. It should be the same value that you used previously step to confirm the report server URL. For more information about this configuration file, see **RSReportServer Configuration File** in SQL Server Books Online.

▣ **See Also**

Tasks

[How to: Start Reporting Services Configuration](#)

Concepts

[Installing Multiple Instances of Reporting Services](#)

[Configuring Report Server Virtual Directories](#)

[Creating a Report Server Database](#)

[Reporting Services Configuration How-to Topics](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Create a Report Server Database (Reporting Services Configuration)

 [See Also](#)



[Configuring a SQL Server Installation](#) > [Reporting Services Configuration](#) > [Reporting Services Configuration How-to Topics](#) >

Reporting Services uses a SQL Server database for internal storage. The database is a required component; it is used to store reports, session data, resources, and server metadata.

If you installed Reporting Services in the default configuration, the report server database is created automatically when the report server instance is installed. However, if you performed a files-only installation, you must create the database after setup is finished. To create a report server database, you can use the Reporting Services Configuration tool.

To create a report server database

1. Start the Reporting Services Configuration tool and connect to the report server instance for which you are creating the database. For more information about how to connect to a report server instance, see [How to: Start Reporting Services Configuration](#).
2. On the Database Setup page, type or select the SQL Server Database Engine instance that you want to use. You can choose either a SQL Server 2000 or SQL Server 2005 instance. Not all editions can be used to host the database. For more information about edition compatibility, see [Creating a Report Server Database](#).
3. Click **Connect**. This opens a SQL Server Connection dialog box so that you can specify credentials have permission create a database on the server. Click **OK** to connect to the server.

4. On the Database Setup page, click **New**. This reopens the SQL Server Connection dialog box.
5. Type the name of the report server database. A temporary database is created along with the primary database.
6. Choose the collation language to use, and then click **OK**.
7. On the Database Setup page, in Credentials Type, specify the credentials used by the report server to connect to the report server database. For more information about the choosing different accounts, see [Setting Up Windows Service Accounts](#) and [Configuring a Report Server Database Connection](#).
 1. Choose **Service credentials** to use the Windows service account and Web service account to connect through integrated security.
 2. Choose **Windows credentials** to specify a domain user account. A domain user account must be specified as <domain>\<user>.
 3. Choose **SQL Server credentials** to specify a SQL Server login.
8. Click **Apply**.

▣ **See Also**

Tasks

[How to: Start Reporting Services Configuration](#)

Reference

[Database Setup \(Reporting Services Configuration\)](#)

Concepts

[Administering a Report Server Database](#)

[Creating a Report Server Database](#)

[Reporting Services Configuration How-to Topics](#)

[Configuring a Report Server Database Connection](#)

Other Resources

Report Server Database

Configuring Reporting Services Components

Reporting Services Configuration Tool

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Create a Report Server Database for SharePoint Integrated Mode (Reporting Services Configuration)

 [See Also](#)



[Configuring a SQL Server Installation](#) > [Reporting Services Configuration](#) > [Reporting Services Configuration How-to Topics](#) >

New: 12 December 2006

Use the Reporting Services Configuration tool included with SQL Server 2005 Service Pack 2 to create a report server database in SharePoint integrated mode and configure the connection from the report server to the report server database. Only the report server connects to the report server database. The Windows SharePoint Services or Office SharePoint Server instance that you are integrating the report server with never connects to or retrieves data from the report server database directly.

A report server database can support native mode operations or SharePoint integration mode, but not both. If you create a report server database to support SharePoint integrated mode, you cannot convert or migrate the database to run with a native mode report server instance later. The report server database contains mode-specific metadata (specifically, item location, and permission information) that is not compatible in the other mode.

When created for SharePoint integrated mode, the report server database stores server properties, report execution snapshots, report history, subscription definitions, and schedules. It stores a secondary copy of reports, report models, shared data sources, and resources to improve processing performance on the server. Primary storage for report documents is in the SharePoint content databases. For more information, see [Storing and Synchronizing Report Server Content With SharePoint Databases](#).

There are three steps to creating the report server database:

- Connect to the Database Engine instance that hosts the database.
- Enter a name for the database, select a check box for SharePoint

integrated mode, and create the database.

- Specify credentials for the report server connection to the database.

Be sure to perform all three steps before you click **Apply**.

After you create the database, complete the integration steps by starting SharePoint 3.0 Central Administration and specifying the report server integration settings. For more information, see [How to: Set Report Server Integration in SharePoint Central Administration](#).

▣ **To create a report server database in SharePoint Integrated Mode**

1. Click **Start**, point to **Programs**, point to **Microsoft SQL Server 2005**, and then point to **Configuration Tools**.
2. Click **Reporting Services Configuration**. The **Report Server Installation Instance Selection** dialog box appears so that you can select the report server instance you want to configure.
3. In **Machine Name**, specify the name of the computer on which the report server instance is installed. The name of the local computer is specified by default, but you can also type the name of a remote SQL Server instance.

If you specify a remote computer, click **Find** to establish a connection. The report server must be configured for remote administration in advance. For more information, see **Configuring a Remote Report Server Instance** in SQL Server Books Online.

4. In **Instance Name**, choose the SQL Server 2005 Reporting Services instance that you want to configure. Only SQL Server 2005 report server instances appear in the list. You cannot configure earlier versions of Reporting Services.
5. Click **Connect**.
6. Click **Database Setup** to open the Database Setup page.

7. Enter the name of the SQL Server Database Engine you want to use.
8. Click **Connect**.
9. Click **New**.
10. In the **SQL Server Connection** dialog box, enter a name for the new database.
11. Select the **Create the report server database in SharePoint integrated mode** check box.
12. Click **OK**.
13. On the Database Setup page, specify the credentials that are used to connect to the report server database.
 1. Choose **Service credentials** to use the Windows service account and Web service account to connect through integrated security.
 2. Choose **Windows credentials** to specify a domain user account. A domain user account must be specified as <domain>\<user>.
 3. Choose **SQL Server credentials** to specify a SQL Server login.
14. Click **Apply**.
15. Restart the Report Server Windows service.

▣ **See Also**

Tasks

[How to: Switch Server Modes \(Reporting Services Configuration\)](#)

Concepts

[Configuring Reporting Services for SharePoint 3.0 Integration](#)

[Deployment Modes for Reporting Services](#)

[Configuring a Report Server Database Connection](#)

[Administering a Report Server Database](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Set Report Server Integration in SharePoint Central Administration

 [See Also](#)



[Configuring a SQL Server Installation](#) > [Reporting Services Configuration](#) > [Reporting Services Configuration How-to Topics](#) >

New: **12 December 2006**

As a server administrator, you must specify integration settings that determine the connection between the SharePoint technology instance and a Reporting Services report server.

You must be an administrator on the SharePoint site or farm to configure report server integration, and you must have installed the Reporting Services Add-in that provides the configuration pages. You must also have a target report server that is already configured for SharePoint integrated mode.

The settings that you specify are stored in the configuration database. If you are configuring integration for a farm that includes several virtual servers, the settings will be used by all Web front-ends in a SharePoint farm.

Specifying the settings in Central Administration is only one part of configuring server integration. For more information about the complete set of tasks, see [Configuring Reporting Services for SharePoint 3.0 Integration](#).

 **Note:**

After you define a report server URL, use caution when modifying it later. Modifying a report server URL will disrupt the connection between the servers, affecting reports and models that are currently processing. If you change the report server URL on a production server, be sure to follow the planned downtime procedures established for your organization.

□ To specify report server integration settings in Central Administration

1. In Administrator Tools, click **SharePoint 3.0 Central Administration**.
2. Click the **Application Management** tab.
3. In the Reporting Services section, click **Manage integration settings**. If the Reporting Services section is not available, the Reporting Services Add-in must be downloaded and installed.
 1. In **Report Server Web Service URL**, specify the report server that you are integrating with the SharePoint deployment. This URL identifies the report server site that is used with the current SharePoint site or farm. The URL can point to a single report server instance, or it can be the virtual server name for a set of load-balanced report servers that run in a scale-out deployment. Examples of how a report server URL might be constructed include:
 - <http://your-server-name/reportserver> (this example might be used if the report server is on a different computer).
 - <http://your-server-name:8080/reportserver> (this example might be used if the report server is on the same computer as the SharePoint deployment that you are integrating and you mapped the default Web site to port 8080)
 - <https://your-server-name:443/reportserver>
If you installed the report server and the SharePoint technology instance side-by-side, and the SharePoint technology instance is using default values, port 80 will be used to access the top-level SharePoint site. When you install the report server on the same Web server, you must either assign and use a different port number on the default Web site, or create the report server virtual directory under a custom Web site that is uniquely identified through a port number, IP address, or a host header. For more information about report server URLs and the virtual directory, see [Configuring Report Server Virtual Directories](#).
4. In the Reporting Services section, click **Grant database access**. During

this step, the service account information is retrieved from the report server instance. A database login and permissions is created for each Reporting Services service account. Integrating a Reporting Services report server with a SharePoint Web application requires that the report server be able to access the SharePoint configuration and content databases. Both the Report Server Web service and Report Server Windows service must be able to connect as trusted users with read and write permissions.

1. Specify the name of the report server. By default, this is the name of the computer.
 2. Specify whether the instance is the default instance or a named instance. If you are not sure, start the Reporting Services Configuration tool and connect to the report server. In the **Instance Selection** dialog box, expand **Instance Name** to view a list of the instances installed on the server. MSSQLSERVER is the name of the default instance. All other instances are named instances.
 3. Click **OK**. The Enter Credentials dialog box opens.
 4. In the **Enter Credentials** dialog box, specify the User Name and Password to connect to the report server to retrieve the Web service and Windows service accounts. Be sure to enter credentials for an account that is a member of the local Administrators group on the report server computer.
5. In the Reporting Services section, click **Set server defaults**.
1. In **Report History Default**, set a site-wide default value for the number of copies of report history to retain. The default value provides an initial setting that establishes the number of snapshots that can be stored for each report. You can specify different limits in property pages for specific reports
 2. In **Report Processing Timeout**, specify whether report processing times out on the report server after a certain number of seconds. This value applies to report processing on a report server. It does not affect data processing on the database server that provides the data for your report. The report processing timer clock begins when the report is selected and ends when the report opens. When you set this value, specify enough time to complete both data processing and report processing.

3. In **Report Processing Log**, specify whether the report server generates trace logs and the number of days the log is kept. You can specify whether the report server generates trace logs and the number of days the logs are kept. The logs are stored on the report server computer in the \Microsoft SQL Server\MSSQL.n\ReportServer\Log folder. A new log file is started each time the service is restarted. For more information about log files, see **Reporting Services Trace Logs**.
4. In **Enable Windows Integrated Security**, specify whether a connection to a report data source can be made using the Windows security token of the user who requested the report.
5. In **Enable Ad Hoc Reporting**, specify whether users can perform ad hoc queries from a Report Builder report. Setting this option sets the EnableLoadReportDefinition property on the report server. If you clear this option, the report server will not generate clickthrough reports for reports that use a report model as a data source.
6. Click **OK**.

▣ **See Also**

Concepts

[Reporting Services and SharePoint Technology Integration](#)

[Configuring Report Server Virtual Directories](#)

[Installing the Reporting Services Add-in](#)

[Viewing and Managing Report Server Items from a SharePoint Site](#)

Other Resources

Managing Report History

Setting Time-out Values for Report Execution

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Switch Server Modes (Reporting Services Configuration)



[See Also](#)



[Configuring a SQL Server Installation](#) > [Reporting Services Configuration](#) > [Reporting Services Configuration How-to Topics](#) >

New: **12 December 2006**

If you configured a report server to run in SharePoint integrated mode, you can detach the report server from a deployment of a SharePoint technology instance by switching the server to native mode.

To switch modes, you point the server to a native mode report server database. You do not have to re-install the report server to switch modes.

Although you can switch the mode for the report server, you cannot switch the mode for the report server database. A report server database contains mode-specific data that is unusable in different context. If you created a report server database in SharePoint integrated mode, you cannot use it with a native mode report server.

If you want to remove integration settings from both the report server and the SharePoint product or technology instance, you should also uninstall the Reporting Services Add-in. For more information, see [How to: Uninstall the Reporting Services Add-in](#).

There are three ways to switch from SharePoint integrated mode to native mode. From the Reporting Services Configuration tool, in the Database Setup page, do one of the following:

- Click **Change** and create a new database.
- From the list of existing databases, select an existing report server database that already runs in native mode, specify credentials, and click **Apply**.
- Click **New** and create a new database. Be sure to clear the **Create the report server database in SharePoint integrated mode** check box.

This topic provides steps for the first approach. For more information about how to configure a native mode report server to run in SharePoint integrated mode, see [Configuring Reporting Services for SharePoint 3.0 Integration](#).

□ **To switch to native mode**

1. Click **Start**, point to **Programs**, point to **Microsoft SQL Server 2005**, and then point to **Configuration Tools**.
2. Click **Reporting Services Configuration**. The **Report Server Installation Instance Selection** dialog box appears so that you can select the report server instance you want to configure.
3. In **Machine Name**, specify the name of the computer on which the report server instance is installed. The name of the local computer is specified by default, but you can also type the name of a remote SQL Server instance.
4. In **Instance Name**, choose the SQL Server 2005 Reporting Services instance that you want to configure. Only SQL Server 2005 report server instances appear in the list. You cannot configure earlier versions of Reporting Services.
5. Click **Connect**.
6. Click **Database Setup** to open the Database Setup page. The page contains information about the database currently in use. The **Server Mode** should be set to **SharePoint integrated**.
7. Click **Change**.
8. Click **OK** to confirm that you want to create a new database.
9. In the **SQL Server Connection** dialog box, enter a name for the new database.
10. Verify that the **Create the report server database in SharePoint integrated mode** check box is not selected.

11. Click **OK**.
12. On the Database Setup page, specify the credentials that are used to connect to the report server database.
 1. Choose **Service credentials** to use the Windows service account and Web service account to connect through integrated security.
 2. Choose **Windows credentials** to specify a domain user account. A domain user account must be specified as <domain>\<user>.
 3. Choose **SQL Server credentials** to specify a SQL Server login.
13. Click **Apply**.

When you switch from a SharePoint integrated report server to native mode, the location of the content and the permissions used to control access to the content is redefined. Content must be republished to the report server, you must assign roles on the report server that grants access to items and operations, and you must redefine subscriptions and scheduled operations. Report history that you created on the report server that ran in SharePoint integrated mode cannot be recreated on a native mode report server.

▣ **See Also**

Tasks

[How to: Uninstall the Reporting Services Add-in](#)

[How to: Set Report Server Integration in SharePoint Central Administration](#)

[How to: Create a Report Server Database for SharePoint Integrated Mode \(Reporting Services Configuration\)](#)

Concepts

[Deployment Modes for Reporting Services](#)

[Configuring Reporting Services for SharePoint 3.0 Integration](#)

[Configuring a Report Server Database Connection](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Configure a Report Server Scale-Out Deployment (Reporting Services Configuration)

 [See
Also](#)



[Configuring a SQL Server Installation](#) > [Reporting Services Configuration](#) > [Reporting Services Configuration How-to Topics](#) >

Reporting Services supports a scale-out deployment model that allows you to run multiple report server instances that share a single report server database. To create a scale-out deployment, you use Setup and the Reporting Services Configuration tool. You run Setup to install the report servers. You run the Reporting Services Configuration tool to configure each report server to use the shared database. You can only connect to and configure one report server at a time.

When specifying a report server database that will be used for storing internal data, you can create a new database or you can upgrade an existing report server database. This topic assumes you are creating a new report server database. For more information about upgrading a report server database or an existing scale-out deployment, see [Upgrading a Report Server Database](#) and [Upgrading a Report Server Web Farm](#).

To avoid database compatibility errors when connecting the server instances to the shared database, be sure that all instances are the same version. For example, if you create the report server database using a report server instance that has Service Pack 1, all other instances in the same deployment must also have Service Pack 1 applied. You can use different editions as long as all of the editions support scale-out deployment. For more information, see [Configuring a Report Server Scale-Out Deployment](#).

If you are installing new report server instances that will be part of the deployment, you can choose whether to install a default configuration or a files-only installation (in the SQL Server Installation Wizard, this is the **Install but do not configure server** option on the Report Server Installation Options page). These steps assume a files-only installation, and that you will run the Reporting Services Configuration tool to configure the server after Setup is finished. For

more information, see [Installing Reporting Services Using Setup](#).

 **Note:**

You must use other products to support access through a virtual server and cluster deployment. Reporting Services does not provide load balancing functionality or virtual server features that allow you to access all report server instances through a shared URL address.

□ **To install the first report server instance**

1. Install the first report server instance that is part of the deployment.
2. Start the Reporting Services Configuration, and then configure the virtual directories and service accounts using the Report Server Virtual Directory page, Report Manager Virtual Directory page, and the Report Manager Web Service Identity page. For more information, see [Configuring Report Server Virtual Directories](#) and [How to: Start Reporting Services Configuration](#).
3. On the Database Setup page, type or select the name of the SQL Server instance that will host the report server database. You can select a local or remote instance. You can use a SQL Server 2000 or SQL Server 2005 instance.
4. If you are connecting to a remote instance, you might need to enable that instance for remote connections. Some versions of SQL Server 2005 do not enable remote connections by default. To confirm whether remote connections are allowed, use the SQL Server Surface Area Configuration tool and view the Remote Connections setting of the target instance. If the remote instance is also a named instance, verify that SQL Server Browser service is enabled and running on the target server. SQL Server Browser service provides the port number that is used by the named instance to the Reporting Services Configuration tool.
5. Click **Connect** to connect to the SQL Server instance.

6. Click **New** to create the report server database. The **SQL Server Connection** dialog box appears.
 1. Specify an account that has permission to create a database.
 2. Click **OK** to create the database.
7. On the Database Setup page, in **Credential Type**, select the type of account and credentials that the report server will use to connect to the report server database.

If you select **Service Accounts** and you are using a remote SQL Server instance, be sure that the accounts under which the services run have network access. For more information about credentials, see [Configuring a Report Server Database Connection](#).

8. Click **Apply**. The Reporting Services Configuration tool creates the database and configures the connection.
9. On the Encryption Keys page, backup a copy of the symmetric key. This is a best-practice recommendation. You should always back up the key when you first create the database.
10. On the Initialization page, verify that an entry for the report server instance is created. The **Initialized** checkbox should be selected.
11. (Optional) Finish report server configuration by specifying report server e-mail delivery settings and the unattended execution account.
12. (Optional) Verify that the report server is operational. For more information, see [How to: Verify a Reporting Services Installation](#).

▣ **To install and configure the second report server instance**

1. Run Setup to install a second instance of Reporting Services on a different computer or as a named instance on the same computer.
2. Use the Reporting Services Configuration tool to connect to and

configure the new instance you just installed. For more information, see [How to: Start Reporting Services Configuration](#).

3. Configure the service accounts and virtual directories for the new server instance. You can use the same accounts and virtual directory names that you specified for the first instance, or different ones.
4. On the Database Setup page, in **Server Name**, specify the SQL Server Database Engine instance that hosts the report server database you want to use. This must be the same server that you connected to in step 3 of the previous set of the instructions. If you are using a remote instance, refer to step 3 of the previous set of the instructions for recommendations on how to connect to a remote instance.
5. Click **Connect**.
6. On the Database Setup page, in **Database Name**, select the database you created in step 5 of the previous set of the instructions. If the database list is empty, repeat the previous two steps to establish a connection to the server.
7. On the Database Setup page, in **Credential Type**, select the type of account and credentials that the report server will use to connect to the report server database. You can use the same credentials as the first report server instance or different credentials.
8. Click **Apply**. The **SQL Server Connection** dialog box appears. Click **OK** to connect to the server. The report server connects to the report server database and adds its key information. Key information is required for initialization.
9. On the Initialization page, verify that an entry for the second report server instance is created. The **Initialized** checkbox should be empty. The second report server instance is not yet initialized.
10. Reconnect to the first report server instance. Because it is already initialized for reversible encryption operations, it will be used to initialize the new report server instance for scale-out deployment. Initializing a new report server instance can only be done from a report server that is already initialized.

11. On the Initialization page, select the report server instance you are joining to the deployment, and click **Initialize**. After you click Initialize, the **Initialized** checkbox will be selected, indicating that the report server instance is part of the same scale-out deployment. You might need to refresh the page to see the change. If the report server instance is not listed on the Initialization page, verify that you configured the instance to use the shared database.
12. Repeat these steps to add additional report servers to the deployment. If you plan to run the report servers in a server cluster, additional configuration is required. For more information, see [Configuring a Report Server Scale-Out Deployment](#).

▣ **See Also**

Tasks

[How to: Start Reporting Services Configuration](#)

Concepts

[Adding and Removing Encryption Keys for Scale-Out Deployment](#)
[Reporting Services Configuration How-to Topics](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Configure a Report Server Database on a Database Engine Upgraded from SQL Server 7.0 (Reporting Services Configuration)

 [See Also](#)



[Configuring a SQL Server Installation](#) > [Reporting Services Configuration](#) > [Reporting Services Configuration How-to Topics](#) >

Before creating a report server database on a SQL Server Database Engine instance that has been upgraded from SQL Server 7.0, you must run a stored procedure to set the compatibility level. Setting the compatibility is necessary to avoid the following exception: "Incorrect syntax near 'COLLATE'. You may need to set the compatibility level of the current database to a higher value to enable this feature."

This error occurs when you try to create a report server database on a SQL Server instance that has been upgraded from version 7.0.

To set the compatibility level on an upgraded database

1. Run `exec sp_dbcmptlevel` on the master database. For more information, see **sp_dbcmptlevel (Transact-SQL)**.
2. Start the Reporting Services Configuration tool and connect to the report server instance you want to configure.
3. In Database setup, connect to the SQL Server Database Engine instance for which you ran the stored procedure.
4. Click **New** to create the database.
5. Specify the database name and the account used to create the database, and then click **OK**.
6. Specify credentials for the report server database connection. For more

information, see [Configuring a Report Server Database Connection](#).

▣ **See Also**

Concepts

[Creating a Report Server Database](#)

[Reporting Services Configuration How-to Topics](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Configure a Report Server for E-mail Delivery (Reporting Services Configuration)

 [See Also](#)



[Configuring a SQL Server Installation](#) > [Reporting Services Configuration](#) > [Reporting Services Configuration How-to Topics](#) >

New: 5 December 2005

Report Server E-mail is the default delivery extension for subscriptions, but it must be configured before you can use it. Configuring a report server for e-mail delivery requires that you have access to a Simple Mail Transport Protocol (SMTP) server. You must also have permission to send e-mail to the SMTP server for subsequent delivery to valid e-mail accounts.

If you do not configure a report server for e-mail delivery, or you do not have access to an SMTP server, you can use the File Share delivery extension to distribute reports to shared folders that are accessible over network connections.

To configure report server e-mail, you can use the Reporting Services Configuration tool or you can edit the configuration files. The Reporting Services Configuration file provides the minimal settings that the report server requires. If you want to use a local SMTP service or specify advanced settings, modify the configuration files instead.

Depending on security policies in your organization, you might need to request permission in advance to send e-mail from a corporate SMTP server. Check with your system or network administrator for more information.

If you want to evaluate report server e-mail functionality or troubleshoot a subscription or delivery problem, you can configure report server e-mail to use a local SMTP service that delivers reports to a local folder. For more information, see [Configuring a Report Server for E-Mail Delivery](#).

 **To configure report server e-mail using the Reporting Services**

Configuration tool

1. Verify that the Report Server Windows service has **Send As** permissions on the SMTP server. Start the Reporting Services Configuration tool and connect to the report server instance.
2. On the Email Settings page, enter the name of the SMTP server. This value can be an IP address, a UNC name of a computer on your corporate intranet, or a fully qualified domain name.
3. In **Sender Address**, enter the name an account that has permission to send e-mail from the SMTP server.
4. Click **Apply**.

☐ To configure a remote SMTP Service for the report server

1. Verify that the Report Server Windows service has **Send As** permissions on the SMTP server.
2. Open the RSReportServer.config file in a text editor.
3. Verify that **<UrlRoot>** is set to the report server URL address. This value is set when you configure the report server and it should be filled in already. If it is not set, type the report server URL address.
4. In the Delivery section, find **<ReportServerEmail>**.
5. In **<SMTPServer>**, type the name of the SMTP server. This value can be an IP address, a UNC name of a computer on your corporate intranet, or a fully qualified domain name.
6. Verify that **<SendUsing>** is set to 2. If it is set another value, the report server is not configured to use a remote SMTP service.
7. In **<From>**, type the name an account that has permission to send e-

mail from the SMTP server.

8. Save the file.

The report server will use the new settings automatically; you do not need to restart the service. You can specify additional SMTP settings to further configure how the SMTP server is used for report server e-mail delivery. For more information, see [Configuring a Report Server for E-Mail Delivery](#) and **RSReportServer Configuration File**.

▣ **To configure a local SMTP Service for the report server**

1. In Control Panel, click **Add or Remove Programs**.
2. Click **Add/Remove Windows Components** to start the Windows Component Wizard.
3. Select **Application Server** and click **Details**.
4. Select **Internet Information Services (IIS)** and click **Details**.
5. Select the **SMTP Service** checkbox and click **OK**.
6. On the Windows Component Wizard, click **Next**. Click **Finish**.
7. Verify that the service is running in the **Services** console.
8. Open the RSReportServer.config file in a text editor.
9. Verify that **<UrlRoot>** is set to the report server URL address. This value is set when you configure the report server and it should be filled in already. If it is not set, type the report server URL address.
10. In the Delivery section, find **<ReportServerEmail>**.
11. In **<SMTPServer>**, clear any values for this setting, but do not delete the tags.

12. Set <**SendUsing**> to 1. If it is set another value, the report server is not configured to use a local SMTP service.
13. Set <**SMTPServerPickupDirectory**> to a folder on the local drive.
14. Set <**From**> to an account that has permission to send e-mail from the SMTP server.
15. Save the file.

▣ **See Also**

Concepts

[Configuring a Report Server for E-Mail Delivery](#)

Other Resources

How to: Configure the Default Delivery Extension (Reporting Services Configuration)

File Share Delivery in Reporting Services

E-Mail Delivery in Reporting Services

Reporting Services Configuration Tool

Distributing Reports Through Subscriptions

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

How to: Enable Remote Errors (Reporting Services Configuration)



[See Also](#)



[Configuring a SQL Server Installation](#) > [Reporting Services Configuration](#) > [Reporting Services Configuration How-to Topics](#) >

New: 17 July 2006

You can set server properties on a report server to return additional information about error conditions that occur on remote servers. If an error message contains the text "For more information about this error, navigate to the report server on the local server machine, or enable remote errors", you can set the **EnableRemoteErrors** property to access additional information that can help you troubleshoot the problem.

Note:

You can edit the **ConfigurationInfo** table in the report server database to set **EnableRemoteErrors** to **True**, but if the report server is actively used, you should use script to modify the settings.

Enable remote errors through script

1. Create a text file and copy the following script into the file.

[Copy Code](#)

```
Public Sub Main()  
    Dim P As New [Property]()  
    P.Name = "EnableRemoteErrors"  
    P.Value = True  
    Dim Properties(0) As [Property]  
    Properties(0) = P  
    Try  
        rs.SetSystemProperties(Properties)
```

```
        Console.WriteLine("Remote errors enabled.")
    Catch SE As SoapException
        Console.WriteLine(SE.Detail.OuterXml)
    End Try
End Sub
```

2. Save the file as EnableRemoteErrors.rss.
3. Click **Start**, point to **Run**, type **cmd**, and click **OK** to open a command prompt window.
4. Navigate to the directory that contains the .rss file you just created.
5. Type the following command line, replacing *servername* with the actual name of your server:

 [Copy Co](#)

```
rs -i EnableRemoteErrors.rss -s http://servername/ReportServe
```

▣ **See Also**

Other Resources

Report Server System Properties

Configuring a Report Server for Remote Administration

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Reporting Services Configuration F1 Help



[See Also](#)



[Configuring a SQL Server Installation](#) > [Reporting Services Configuration](#) >

This section provides F1 help for the Reporting Services Configuration tool.

▣ In This Section

Topic	Description
Report Server Status (Reporting Services Configuration)	Provides information about the report server instance you are configuring.
Report Server Virtual Directory (Reporting Services Configuration)	Provides information about the virtual directory configuration for the report server.
Report Manager Virtual Directory (Reporting Services Configuration)	Provides information about the virtual directory configuration used to access Report Manager.
Windows Service Identity (Reporting Services Configuration)	Provides information about the Report Server Windows service settings.
Web Service Identity - Windows Server 2003 (Reporting Services Configuration)	Provides information about the Report Server Web service settings for Internet Information Services 6.0.
Web Service Identity (Reporting Services Configuration)	Provides information about the Report Server Web service settings for Internet Information Services 5.1 and 5.5.
Database Setup (Reporting Services Configuration)	Provides information about the report server databases that store reports, report models, and server metadata. Use this page to create and configure the database.
Encryption Keys (Reporting Services Configuration)	Provides information about the symmetric keys used for reversible encryption.
Initialization (Reporting Services Configuration)	Provides information about adding and removing report server instances to a scale-out deployment.
E-mail Settings (Reporting Services Configuration)	Provides information about the report server email delivery extension settings.
Execution Account (Reporting Services Configuration)	Provides information about unattended execution account configuration.

▣ **See Also**

Tasks

[How to: Start Reporting Services Configuration](#)

Other Resources

Reporting Services Configuration Tool
Configuring Reporting Services Components

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Report Server Status (Reporting Services Configuration)



[See Also](#)



[Configuring a SQL Server Installation](#) > [Reporting Services Configuration](#) > [Reporting Services Configuration F1 Help](#) >

Use this page to view information about the report server instance you are currently connected to. This page is the start page for report server configuration. Additional pages are available to configure virtual directories, service accounts, the report server database, report server e-mail delivery, scale-out deployment and single server initialization, and symmetric key management.

The Legend at the bottom of the page shows icons that indicate whether settings are configured. It is possible to specify settings in ways that do not result in a valid configuration. You should always test a Reporting Services installation to verify it works as expected. Visual indicators about configuration status are not a substitute for tests that verify a successful deployment. For more information about how to verify a deployment, see [How to: Verify a Reporting Services Installation](#).

Users of the SQL Server 2005 Express Edition with Advanced Services (SQL Server Express) should note the following:

- Report server e-mail and scale-out deployment are not supported in SQL Server Express.
- The report server database must be hosted on a local SQL Server Express Database Engine instance. You cannot use a remote SQL Server instance to host the database.
- Data sources used in a report must also be SQL Server databases that run on the local SQL Server Express Database Engine instance. You cannot use remote data sources or other data source types. To use additional data source types, you must use a different edition of Reporting Services.

▣ Options

Connect

Opens the Report Server Installation Instance Selection dialog box so that you can specify a different report server to configure. To connect, you must be a local administrator on the computer that runs the report server you are configuring.

You can connect to a local or a remote report server instance. For more information, see **Configuring a Report Server for Remote Administration** in SQL Server Books Online.

Refresh

Updates the current page with the most recent report server status.

Instance Name (Instance Properties)

Displays information about the report server instance to which you are currently connected. Report server instance names are based on SQL Server named instances. The default instance for a report server is MSSQLSERVER. If you are using SQL Server Express, the default instance is SQLEXPRESS.

Instance ID (Instance Properties)

Each component of SQL Server 2005 can be run as a separate instance. You can also have multiple instances of the same component. Each component instance is assigned an identifier that corresponds to a folder on the file system that stores program files for that instance. The identifier is assigned by Setup in the order in which the instance is installed. For example, if you install a Database Engine instance, an Analysis Services instance, and a Reporting Services instance, the Database Engine will be installed first (as MSSQL.1), Analysis Services will be installed second (as MSSQL.2), and Reporting Services instance will be installed third (as MSSQL.3). If you install a second instance of a component that you already installed (for example, another instance of the Database Engine), it will be installed as

MSSQL.4.

Initialized (Instance Properties)

Shows whether the report server is operational and has the ability to store and retrieve encrypted data.

Server Status (Instance Properties)

Shows whether the Report Server Windows service is running.

Start

Starts the Report Server Windows service. Restarting the Windows service is necessary after some configuration changes (for example, when reconfiguring a report server after a computer name change).

Stop

Stops the Report Server Windows service. Stopping the service causes the report server to stop working. For more information, see [Starting and Stopping the Report Server Windows Service](#).

Legend

Shows the icons used for each configuration page to indicate the state for specific categories of settings.

Not supported in the current mode refers to the deployment mode of the server, which can run in native mode or SharePoint integrated mode. Unless you specifically configure the server for SharePoint integrated mode, configuration settings for that mode are flagged as "not supported". For more information about modes, see [Deployment Modes for Reporting Services](#) in SQL Server Books Online.

▣ **See Also**

Tasks

[How to: Start Reporting Services Configuration](#)

Concepts

[Reporting Services Configuration F1 Help](#)

[Initializing a Report Server](#)

[Installing Multiple Instances of Reporting Services](#)

Other Resources

[Configuring Reporting Services Components](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Report Server Virtual Directory (Reporting Services Configuration)



[See Also](#)



[Configuring a SQL Server Installation](#) > [Reporting Services Configuration](#) > [Reporting Services Configuration F1 Help](#) >

Use the Report Server Virtual Directory page to configure the virtual directory for the report server. The virtual directory name is part of the URL used to access the Report Server Web service. The complete URL includes a prefix (http:// or https://), a server name, and a virtual directory. The server name might be a computer name on an internal network or a fully qualified domain name that is resolved by a domain name server on a private or public network.

The Report Server Web service requires a dedicated virtual directory that is used only by the report server. All Internet Information Services (IIS) settings that are required by Reporting Services are configured through the virtual directory you create.

You can use this page to create a report server virtual directory or re-apply the default settings to an existing virtual directory. If you are creating a virtual directory for a new Reporting Services installation, click **New**. When you create the virtual directory, the Reporting Services Configuration tool creates and configures the virtual directory through Internet Information Services (IIS). It also updates the configuration files with the report server URL. If the computer has a Secure Sockets Layer (SSL) certificate, you can specify that all calls to the Report Server Web service use a certificate.

You cannot select a virtual directory that you previously created. If you want to reuse a virtual directory name, you must use IIS Manager to delete the virtual directory. You can then use this page to create a new virtual directory with a previously used name.

Note:

Reporting Services requires a dedicated virtual directory for a report server. You cannot use an existing directory. In a new deployment, you must create a new virtual directory for report server access.

▣ Options

Name

Specifies the virtual directory name.

Web Site

Specifies the name of the Web site used to access the report server instance.

If you are using a custom Web site, additional configuration is usually required if you are to avoid access denied errors. For more information, see [How to: Configure Reporting Services to Use a Non-Default Web Site \(Reporting Services Configuration\)](#).

New

Opens the **Create a Virtual Directory** dialog box so that you can select an existing Web site to host the report server and type a virtual directory name. You must specify a unique name for the virtual directory. You must choose an existing Web site to host it.

Apply Default Settings

Click **Apply Default Settings** to replace the existing virtual directory configuration with the default configuration. No custom settings are preserved.

Require Secure Sockets Layer (SSL) connections

Select this checkbox to specify the SSL connection level. You must have a certificate installed to use this option. You must also modify the UrlRoot configuration setting in the RSReportServer.config file so that it specifies the fully qualified name of the computer for which the certificate is registered. For more information, see **Configuring a Report Server for Secure Sockets Layer (SSL) Connections** in SQL Server Books Online.

Require for:

Specifies the SSL connection level in the report server properties. There are three levels:

1 - Connections only

2 - Report data

3 - Entire Web service API

The levels are cumulative. Level 3 is the most secure. Level 0 is not a value in the dropdown list, but you can use it to specify no SSL connection. For more information, see **Using Secure Web Service Methods** in SQL Server Books Online.

Certificate Name

Specifies the fully qualified computer name for which the certificate is registered. The name that you specify must be identical to the name for which the certificate is registered.

▣ **See Also**

Concepts

[Reporting Services Configuration F1 Help](#)

[Configuring Report Server Virtual Directories](#)

[Configuring a Report Server for Internet Access](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Report Manager Virtual Directory (Reporting Services Configuration)

 [See Also](#)



[Configuring a SQL Server Installation](#) > [Reporting Services Configuration](#) > [Reporting Services Configuration F1 Help](#) >

Use the Report Manager Virtual Directory page to create a virtual directory used to access Report Manager or re-apply the default settings to the Report Manager virtual directory. The virtual directory name is part of the URL used to access Report Manager.

If you are creating a virtual directory for a new Reporting Services installation, click **New** to specify a name. When you specify the virtual directory, the Reporting Services Configuration tool creates and configures the virtual directory through Internet Information Services (IIS). It also updates the configuration files with the Report Manager virtual directory name.

If you are in the process of configuring the report server to run in SharePoint integrated mode, do not create the virtual directory. Report Manager is not supported on a report server that runs in SharePoint integrated mode.

 **Note:**

Reporting Services requires a dedicated virtual directory for Report Manager. You cannot use an existing directory. In a new deployment, you must create a new virtual directory for Report Manager access.

You cannot select a virtual directory that you previously created. If you want to reuse a virtual directory name, you must use IIS Manager to delete the virtual directory. You can then use this page to create a new virtual directory with a previously used name.

▣ Options

Name

Specifies the virtual directory name.

Web Site

Specifies the name of the Web site used to access Report Manager.

New

Opens the **Create a Virtual Directory** dialog box so that you can select an existing Web site to host the report server and type a virtual directory name. You must specify a unique name for the virtual directory. You must choose an existing Web site to host it.

Apply Default Settings

Click **Apply Default Settings** to replace the existing virtual directory configuration with the default configuration. No custom settings are preserved.

▣ **See Also**

Concepts

[Reporting Services Configuration F1 Help](#)

[Configuring Report Server Virtual Directories](#)

[Deployment Modes for Reporting Services](#)

Other Resources

Report Manager

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Windows Service Identity (Reporting Services Configuration)



[See Also](#)



[Configuring a SQL Server Installation](#) > [Reporting Services Configuration](#) > [Reporting Services Configuration F1 Help](#) >

Updated: 12 December 2006

Use the Windows Service Identity page to specify the account under which the Report Server Windows service runs. This account is initially configured during Setup. You can modify it if you want to change the account or password.

It is strongly recommended that you use the Reporting Services Configuration tool to update the service identity because other internal settings that depend on the service identity are automatically updated at the same time.

When choosing an account, it is helpful to know how the service is used. The Report Server Windows service is used to initialize a report server so that it can encrypt and store sensitive data, run scheduled report and subscription processing in the background, and maintain the report server database.

Account types that you can choose for the Windows service include a domain user account, a local user account, or a built-in account (such as **Network Service**, **Local Service**, or **Local System**). For most of the tasks that the Windows service must perform, the type of account you choose does not affect its ability to perform those tasks; the Report Server Windows service is fully functional no matter which account you choose.

Domain accounts and **Network Service** are recommended; both offer the same benefits in terms of access to network domain controllers, public file shares, and corporate e-mail servers. Both provide the Report Server Windows service with network logon permissions.

Local Windows accounts or built-in local accounts can be used successfully, but they introduce requirements or considerations for how you set other configuration settings, and on subscription creation and delivery:

- Running the service under a local account will limit your options later when you configure a connection to a remote report server database. Specifically, if you are using a remote report server database, you will need to configure the connection to use a domain user account or SQL Server database user that has permission to log on to the remote SQL Server instance.
- Running the service under a local account will introduce new requirements on subscription creation. The report server stores information about the user who creates the subscription. If the user creates the subscription while logged on under a domain account, the Report Server Windows service will try to connect to a domain controller to authenticate the user when the subscription is processed. If the Report Server Windows service runs under a local account, the authentication request will fail when the report server tries to send the request to a remote domain controller. To work around this limitation, you can use a custom forms-based authentication extension or have all users connect to a report server under a local user account.
- Running the service under a local account will introduce new requirements for subscription delivery. Some delivery extensions have user account information in the subscription definition. If you are sending reports to e-mail addresses that are based on domain user accounts and you run the Report Server Windows service under a local account, it cannot access a remote domain controller to resolve the target e-mail account.

Use the following guidelines when updating the Windows service identity:

Account	Explanation
Domain accounts	<p>Windows domain accounts are the default for SQL Server services.</p> <p>Choose a domain account if the Report Server Windows service requires access to remote computers or you want to precisely configure the permissions on the service. Be sure to choose an account has minimal permissions.</p> <p>Avoid domain accounts if you do not want to change the password or account due to expiration security policies used in your organization.</p>

Network Service	<p>This is a built-in least-privilege account that has network logon permissions. It is the default account for running the Report Server Web service on Microsoft Windows Server 2003 and Windows XP Service Pack 2 (Network Service is not available on other Windows operating systems), but you can also use it to run the Windows service.</p> <p>Choose Network Service if you want to avoid any down-time that might occur as a result of password expiration policies. Avoid Network Service if you do not want to run the report server under accounts that might also be used by other applications.</p>
Local Service	<p>This is a built-in account that is similar to an authenticated user account. Services that run as the Local Service account access network resources as a null session with no credentials.</p> <p>This account is not appropriate for intranet deployment scenarios. If user authentication is performed by a network domain controller or if you want to use the service account for a remote report server database connection, choose a different account.</p> <p>Local Service is the default service account for a report server you install through SQL Server 2005 Express Edition with Advanced Services. That edition has built-in restrictions that prevent access to remote servers and subscription functionality, so the inability of the Windows service to access network resources is not an issue. For more information about report server features in the Express Edition, see Reporting Services in SQL Server 2005 Express Edition with Advanced Services.</p>
Local System	<p>Avoid this account for report server installations. Local System is a highly privileged account that is not necessary for running a report server. Choose a domain account, Network Service, or Local Service instead.</p>

If you switch the account (for example, replacing one Windows account with

another or replacing a built-in account with a Windows domain account), you will be prompted to create a backup copy of the symmetric key. The backup copy will be restored automatically once you select the new account.

 **Note:**

The Reporting Services Configuration tool prompts you to back up and restore the encryption key whenever you modify the service account. These steps are necessary for ensuring that encrypted data remains accessible to the report server. For more information about these actions, see [Encryption Keys \(Reporting Services Configuration\)](#).

▣ Options

Service Name

Specifies the Report Server Windows service account name (ReportServer).

Service Account

Specifies the account the Windows service runs as. You can only select an existing account; you cannot create new accounts in Reporting Services Configuration.

Built-in Account

Select this option to choose a built-in system account.

Windows Account

Select this option to choose a domain or local user account.

User Name

Specify a domain account in this format: <domain>\<user>. Specify a local Windows user account in this format: <computer name>\<user>.

Password

Specify the password.

Change History

Release	History
12 December 2006	Changed content: <ul style="list-style-type: none">All content has been clarified and expanded.
	New content:

17 July 2006

- Guidelines table.

▣ **See Also**

Tasks

[How to: Configure Service Accounts \(Reporting Services Configuration\)](#)

Concepts

[Configuring Service Accounts and Passwords in Reporting Services](#)

[Reporting Services Configuration F1 Help](#)

[Starting and Stopping the Report Server Windows Service](#)

Other Resources

Report Server

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Web Service Identity - Windows Server 2003 (Reporting Services Configuration)

 [See Also](#)



[Configuring a SQL Server Installation](#) > [Reporting Services Configuration](#) > [Reporting Services Configuration F1 Help](#) >

Updated: **17 July 2006**

Use this page to view or modify the service account used by the Report Server Web service for a report server that runs on Microsoft Internet Information Services (IIS) 6.0.

The Report Server Web service account is configured automatically when you create the Report Server virtual directory. The Web service runs under the security identity that is configured for the application pool of the Web site that you select. However, you can run the Web service under a different account by selecting or creating an application pool that is configured for the account you want to use.

We recommend that you always use the Reporting Services Configuration tool instead of IIS Manager to set the application pool so that settings that are affected by the change can be updated automatically. Specifically, if you are using service accounts for the report server database connection, the tool grants the new account access to the report server database whenever you change the Web service identity and updates the encryption keys so that existing encrypted data is available to the new account. If you use other tools to create or configure an application pool, the report server configuration might be incomplete.

▣ **Using a Non-default Account for Application Pools**

By default, application pools operate under Network Service, a least-privilege account with network credentials to support connections to remote servers. Depending on your deployment requirements, Network Service might not be the best choice if you want to use a domain account exclusively for report server operations. For guidance and recommendations on when to run ASP.NET applications under a domain account, see [How](#) on MSDN.

Note that if you use a domain service account and your domain is configured for Kerberos authentication, you might encounter HTTP 401 access denied errors if you did not create a Service Principal Name (SPN) that registers the domain account for the Web site. Be sure that the domain account that you register is the same one used for the application pool. For more information, see [Configuring Constrained Delegation for Kerberos \(IIS 6.0\)](#) on the Microsoft TechNet Web site.

Note:

Configuring an SPN is a global setting in IIS. If you configure an SPN, all application pools that are defined on the Web server must run under the identity of the SPN. You must be a domain administrator to set the SPN.

▣ **About Red X Indicators**

Sometimes, a red X can appear next to Web Service Identity in the navigation pane after you upgrade Reporting Services. This can occur if settings from the previous installation are still in IIS or if you are running multiple report server instances on the same computer, and you are using the same application pool for all instances.

A red X indicates that there is a discrepancy between the actual Web service identity and the Web service identity information that is stored by the Reporting Services WMI provider. This discrepancy can occur if you modify the Web service identity settings in the configuration files. The Web Service Identity page shows the actual Web service identity in **ASP.NET Service Account**. In some cases, you can synchronize settings by clicking **Apply**.

If that does not resolve the issue, choose a different application pool or click **New** to create a new application pool for the report server. You must click **Apply** after you specify the new application pool to save your changes.

📌 **Note:**

If **ASP.NET Service Account** is empty, there might be a problem with how the **<configuration>** element is specified in the report server Web.config file. If you used Visual Studio to edit the Web.config file, Visual Studio might have added a namespace to the **<configuration>** element. If the **<configuration>** element looks like `<configuration xmlns="http://schemas.microsoft.com/.NetConfiguration/v2.0">`, remove the namespace so that it looks like this: **<configuration>**.

▣ Options

Report Server

Specifies the application pool for the Report Server Web service.

Report Manager

Specifies the application pool for Report Manager. By default, Report Manager uses the same application pool as the report server.

New

Create a new application pool and specify a built-in or domain user account for the security identity.

The report server and Report Manager can run in the same application pool, but you can choose different application pools if you want to isolate them.

▣ Change History

Release	History
12 December 2006	Changed content <ul style="list-style-type: none">• Added information about how to resolve a red X indicator.
17 July 2006	New content <ul style="list-style-type: none">• Added information about how to resolve a red X indicator.• Added information about using non-default accounts.

▣ **See Also**

Tasks

[How to: Configure Service Accounts \(Reporting Services Configuration\)](#)

Concepts

[Configuring Service Accounts and Passwords in Reporting Services](#)

[Configuring a Report Server Database Connection](#)

[Configuring Report Server Virtual Directories](#)

[Reporting Services Configuration F1 Help](#)

[Administering the Report Server Web Service and Windows Service](#)

Other Resources

Introducing the Report Server Web Service

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Web Service Identity (Reporting Services Configuration)



[See Also](#)



[Configuring a SQL Server Installation](#) > [Reporting Services Configuration](#) > [Reporting Services Configuration F1 Help](#) >

Use this page to view the service account information used by the Report Server Web service. The Report Server Web service account is configured when you create the Report Server virtual directory.

This page appears when you use Internet Information Services (IIS) 5.0 or 5.01, or if you are running IIS 6.0 in isolation mode. If you are using these versions of IIS, you cannot configure the Report Server Web service identity. It always runs under the ASP.NET process identity.

▣ **See Also**

Tasks

[How to: Configure Service Accounts \(Reporting Services Configuration\)](#)

Concepts

[Configuring Report Server Virtual Directories](#)

[Reporting Services Configuration F1 Help](#)

[Configuring Service Accounts and Passwords in Reporting Services](#)

[Administering the Report Server Web Service and Windows Service](#)

Other Resources

Introducing the Report Server Web Service

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Database Setup (Reporting Services Configuration)



[See Also](#)



[Configuring a SQL Server Installation](#) > [Reporting Services Configuration](#) > [Reporting Services Configuration F1 Help](#) >

Updated: 12 December 2006

Use the Database Setup page to create and configure the report server databases that provide internal storage for one or more report server instances. Creating a report server database and configuring the connection is a multi-step process. It is important that you follow all the steps:

- Specify the SQL Server Database Engine instance and click **Connect**.
- Select, create, or upgrade a report server database. Click **New** to create a new database. To use an existing database, choose it from the list. Click **Upgrade** if the existing database is from a previous installation.
- Assign permissions and create a login used for the database connection. These steps are performed by the tool and occur when you click **Apply**. You can monitor the status of steps 2 and 3 in the Task Status pane. Be sure to scroll to the end of the messages to verify that all steps were completed successfully.

A report server database must support a specific server mode. The default mode is native mode, but you can also create a report server database for SharePoint integrated mode if you are running a report server in a larger deployment of a SharePoint product or technology. For more information, see [How to: Create a Report Server Database \(Reporting Services Configuration\)](#) and [How to: Create a Report Server Database for SharePoint Integrated Mode \(Reporting Services Configuration\)](#).

Note:

If you are using SQL Server 2005 Express Edition with Advanced Services, you must use the local SQL Server Express Database Engine instance to host the report server database. If you are creating a report server database on an instance of SQL Server that has been upgraded from version 7.0, you

might need to set the database compatibility level before you create the database. For more information, see **Reporting Services in SQL Server 2005 Express Edition with Advanced Services** and [How to: Configure a Report Server Database on a Database Engine Upgraded from SQL Server 7.0 \(Reporting Services Configuration\)](#).

▣ Options

Server Name

Specifies the name of the SQL Server Database Engine that runs the report server database.

You can use a SQL Server 2000 or SQL Server 2005 instance. You can use a default or named instance on a local or remote computer.

If you are connecting to a named instance, enter the server name in this format: *<server>\<instance>*.

If you are connecting to a remote instance, you might need to enable that instance for remote connections. Some versions of SQL Server 2005 do not enable remote connections by default. To confirm whether remote connections are allowed, use the SQL Server Surface Area Configuration tool and view the Remote Connections setting of the target instance. If the remote instance is also a named instance, verify that the SQL Server Browser service is enabled and running on the target server. SQL Server Browser provides the port number that is used by the named instance to the Reporting Services Configuration tool.

Connect

Click **Connect** to log on to the SQL Server Database Engine. The SQL Server Connection dialog box opens so that you can specify credentials used to log on to the server. You can use your credentials or specify a SQL Server login.

Database Name

Specifies the name of the report server database that stores server data. You can specify an existing database or create a new database.

Choose an existing database if you are upgrading or migrating a Reporting Services installation. Only databases that contain the report server tables are available to choose.

If you choose a report server database that was created using the SQL Server 2000 version of Reporting Services, you can upgrade the database to the new format.

Database Version

Identifies the version of the SQL Server Database Engine instance that hosts the report server database.

Server Mode

Indicates whether the report server database supports native mode or SharePoint integrated mode. The server mode that you specify determines the structure of the report server database and sets the **SharePointIntegrated** report server property to **true**. For more information, see [Deployment Modes for Reporting Services](#).

New

Click **New** to create a new database on the SQL Server instance to which you are currently connected. The SQL Server Connection dialog box opens so that you can specify credentials that have permission to create a new database on that server.

The Reporting Services Configuration tool creates two databases that are bound by name: a database to contain static data and a temporary database for storing session and work data.

Upgrade

Click **Upgrade** to migrate the tables to the new database schema. The database is upgraded in place. Upgrading the database is required if you want to use it with a SQL Server 2005 Reporting Services instance.

Note:

After you upgrade a report server database, you cannot roll it back to a previous version. Be sure that you have a backup of the database before you upgrade it.

Change

Opens the **SharePoint Database Change** dialog box so that you can create a new report server database in SharePoint integrated mode.

When you create the database, the **SharePointIntegrated** server property is set simultaneously.

Credentials Type

Specifies credentials that the report server uses to connect to the report server database. For more information about choosing credentials, see [Configuring a Report Server Database Connection](#).

Account Name

Specifies a domain user account if you are using Windows credentials or a SQL Server login if you are using SQL Server credentials. If you are using Windows credentials, specify them in this format: *<domain>\<account>*

Password

Specifies the password for the account.

Script

Click **Script** to create SQL script (.sql) files for future use in a scripted environment. You can create scripts to create and upgrade a report server database to the current schema. You can also create and run a script that configures the SQL Server database roles for the account(s) used to connect to the report server database.

Change History

Release	History
12	New content: <ul style="list-style-type: none">Description of new settings used to create a report server database in SharePoint integrated mode.

December
2006

Changed content:

- Added instructions about how to connect to a remote SQL Server instance.

▣ **See Also**

Tasks

[How to: Create a Report Server Database \(Reporting Services Configuration\)](#)

Concepts

[Upgrading a Report Server Database](#)

[Upgrading a Default Installation of Reporting Services](#)

[Reporting Services Configuration F1 Help](#)

[Configuring a Report Server Database Connection](#)

[Administering a Report Server Database](#)

[Configuring a Report Server Scale-Out Deployment](#)

Other Resources

Report Server Database

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

SharePoint Integration (Reporting Services Configuration)



[See Also](#)



[Configuring a SQL Server Installation](#) > [Reporting Services Configuration](#) > [Reporting Services Configuration F1 Help](#) >

New: **12 December 2006**

A report server can be configured to run in native mode or in SharePoint integrated mode, allowing you to synchronize report server content and operations with a SharePoint site or farm.

Use this page to view the server mode status for this report server instance. There are four possible states. Each state is indicated by a color-coded icon next to **SharePoint Integration** on the page selection pane in the Reporting Services Configuration tool:

- A gray circle indicates that SharePoint integrated mode is not supported for the current instance. Most likely, the version or edition of Reporting Services does not support SharePoint integrated mode. The version might be a pre-Service Pack 2 release of SQL Server 2005. If you are running SP2, this indicator will occur if you are running the Workgroup Edition or Express Edition with Advanced Services.
- A blue ! indicates that SharePoint integrated mode is supported on the current version and edition, but it is not configured.
- A green check indicates that the current report server instance is configured for SharePoint integration. The green check occurs when you create and then connect to a report server database in SharePoint integrated mode. The green check does not mean that the server technologies are operational. Additional steps might be required.
- A red X indicates that the Report Server Web service or Windows service cannot connect to the SharePoint content databases. Prerequisite software might not be installed, or the service accounts might not be defined as trusted accounts in SharePoint Central Administration on the target SharePoint instance.

▣ Options

There are no options on this page. This page provides status information about the deployment mode used by the current report server instance. To review the complete list of steps and tools, see [Configuring Reporting Services for SharePoint 3.0 Integration](#).

▣ **See Also**

Concepts

[Requirements for Running Reporting Services in SharePoint Integrated Mode](#)

[Deployment Modes for Reporting Services](#)

[Reporting Services and SharePoint Technology Integration](#)

[Configuring Reporting Services for SharePoint 3.0 Integration](#)

[Security Overview for Reporting Services in SharePoint Integration Mode](#)

[Storing and Synchronizing Report Server Content With SharePoint Databases](#)

[Requirements for Running Reporting Services in SharePoint Integrated Mode](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Encryption Keys (Reporting Services Configuration)



[See Also](#)



[Configuring a SQL Server Installation](#) > [Reporting Services Configuration](#) > [Reporting Services Configuration F1 Help](#) >

Updated: 12 December 2006

Use the Encryption Keys page to manage the symmetric key that is used to encrypt and decrypt data in a report server. Managing the encryption keys is an important part of report server configuration. Although the symmetric key is created and applied automatically when you create the report server database, you must create a backup copy of the symmetric key so that you can perform routine maintenance operations. The following maintenance tasks require that you have a valid copy of the symmetric key:

- Changing the service account for the Report Server Windows service.
- Migrating a Reporting Services installation to a different computer.
- Configuring a new report server instance to share or use an existing report server database.

Restoring the symmetric key is necessary if you updated the user account of the Report Server Windows service (and you used a tool other than the Reporting Services Configuration tool to change the account), or if you are migrating a report server installation to a new server.

To protect the symmetric key from unauthorized access, the symmetric key is encrypted using the private key of the Report Server Windows service. Only the Report Server Windows service is able to unlock and use the symmetric key for the purpose of storing sensitive data in the report server database. If you change the identity of the Report Server Windows service, or if you migrate the report server to a new computer, the private key of the Report Server Windows service will no longer be able to unlock the symmetric key. To restore access to the symmetric key, the symmetric key must be re-encrypted using the private key of the new Report Server Windows service identity. Restoring the symmetric key is the process by which the re-encryption occurs.

You should restore a symmetric key only if it is the same key that is currently used to encrypt and decrypt data in the report server database. If you restore a symmetric key that is not valid, you will no longer be able to access sensitive data. In this case, you must delete and recreate the key.

 **Important:**

Deleting and recreating the symmetric key is a non-reversible action that can have important ramifications on your current installation. If you delete the key, any existing data that is encrypted by the symmetric key will be deleted along with the key. Deleted data includes connection strings to external report data sources, stored connection strings, and some subscription information.

▣ Options

Backup

Copies the symmetric key to a file that you specify. The symmetric key is never stored in plain text. You must type a password to protect the file.

Restore

Applies a previously saved copy of the symmetric key to the report server database. You must provide the password to unlock the file.

The previous copy of the symmetric key for the report server instance you are currently connected to is overwritten by the restored version. After you restore the symmetric key, you must initialize all of the report servers that use the report server database. For more information about initializing report servers, see [Initializing a Report Server](#).

Change

Recreates the symmetric key and re-encrypts all encrypted values in the report server database. Be sure to stop the Report Server Windows service before recreating the symmetric key.

In a scale-out deployment, all copies of the symmetric key are replaced with newer versions. Before changing the symmetric key, be sure to review the list of servers that are joined to the scale-out deployment to verify that only valid report server instances are given access to the new key. The servers that are part of a scale-out deployment are listed in the **Initialized** page. Stop the Windows service on each report server in the deployment before recreating the key.

Note that regenerating the symmetric key can be a long-running process if you have many data sources and subscriptions.

Delete

Deletes the symmetric key and all encrypted content, including connection

strings and stored credentials. You should only delete the symmetric key if you cannot restore it.

Once you delete the symmetric key, you must re-enter the missing connection strings and stored credentials in the reports and shared data sources that no longer have these values. You must also update all subscriptions that use delivery extensions that store encrypted data. This includes the file share delivery extension and any third-party delivery extension that use encrypted value.

There is no automated way to update this information. Each report, subscription, and shared data source that uses stored credentials and connection strings must be updated one at a time.

Change History

Release	History
12 December 2006	New content: <ul style="list-style-type: none"><li data-bbox="544 1056 1284 1136">• Information regarding restoring, deleting, and recreating encryption keys.

▣ **See Also**

Concepts

[Reporting Services Configuration F1 Help](#)

[Backing Up and Restoring Encryption Keys](#)

[Deleting and Re-creating Encryption Keys](#)

[Initializing a Report Server](#)

Other Resources

Troubleshooting Initialization and Encryption Key Errors

Storing Encrypted Data in a Report Server Database

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Initialization (Reporting Services Configuration)

 [See Also](#)



[Configuring a SQL Server Installation](#) > [Reporting Services Configuration](#) > [Reporting Services Configuration F1 Help](#) >

An initialized report server describes a server that can encrypt and decrypt sensitive data that is stored in a report server database (for example, stored credentials and connection strings). Report server initialization is a requirement for report server operations.

Use this page to view the initialization status for a scale-out deployment or to join a report server to a scale-out deployment. This page does not show initialization status for a single server installation. A single server installation is initialized automatically when you set the database connection.

For a report server that is currently part of a scale-out deployment, this page shows initialization status for all of the report server instances that share the same report server database. When viewing status for a scale-out deployment, it does not matter which server you are connected to. The status information for each node will be available on this page regardless of the instance you connect to.

There are three possible initialization states for a scale-out deployment. Each state is indicated by a color-coded icon next to Initialization on the page selection pane in the Reporting Services Configuration tool:

- A grayed-out status icon appears if the database connection is not set or if you are using an edition that does not support scale-out deployment.

 **Note:**

Scale-out deployment is not supported by SQL Server 2005 Express Edition with Advanced Services, SQL Server 2005 Workgroup Edition, or SQL Server Standard Edition. For more information about feature availability in the Express edition, see **Reporting Services in SQL Server 2005 Express Edition with Advanced Services** in SQL Server Books Online.

- 
- A green check indicates that the current report server instance is initialized. The green check appears when you configure the report server database connection and the report server instance has a copy of the symmetric key in the database.
 - A red **X** indicates that a server is not initialized. A red **X** will always appear for a report server that has been removed from a scale-out deployment and does not yet have a report server database dedicated for its exclusive use. This condition will resolve itself when you create a new report server database for the report server.

A red **X** will also appear if the report server database is offline or unavailable. To verify whether this is the case, use SQL Server Management Studio to connect the Database Engine instance that hosts the report server database.

In all other cases, a red **X** indicates that the server cannot encrypt or decrypt data in the report server database. You should restore a backup copy of the symmetric key to resolve this issue. If a red **X** appeared after you changed a service account (and you did not use the Reporting Services Configuration tool to change the account), and you do not have a backup of the key, you should temporarily change the server to run under the old account so that you can obtain a backup of the key. For more information, see [Managing Encryption Keys](#) and [Configuring Service Accounts and Passwords in Reporting Services](#).

▣ Options

Machine

Shows the name of the computer that hosts the report server instance. If you upgraded a report server, and this entry is missing, you can add the computer name to the Keys table in the report server database.

Instance

Shows the report server instance name. Report server instances are based on SQL Server instances.

ID

Shows a GUID created during report server setup. You can use this value with the **rskeymgmt -n** argument to initialize a remote report server instance that is in a state where it cannot encrypt or decrypt data in the report server database.

Initialized

Indicates whether the report server is initialized.

Initialize

Click **Initialize** to configure a report server to use a symmetric key in a shared report server database. You can use **Initialize** to add a report server instance to a scale-out deployment or to troubleshoot a migration or installation problem.

To configure a scale-out deployment, select the report server instance you want to add, and then click **Initialize**. A report server instance is only available if you have previously configured a connection to the shared report server database. In addition, you must perform the initialization from a report server that is already initialized to use the report server database.

Remove

Click **Remove** to remove the encryption keys of the selected report server instance from the report server database. You can remove keys to remove a report server from a scale-out deployment or to troubleshoot a migration or installation problem. With this option, only the encryption keys for the specified report server instance are removed. Encrypted data in the report server database is not affected.

As a precaution, be sure to create a backup copy of the symmetric key before you remove it. Once you remove the encryption keys of the last report server in the list, you introduce new requirements for any subsequent report server initialization for that database. The new requirement is that after you initialize a report server you must restore a backup copy of the symmetric key. Restoring the symmetric key is necessary if you want to access the encrypted data that is currently in the report server database.

If you no longer need the encrypted data or if you do not have a backup copy of the key, you must delete the encrypted data. For more information, see [Encryption Keys \(Reporting Services Configuration\)](#).

Change History

Release	History
12 December 2006	Changed content: <ul style="list-style-type: none"><li data-bbox="667 1272 1284 1308">• Added information about status icons.

▣ **See Also**

Tasks

[How to: Configure a Report Server Scale-Out Deployment \(Reporting Services Configuration\)](#)

Concepts

[Configuring a Report Server Scale-Out Deployment](#)

[Upgrading a Report Server Web Farm](#)

[Initializing a Report Server](#)

[Managing Encryption Keys](#)

Other Resources

rskeymgmt Utility

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

E-mail Settings (Reporting Services Configuration)



[See Also](#)



[Configuring a SQL Server Installation](#) > [Reporting Services Configuration](#) > [Reporting Services Configuration F1 Help](#) >

Use this page to specify Simple Mail Transport Protocol (SMTP) settings that enable the Report Server E-Mail delivery extension. You can use Report Server E-Mail to distribute reports or report processing notifications through e-mail subscriptions. The Report Server E-Mail delivery extension requires an SMTP server and an e-mail address to use in the From: field.

Additional configuration settings can be used to further customize e-mail subscriptions, including settings that restrict mail server hosts and rendering extension availability. You cannot specify these settings in the Reporting Services Configuration tool. To configure the additional settings, you must manually edit the RSReportServer.config file. For more information, see [Configuring a Report Server for E-Mail Delivery](#) and **Reporting Services Configuration Files**.

Note:

This feature is not supported in SQL Server 2005 Express Edition with Advanced Services. For more information about feature availability in the Express edition, see **Reporting Services in SQL Server 2005 Express Edition with Advanced Services** in SQL Server Books Online.

▣ Options

Sender E-mail

Specifies the e-mail address to use in the From: field of a generated e-mail. You must specify a user account that has permission to send mail from the SMTP server.

Delivery method

Specifies that report server e-mail is routed through an SMTP server. This is the only delivery method that you can specify in the Reporting Services Configuration tool.

An alternative delivery method is to use a local SMTP service pickup directory. You can use this delivery method if a network SMTP service is not available. To specify a local SMTP service pickup directory, you must edit the RSReportServer.config file. If you edit the configuration file to use a local SMTP service pickup directory, the Reporting Services Configuration tool sets the **Delivery method** option to *custom* to indicate that the delivery method is specified in the configuration file.

In the configuration file, the delivery method is set through the **SendUsing** configuration setting. For more information about specifying the **SendUsing** setting, see [Configuring a Report Server for E-Mail Delivery](#).

SMTP Server

Select the SMTP server or gateway to use. You can use a local server or an SMTP server on your network.

▣ **See Also**

Concepts

[Configuring a Report Server for E-Mail Delivery](#)
[Reporting Services Configuration F1 Help](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Execution Account (Reporting Services Configuration)



[See Also](#)



[Configuring a SQL Server Installation](#) > [Reporting Services Configuration](#) > [Reporting Services Configuration F1 Help](#) >

Updated: 17 July 2006

Use this page to configure an account to use for unattended processing. This account is used under special circumstances when other sources of credentials are not available:

- When the report server connects to a data source that does not require credentials. Examples of data sources that might not require credentials include XML documents and some client-side database applications.
- When the report server connects to another server to retrieve external image files or other resources that are referenced in a report.

Setting this account is optional, but not setting it limits your use of external images and connections to some data sources. When retrieving external image files, the report server checks to see if an anonymous connection can be made. If the connection is password protected, the report server uses the unattended report processing account to connect to the remote server. When retrieving data for a report, the report server either impersonates the current user, prompts the user to provide credentials, uses stored credentials, or uses the unattended processing account if the data source connection specifies **None** as the credential type. The report server does not allow its service account credentials to be delegated or impersonated when connecting to other computers, so it must use the unattended processing account if no other credentials are available.

The account you specify should be different from the one used to run the service accounts. If you configure this account, it is stored in the RSReportServer.config file as an encrypted value. If you are running the report server in a scale-out deployment, you must configure this account on each report server.

You can use any Windows user account. For best results, choose an account that has read permissions and network logon permissions to support connections to

other computers. It must have read permissions on any external image or data file that you want to use in a report. Do not specify a local account unless all report data sources and external images are stored on the report server computer. Use the account only for unattended report processing.

 **Note:**

If you are using SQL Server 2005 Express Edition with Advanced Services, you only need to configure this account if you are referencing external images in a report and permission is required to access the image file. The Express Edition does not support a data source connection to a remote server. For more information, see **Adding an Image to a Report**.

▣ Options

Specify an execution account

Click this checkbox to specify an account.

Account

Specifies a Windows domain user account. Use this format to specify the account: <domain>\<user account>.

Password

Type the password.

Verify password

Retype the password.

Change History

Release	History
17 July 2006	Changed content: <ul style="list-style-type: none">• Significant revision.

▣ **See Also**

Concepts

[Reporting Services Configuration F1 Help](#)

[Configuring an Account for Unattended Report Processing](#)

Other Resources

RSReportServer Configuration File

Storing Encrypted Report Server Data

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Network Protocols and Network Libraries

 [See Also](#)



[Configuring a SQL Server Installation](#) >

Updated: **5 December 2005**

A server can listen on, or monitor, multiple network protocols at one time. However, each protocol must be configured. If a particular protocol is not configured, the server cannot listen on that protocol. After installation, you can change the protocol configurations using the SQL Server Configuration Manager.

▣ **Default SQL Server Network Configuration**

A default instance of SQL Server 2005 is configured for TCP/IP port 1433, and named pipe `\\.\pipe\sql\query`. SQL Server 2005 named instances are configured for TCP dynamic ports, with a port number assigned by the operating system.

If you cannot use dynamic port addresses (for example, when SQL Server connections must pass through a firewall server configured to pass through specific port addresses, or when some connections are made using the client components from SQL Server version 7.0 or earlier), we recommend using a port address less than 1024. Choose a port in this range that is not used by the operating system or another application.

To enhance security, network connectivity is not fully enabled when SQL Server 2005 is installed. To enable, disable, and configure network protocols after Setup is complete, use the SQL Server 2005 Surface Area Configuration tool, or the SQL Server 2005 Network Configuration area of the SQL Server Configuration Manager.

▣ **Types of Endpoints**

SQL Server 2005 introduces a new concept for SQL Server connections; the connection is represented on the server end by a Transact-SQL *endpoint*. Permissions can be granted, revoked, and denied for Transact-SQL endpoints. By default, all users have permissions to access an endpoint unless the permissions are denied or revoked by a member of the sysadmin group or by the endpoint owner. The GRANT, REVOKE, and DENY ENDPOINT syntax uses an endpoint ID that the administrator must get from the endpoint's catalog view.

SQL Server Setup creates Transact-SQL endpoints for all supported network protocols, as well as for the dedicated administrator connection. See the table below for default network configuration settings.

Transact-SQL endpoints created by SQL Server Setup are as follows:

- TSQL local machine
- TSQL named pipes
- TSQL default TCP
- TSQL default VIA

For more information on managing Transact-SQL endpoints, see "Network Protocols and TDS Endpoints" in SQL Server 2005 Books Online.

□ Default Configuration

The following table describes default network configuration settings.

SQL Server edition	New installation or upgrade	Shared memory	TCP/IP	Named pipes	VIA
Enterprise	New	Enabled	Enabled	Disabled (Local only)	Disabled
Developer	New	Enabled	Disabled	Disabled (Local only)	Disabled
Standard	New	Enabled	Enabled	Disabled (Local only)	Disabled
Workgroup	New	Enabled	Enabled	Disabled (Local only)	Disabled
Evaluation	New	Enabled	Disabled	Disabled (Local only)	Disabled
SQL Server Express	New	Enabled	Disabled	Disabled (Local only)	Disabled
All editions	Upgrade or side-by-side installation	Enabled	Settings preserved from the previous installation	Settings preserved from the previous installation	Disabled

For failover cluster installations, note the following additional information:

- If the instance is running on a SQL Server 2005 failover cluster, it will listen on the ports on each IP address selected during SQL Server Setup. To enable additional network protocols, use either the Surface Area Configuration Tool, or the SQL Server Configuration Manager.

When installing SQL Server 2005 from the command prompt, you can specify the startup type of the SQL Server Browser service. SQL Server Setup will install SQL Server Browser disabled and stopped by default. To configure the service to start automatically, see the `DISABLENETWORKPROTOCOLS`

parameter in [How to: Install SQL Server 2005 from the Command Prompt](#).

For more information on managing SQL Server network configurations, see the following topics in SQL Server 2005 Books Online:

- "Server Network Configuration"
- "Configuring Server Network Protocols and Net-Libraries"
- "Default SQL Server Network Configuration"
- "Net-Libraries and Network Protocols"

▣ **See Also**

Concepts

[Preparing to Install SQL Server 2005](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

File Locations for Default and Named Instances of SQL Server 2005



[See Also](#)



[Configuring a SQL Server Installation](#) >

Updated: 5 December 2005

An installation of Microsoft SQL Server 2005 consists of one or more separate instances. An instance, whether default or named, has its own set of program and data files, as well as a set of common files shared between all instances on the computer.

For an instance of SQL Server that includes the Database Engine, Analysis Services, and Reporting Services, each component has a full set of data and executable files, and common files shared by all components.

To isolate install locations for each component, unique instance IDs are generated for each component within a given SQL Server instance. This allows upgrade of a single component to future SQL Server versions without dependencies on other SQL Server components.

Important:

Do not delete any of the following directories or their contents: Binn, Data, Ftdata, HTML, or 1033. You may delete other directories, if necessary; however, you may not be able to retrieve any lost functionality or data without uninstalling and then reinstalling SQL Server 2005.

Note:

Do not delete or modify any of the .htm files in the HTML directory. They are required for SQL Server tools to function properly.

▣ **Shared Files for All Instances of SQL Server 2005**

Common files used by all instances on a single computer are installed in the folder *systemdrive*:\Program Files\Microsoft SQL Server\90, where *systemdrive* is the drive letter where components are installed. Normally this is drive C.

 **Note:**

Program files and data files cannot be installed on a removable disk drive, cannot be installed on a file system that uses compression, and cannot be installed on shared drives on a failover cluster instance.

▣ File Locations and Registry Mapping

During SQL Server Setup, an instance ID is generated for each server component. The server components in this SQL Server release are the Database Engine, Analysis Services, and Reporting Services. The instance ID is in the format MSSQL.n, where n is the ordinal number of the component being installed. The instance ID is used in the file directory and the registry root.

The first instance ID generated is MSSQL.1; ID numbers are incremented for additional instances as MSSQL.2, MSSQL.3, and so on. If gaps occur in the ID sequence due to uninstalls, ID numbers are generated to fill them. The most recently installed instance may not always have the highest instance ID number.

Server components are installed in directories with the format <instanceID>\<component name>. For example, a default or named instance with the Database Engine, Analysis Services, and Reporting Services would have the following default directories:

- <Program Files>\Microsoft SQL Server\MSSQL.1\MSSQL\ for the Database Engine
- <Program Files>\Microsoft SQL Server\MSSQL.2\OLAP\ for Analysis Services
- <Program Files>\Microsoft SQL Server\MSSQL.3\RS\ for Reporting Services

Instead of <Program Files>\Microsoft SQL Server, a <custom path> is used if the user chooses to change the default installation directory.

▣ **Note:**

SQL Server 2005 Integration Services, Notification Services, and client components are not instance aware and, therefore, are not assigned an instance ID. Non-instance-aware components are installed to the same directory by default: <system drive>:\Program Files\Microsoft SQL Server\90\. Changing the installation path for one shared component also changes it for the other shared components. Subsequent installations install non-instance-aware components to the same directory as the original installation.

The registry hive is created under HKLM\Software\Microsoft\Microsoft SQL Server\MSSQL.n for instance-aware components. For example,

- HKLM\Software\Microsoft\Microsoft SQL Server\MSSQL.1
- HKLM\Software\Microsoft\Microsoft SQL Server\MSSQL.2
- HKLM\Software\Microsoft\Microsoft SQL Server\MSSQL.3

The registry also maintains a mapping of instance ID to instance name. Instance ID to instance name mapping is maintained as follows:

- [HKEY_LOCAL_MACHINE\Software\Microsoft\Microsoft SQL Server\Instance Names\SQL] "InstanceName"="MSSQL.1"
- [HKEY_LOCAL_MACHINE\Software\Microsoft\Microsoft SQL Server\Instance Names\OLAP] "InstanceName"="MSSQL.2"
- [HKEY_LOCAL_MACHINE\Software\Microsoft\Microsoft SQL Server\Instance Names\RS] "InstanceName"="MSSQL.3"

▣ **AdventureWorks Sample Databases (new)**

AdventureWorks samples are installed in the Tools\Samples directory. For unattended installations, if you want the sample databases to be attached to a specific SQL Server instance, use the SAMPLEDATABASESERVER=\<instancename> switch. The instance to which the sample database is attached must be a local instance.

For more information on unattended installations, see [Remote Setup Information for SQL Server 2005](#), [How to: Install SQL Server 2005 from the Command Prompt](#), or instructions included with the Template.ini file on the SQL Server DVD.

▣ Specifying File Paths

During Setup, you can change the installation path for the following features:

The installation path is displayed in Setup only for features with a user-configurable destination folder:

Component	Default path^{1, 2}	Configurable³ or Fixed Path
Database Engine server components	\Program Files\Microsoft SQL Server\MSSQL.n\MSSQL\Binn\	Configurable
Database Engine data files	\Program Files\Microsoft SQL Server\MSSQL.n\MSSQL\Data\	Configurable
Analysis Services server	\Program Files\Microsoft SQL Server\MSSQL.n\OLAP\Bin\	Configurable
Analysis Services data files	\Program Files\Microsoft SQL Server\MSSQL.n\OLAP\Data\	Configurable
Reporting Services report server	\Program Files\Microsoft SQL Server\MSSQL.n\Reporting Services\ReportServer\Bin\	Configurable
Reporting Services report manager	\Program Files\Microsoft SQL Server\MSSQL.n\Reporting Services\ReportManager\Bin\	Fixed path
SQL Server Integration Services	<Install Directory>\90\DTS\	Configurable ⁴
Notification Services	<Install Directory>\90\Notification Services\	Configurable ⁴
Client Components	<Install Directory>\90\Tools\	Configurable ⁴
Replication and server-side COM objects	Program Files\Microsoft SQL Server\90\COM ⁵	Fixed path
SQL Server Integration Services component DLL's for the Data Transformation Run-time engine, the Data	Program Files\Microsoft SQL Server\90\DTS\Binn	Fixed path

Transformation Pipeline engine, and the dtexec command prompt utility		
DLL's that provide managed connection support for SQL Server Integration Services	Program Files\Microsoft SQL Server\90\DTS\Connections	Fixed path
DLL's for each type of enumerator that SQL Server integration Services supports	Program Files\Microsoft SQL Server\90\DTS\ForEachEnumerators	Fixed path
SQL Server Browser Service, WMI providers	\Program Files\Microsoft SQL Server\90\Shared\	Fixed path
Components that are shared between all instances of SQL Server 2005	\Program Files\Microsoft SQL Server\90\Shared\	Fixed path

¹Make sure that the \Program Files\Microsoft SQL Server\ folder is protected with limited permissions.

²The default drive for these locations is *systemdrive*, normally drive C.

³Installation paths for child features are determined by the installation path of the parent feature.

⁴A single installation path is shared between SQL Server Integration Services, Notification Services, and Client Components. Changing the installation path for one component also changes it for other components. Subsequent installations install components to the same location as the original installation.

⁵This directory is used by all instances of SQL Server on a computer. If you apply an update, like a service pack, to any of the instances on the computer, any changes to files in this folder will affect all instances on the computer.

 **Note:**

For clustered configurations, you must select a local drive that is available on every node of the cluster.

 **Note:**

When adding features to an existing installation, you cannot change the location of a previously installed feature, nor can you specify the location for a new feature. You must either install additional features to the directories already established by Setup, or uninstall and reinstall the product.

When you specify an installation path during Setup for the server components or data files, the Setup program uses the instance name in addition to the specified location for program and data files. Setup does not use the instance name for tools and other shared files. Setup also does not use any instance name for the Analysis Services program and data files, although it does use the instance name for the Analysis Services repository.

If you set an installation path for the Database Engine feature, SQL Server Setup uses that path as the root directory for all instance-specific folders for that installation, including SQL Data Files. In this case, if you set the root to "C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\," instance-specific directories are added to the end of that path.

Customers choosing to use the USESYSDB upgrade functionality in the SQL Server Installation Wizard (Setup UI mode) can easily lead themselves into a situation where the product gets installed into a recursive folder structure - for example, <SQLProgramFiles>\MSSQL.1\MSSQL\MSSQL.1\MSSQL\Data\. Instead, to use the USESYSDB feature, set an installation path for the SQL Data Files feature instead of the Database Engine feature.

 **Note:**

Data files are always expected to be found in a child directory named Data. For example, specify C:\Program Files\Microsoft SQL Server\MSSQL.1\ to specify the root path to the data directory of the system databases during upgrade when data files are found under C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Data.

▣ **See Also**

Reference

[Feature Selection](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Changing Passwords and User Accounts



[See Also](#)



[Configuring a SQL Server Installation](#) >

Updated: **17 July 2006**

Microsoft SQL Server 2005 services accounts and passwords are linked to Microsoft Windows user accounts and passwords. Changes in one location may require changes in the other.

▣ **Changing SQL Server Services Accounts After Installation**

After you have installed SQL Server 2005, use SQL Server Configuration Manager to change the assigned password or other properties of any SQL Server–related service. Each service must be changed individually. The new user account takes effect when the service is restarted.

Important:

You should not change the passwords for any of the SQL Server service accounts when a failover cluster node is down or offline. If you have to do this, you will need to reset the password again using Configuration Manager when all nodes are back online.

The following rights are granted to the accounts:

- **SeServiceLogonRight**, which allows the account to run as a service.
- **SeLockMemoryPrivilege**, which allows the account to use the AWE memory feature of SQL Server.
- **SeTcbPrivilege**, which allows the account to impersonate other accounts.

If you are running SQL Server in a failover cluster configuration, permissions are also set for all files in the binary and data installation locations for all nodes in the cluster. Permission is also granted for the service account on the Cluster Object.

Caution:

If you are running Microsoft Windows 2000 and want to use the Windows 2000 Encrypted File System to encrypt any SQL Server files, you must unencrypt the files before you can change the SQL Server service accounts. If you do not unencrypt the files and then reset the SQL Server service accounts, you cannot unencrypt the files.

Changing the current service account for SQL Server to a non-administrator account causes existing full-text catalogs to become inaccessible. Either rebuild and perform a full population of all catalogs belonging to this instance of SQL

Server, or switch back to an account with administrator permissions.

You can change the MSSQLSERVER service account to a non Windows NT 4.0 administrator account. However, the Windows NT 4.0 account must be a member of the **sysadmin** fixed server role to run SQL Server Agent.

For more information, see [Setting Up Windows Service Accounts](#). For information about using the Services add-in for Windows to change SQL Server service accounts, see [How to change the SQL Server or SQL Server Agent service account without using SQL Enterprise Manager in SQL Server 2000 or SQL Server Configuration Manager in SQL Server 2005](#).

▣ **Windows Passwords Changes**

If your Windows password changes after SQL Server 2005 is installed – e.g., your password expires - you must also revise the user account information for SQL Server services in Windows.

To change SQL Server services login account information (Windows 2000)

After changing the SQL Server service account information in Control Panel, you must also change the SQL Server service account in SQL Server Configuration Manager. This allows the service account information for Microsoft Search service to remain synchronized as well.

Security Note:

Setting strong passwords is essential to the security of your system. Always use strong passwords.

Important:

Although the Microsoft Search service is always assigned to the local system account, the full-text search engine tracks the SQL Server service account in Windows. Full-text search and failover clustering are not available if Windows password changes are not reset.

For more information about creating Windows user accounts, granting advanced user rights, setting password expiration, and managing group memberships, see the Windows documentation or User Manager for Domains Help. For Microsoft Windows 2000 users, see Computer Management or Group Policy Editor in the Windows 2000 documentation.

▣ Change History

Release	History
17 July 2006	Changed content: <ul style="list-style-type: none">• Added a link to the knowledge base article for using the Services add-in to change SQL Server and SQL Server Agent service accounts.

▣ **See Also**

Concepts

[SQL Server Surface Area Configuration](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

System and Sample Databases

 [See Also](#)



[Configuring a SQL Server Installation](#) >

Updated: 5 December 2005

When SQL Server 2005 is installed, Setup creates the database and log files shown in the following table.

Database	Database file	Log file
master	Master.mdf	Mastlog.ldf
model	Model.mdf	Modellog.ldf
msdb	Msdbdata.mdf	Msdblog.ldf
tempdb	Tempdb.mdf	Templog.ldf

The system databases are **master**, **model**, **msdb**, and **tempdb**.

 **Note:**

The default location of the database and log files is Program Files\Microsoft SQL Server\Mssql.n\MSSQL\Data, where **n** is the ordinal number of the SQL Server instance. This location may vary if the default location was changed when SQL Server was installed.

▣ **AdventureWorks Sample Databases**

The sample databases are provided as learning tools, but are not installed by default in the SQL Server 2005 release. For more information on **AdventureWorks** sample databases, see [Running Setup to Install AdventureWorks Sample Databases and Samples](#).

▣ **See Also**

Other Resources

AdventureWorks Sample Database (Transact-SQL)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Product Specifications for SQL Server 2005



[Configuring a SQL Server Installation](#) >

The following topics provide product specification details for this release.

Topic	Description
Editions and Components of SQL Server 2005	Selecting components and features to install.
Features Supported by the Editions of SQL Server 2005	Support for components and features in each SQL Server edition.
Maximum Capacity Specifications for SQL Server 2005	Maximum capacities for various SQL Server objects.
Maximum Number of Processors Supported by the Editions of SQL Server 2005	The maximum number of processors supported by each edition of SQL Server 2005.

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Maximum Capacity Specifications for SQL Server 2005



[See Also](#)



[Configuring a SQL Server Installation](#) > [Product Specifications for SQL Server 2005](#) >

Updated: **12 December 2006**

The following tables specify the maximum sizes and numbers of various objects defined in Microsoft SQL Server 2005 components.

Database Engine Objects

The following table specifies the maximum sizes and numbers of various objects defined in SQL Server 2005 databases or referenced in Transact-SQL statements. The table does not include SQL Server Windows CE Edition.

SQL Server 2005 Database Engine object	Maximum sizes/numbers SQL Server 2005 (32-bit)	Maximum sizes/numbers SQL Server 2005 (64-bit)
Batch size ¹	65,536 * Network Packet Size	65,536 * Network Packet Size
Bytes per short string column	8,000	8,000
Bytes per GROUP BY, ORDER BY	8,060	8,060
Bytes per index key ²	900	900
Bytes per foreign key	900	900
Bytes per primary key	900	900
Bytes per row ⁸	8,060	8,060
Bytes per varchar(max) , varbinary(max) , xml , text , or image column	2 ³¹ -1	2 ³¹ -1
Characters per ntext or nvarchar(max) column	2 ³⁰ -1	2 ³⁰ -1
Clustered indexes per table	1	1
Columns in GROUP BY, ORDER BY	Limited only by number of bytes	Limited only by number of bytes
Columns or expressions in a		

GROUP BY WITH CUBE or WITH ROLLUP statement	10	10
Columns per index key ⁷	16	16
Columns per foreign key	16	16
Columns per primary key	16	16
Columns per base table	1,024	1,024
Columns per SELECT statement	4,096	4,096
Columns per INSERT statement	1,024	1,024
Connections per client	Maximum value of configured connections	Maximum value of configured connections
Database size	1,048,516 terabytes	1,048,516 terabytes
Databases per instance of SQL Server	32,767	32,767
Filegroups per database	32,767	32,767
Files per database	32,767	32,767
File size (data)	16 terabytes	16 terabytes
File size (log)	2 terabytes	2 terabytes
Foreign key table references per table ⁴	253	253
Identifier length (in characters)	128	128
Instances per computer	50 instances on a stand-alone server for all SQL Server 2005 editions except for Workgroup Edition. Workgroup Edition supports a maximum of 16 instances.	50 instances on a stand-alone server. 25 instances

	SQL Server 2005 supports 25 instances on a failover cluster.	on a failover cluster.
Length of a string containing SQL statements (batch size) ¹	65,536 * Network packet size	65,536 * Network packet size
Locks per connection	Maximum locks per server	Maximum locks per server
Locks per instance of SQL Server ⁵	Up to 2,147,483,647	Limited only by memory
Nested stored procedure levels ⁶	32	32
Nested subqueries	32	32
Nested trigger levels	32	32
Nonclustered indexes per table	249	249
Parameters per stored procedure	2,100	2,100
Parameters per user-defined function	2,100	2,100
REFERENCES per table	253	253
Rows per table	Limited by available storage	Limited by available storage
Tables per database ³	Limited by number of objects in a database	Limited by number of objects in a database
Partitions per partitioned table or index	1,000	1,000
Statistics on non-indexed columns	2,000	2,000
Tables per SELECT statement	256	256

Triggers per table ³	Limited by number of objects in a database	Limited by number of objects in a database
UNIQUE indexes or constraints per table	249 nonclustered and 1 clustered	249 nonclustered and 1 clustered
User connections	32,767	32,767
XML indexes	249	249

¹Network Packet Size is the size of the tabular data stream (TDS) packets used to communicate between applications and the relational Database Engine. The default packet size is 4 kilobytes (KB), and is controlled by the **network packet size** configuration option.

²The maximum number of bytes in any index key cannot exceed 900 in SQL Server 2005. You can define a key using variable-length columns whose maximum sizes add up to more than 900, provided no row is ever inserted with more than 900 bytes of data in those columns. In SQL Server 2005, you can include nonkey columns in a nonclustered index to avoid the maximum index key size of 900 bytes. For more information, see **Index with Included Columns**.

³Database objects include objects such as tables, views, stored procedures, user-defined functions, triggers, rules, defaults, and constraints. The sum of the number of all objects in a database cannot exceed 2,147,483,647.

⁴Although a table can contain an unlimited number of FOREIGN KEY constraints, the recommended maximum is 253. Depending on the hardware configuration hosting SQL Server, specifying additional foreign key constraints may be expensive for the query optimizer to process.

⁵This value is for static lock allocation. Dynamic locks are limited only by memory.

⁶If a stored procedure accesses more than 8 databases, or more than 2 databases in interleaving, you will receive an error.

⁷If the table contains one or more XML indexes, the clustering key of the user table is limited to 15 columns because the XML column is added to the clustering key of the primary XML index. In SQL Server 2005, you can include nonkey columns in a nonclustered index to avoid the limitation of a maximum of 16 key columns. For more information, see **Index with Included Columns**.

⁸SQL Server 2005 supports row-overflow storage which enables variable length columns to be pushed off-row. Only a 24-byte root is stored in the main record for variable length columns pushed out of row; because of this, the effective row limit is higher than in previous releases of SQL Server. For more information, see the "Row-Overflow Data Exceeding 8 KB" topic in SQL Server 2005 Books Online.

▣ Replication Objects

The following table specifies the maximum sizes and numbers of various objects defined in SQL Server 2005 Replication. The table does not include SQL Server Windows CE Edition.

SQL Server 2005 Replication object	Maximum sizes/numbers SQL Server 2005 (32-bit)	Maximum sizes/numbers SQL Server 2005 (64-bit)
Articles (merge publication)	256	256
Articles (snapshot or transactional publication)	32,767	32,767
Columns in a table ¹ (merge publication)	246	246
Columns in a table ² (SQL Server snapshot or transactional publication)	1000	1000
Columns in a table ² (Oracle snapshot or transactional publication)	995	995
Bytes for a column used in a row filter (merge publication)	1024	1024
Bytes for a column used in a row filter (snapshot or transactional publication)	8000	8000

¹If row tracking is used for conflict detection (the default), the base table can include a maximum of 1,024 columns, but columns must be filtered from the article so that a maximum of 246 columns is published. If column tracking is used, the base table can include a maximum of 246 columns. For more information on the tracking level, see the "Tracking Level" section of **How Merge Replication Detects and Resolves Conflicts**.

²The base table can include the maximum number of columns allowable in the publication database (1,024 for SQL Server), but columns must be filtered from

the article if they exceed the maximum specified for the publication type.

▣ Change History

Release	History
12 December 2006	Changed content: <ul style="list-style-type: none">• Removed the restriction under Database Engine Objects about bytes in source text of a stored procedure.• Updated the number of instances per computer.
17 July 2006	Changed content: <ul style="list-style-type: none">• Updated the number of instances per computer.

▣ **See Also**

Reference

[Hardware and Software Requirements for Installing SQL Server 2005](#)
[Check Parameters for the System Configuration Checker](#)

Other Resources

[Installing SQL Server 2005 Components](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Maximum Number of Processors Supported by the Editions of SQL Server 2005

 [See Also](#)



[Configuring a SQL Server Installation](#) > [Product Specifications for SQL Server 2005](#) >

The following table specifies the maximum number of processors supported by each edition of SQL Server 2005.

SQL Server 2005 edition	Number of processors supported (32-bit)	Number of processors supported (64-bit)
Enterprise Edition	Operating system maximum ¹	Operating system maximum ¹
Developer Edition	Operating system maximum ¹	Operating system maximum ¹
Standard Edition	4	4
Workgroup Edition	2	N/A ²
SQL Server Express Edition	1	N/A ²
Evaluation Edition	Operating system maximum ¹	Operating system maximum ¹

¹These editions of SQL Server 2005 support the maximum number of processors supported by the operating system.

²This edition of SQL Server 2005 is not available for the 64-bit platform in this release.

▣ Change History

Release	History
17 July 2006	Changed content: <ul style="list-style-type: none"><li data-bbox="435 499 1380 583">• Updated support for Standard Edition, which supports four processors, not the operating system maximum.

▣ **See Also**

Reference

[Memory Supported by the Editions of SQL Server 2005](#)

[Hardware and Software Requirements for Installing SQL Server 2005](#)

Other Resources

[Product Specifications for SQL Server 2005](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Memory Supported by the Editions of SQL Server 2005

 [See Also](#)



[Configuring a SQL Server Installation](#) > [Product Specifications for SQL Server 2005](#) >

The following table specifies the maximum memory support for each edition of Microsoft SQL Server 2005.

SQL Server 2005 edition	Maximum memory supported (32-bit)	Maximum memory supported (64-bit)
Enterprise Edition	OS maximum ¹	OS maximum ¹
Developer Edition	OS maximum	32 TB
Standard Edition	OS maximum	32 TB
Workgroup Edition	3 GB	N/A ²
SQL Server Express Edition	1 GB	N/A ²
Evaluation Edition	OS maximum	OS maximum

¹SQL Server 2005 Enterprise Edition will support the maximum memory supported by the operating system.

²This edition of SQL Server 2005 is not available for the 64-bit platform in this release.

▣ **See Also**

Reference

[Hardware and Software Requirements for Installing SQL Server 2005](#)

Other Resources

[Product Specifications for SQL Server 2005](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Setup User Interface Reference



[See Also](#)



This section describes the installation options presented by the Microsoft SQL Server Installation Wizard. Options include installing SQL Server 2005 components or upgrading an existing SQL Server instance to SQL Server 2005.

Changing or removing an existing SQL Server 2005 instance is provided through **Add or Remove Programs** in Control Panel. For more information on instance maintenance, see [How to: Change Components in an Existing Instance of SQL Server 2005 \(Setup\)](#). For more information on instance removal, see [How to: Uninstall an Existing Instance of SQL Server 2005 \(Setup\)](#).

▣ **See Also**

Tasks

[How to: Install SQL Server 2005 \(Setup\)](#)

Concepts

[Preparing to Install SQL Server 2005](#)

[Reserved Keywords \(Setup\)](#)

Other Resources

[Upgrading to SQL Server 2005](#)

[Configuring High Availability](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Installing Prerequisites



[See Also](#)



[Setup User Interface Reference](#) >

In order for Microsoft SQL Server to install successfully, the following software components are required on the installation computer. They will be installed by SQL Server Setup:

- Microsoft .NET Framework Version 2.0
- Microsoft SQL Server Native Client
- Microsoft SQL Server 2005 Setup Support Files



Note:

These components are not removed if Setup fails or is cancelled. To remove them, use **Add or Remove Programs** in Control Panel.

▣ **See Also**

Tasks

[How to: Install SQL Server 2005 \(Setup\)](#)

Reference

[Hardware and Software Requirements for Installing SQL Server 2005](#)
[Check Parameters for the System Configuration Checker](#)

Concepts

[Security Considerations for a SQL Server Installation](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

System Configuration Check (SCC)



[See Also](#)



[Setup User Interface Reference](#) >

SQL Server Express users: click [System Configuration Check \(SQL Server Express\)](#) for information that is specific to SQL Server Express Setup.

The **System Configuration Check (SCC)** page of the Microsoft SQL Server Installation Wizard shows progress of the system preparation tool. System Configuration Check (SCC) routines scan the destination computer for conditions that may block Setup. For clusters, SCC runs on all nodes of the virtual server.

Users can click **Stop** at any time to cancel SCC and exit the Installation Wizard.

Progress symbols are as follows:

- Arrow symbol = Check in progress
- Check symbol = Check item successful
- X symbol = Check item failed

Once the SCC check is complete, the **Report** button becomes available. The **Continue** button is available only if all check results are successful, or if failed checks are non-fatal. For failed check items, resolution to blocking issues is included with results in the SCC report.

The **Report** button at the bottom of the SCC page allows you to view the report, save it as a file, copy it to the clipboard, or send it as e-mail. Information presented in the report includes:

- A list of configuration items checked on the destination computer: requirements for the current operating system, the operating system service pack level, SQL Server edition compatibility with the current operating system, and Internet Information Services (IIS).
- The result of each check item.
- The recommended user action to remove blocking issues.

▣ **See Also**

Tasks

[How to: Install SQL Server 2005 \(Setup\)](#)

Reference

[Check Parameters for the System Configuration Checker](#)

[Hardware and Software Requirements for Installing SQL Server 2005](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help Registration Information



[See Also](#)



[Setup User Interface Reference](#) >

SQL Server Express users: click [Registration Information \(SQL Server Express\)](#) for information that is specific to SQL Server Express Setup.

Use the **Registration Information** page of the Microsoft SQL Server Installation Wizard to enter identification information used to register your copy of SQL Server. It includes the User Name, Company Name, and Product Key. The Product Key must be entered for the first installation only; for subsequent installations, it will display automatically.

▣ **Options**

Name

Type your name. When installing SQL Server on a network server, provide the name of a user responsible for administering the server. A user name is required to continue Setup.

Company

Enter the name of your company. For a personal copy of the product, you can provide your name again. Information for this field is optional.

Product Key

Enter the Product Key information from the sticker on the SQL Server 2005 DVD sleeve.

▣ **See Also**

Tasks

[How to: Install SQL Server 2005 \(Setup\)](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Components to Install



[_See Also](#)



[Setup User Interface Reference](#) >

Updated: **14 April 2006**

Use the check boxes on the **Components to Install** page of the Microsoft SQL Server Installation Wizard to select component groups for your SQL Server installation.

Options

A description for each component group appears in the **Components to be Installed pane** when you select it. You can select any combination of check boxes.

To select individual features for your installation, click **Advanced**. To install components to a custom destination folder or to see the disk space requirements for an individual component or feature, click **Advanced**.

Select this component group	To install these components and features
SQL Server Database Services	<p>SQL Server Database Engine includes the following technologies:</p> <ul style="list-style-type: none">• The Database Engine is the core service for storing, processing, and securing data.• Replication is a set of technologies for copying and distributing data and database objects from one database to another and then synchronizing between databases to maintain consistency.• Full-Text Search provides functionality to issue full-text queries against plain character-based data in SQL Server tables.• Tools for managing relational and XML data.
Analysis Services	<p>Analysis Services includes the tools for creating and managing online analytical processing (OLAP) and data mining applications.</p>
Reporting Services ^{1,2,3,4}	<p>Reporting Services includes server and client components for creating, managing, and deploying tabular, matrix, graphical, and free-form reports. Reporting Services is also an extensible platform that you can use to develop report applications.</p>
Notification Services	<p>Notification Services is a platform for developing and deploying applications that send personalized, timely information to subscribers on a variety of devices.</p>

Integration Services	Integration Services is a set of graphical tools and programmable objects for moving, copying, and transforming data.
----------------------	---

	Installs components for communication between clients and servers, including network libraries for DB-Library, OLEDB for OLAP, ODBC, ADODB, and ADOMD+.
--	---

	Management Tools
--	------------------

- | | |
|--|--|
| Workstation Components, Books Online, and Development Tools ^{5,6} | <ul style="list-style-type: none">• SQL Server Management Studio (SSMS), new in Microsoft SQL Server 2005, is an integrated environment for accessing, configuring, managing, administering, and developing all components of SQL Server. SSMS combines the features of Enterprise Manager, Query Analyzer, and Analysis Manager, included in previous releases of SQL Server, into a single environment that provides SQL Server access to developers and administrators of all skill levels.• SQL Server Configuration Manager provides basic configuration management for SQL Server services, server protocols, client protocols, and client aliases.• SQL Server Profiler provides a graphical user interface for monitoring an instance of the Database Engine or an instance of Analysis Services.• Database Engine Tuning Advisor helps create optimal sets of indexes, indexed views, and partitions.• Replication Monitor allows you to track the status and performance of publications and subscriptions across a replication topology.• SQLXML Client Features |
|--|--|

	Documentation
--	---------------

- | | |
|--|--|
| | <ul style="list-style-type: none">• SQL Server Books Online is the core documentation for SQL Server 2005.• Software development kits |
|--|--|

	Development Tools
--	-------------------

- | | |
|--|--|
| | <ul style="list-style-type: none">• The Business Intelligence Development Studio is an integrated development environment for Analysis Services, |
|--|--|

¹Setup does not configure load-balancing and single-URL addressing for multiple nodes in a Report Server scale-out deployment. To complete a scale-out deployment, you must use Windows Server, Microsoft Application Center, or third-party cluster management software. For more information about setting up Web farm deployment, see [Configuring a Report Server Scale-Out Deployment](#).

²Reporting Services is not available if Internet Information Services (IIS) 5.0 or higher is not installed. Microsoft Internet Explorer 6.0 Service Pack (SP) 1 is required for the Report Designer component of Reporting Services.

³For Reporting Services (64-bit) installations on 64-bit servers, the 64-bit version of ASP.NET must be installed. For Reporting Services (32-bit) installations on 32-bit systems, and on the 32-bit subsystem (WOW64) of a 64-bit server, the 32-bit version of ASP.NET must be installed.

⁴Reporting Services is not supported in side-by-side configurations on the 64-bit platform and on the 32-bit subsystem (WOW64) of a 64-bit server at the same time.

⁵Internet Explorer 6.0 SP1 is required for SQL Server Management Studio installations.

⁶Internet Explorer 6.0 SP1 is required for Business Intelligence Development Studio installations.

▣ **Installing a Failover Cluster**

When you select SQL Server or Analysis Services, if Setup detects that you are installing to a virtual server, the **Create a SQL Server failover cluster** or **Create an Analysis Server failover cluster** check boxes are enabled. You must select this option to install a failover cluster.

To upgrade a SQL Server instance to a SQL Server 2005 failover cluster, the instance being upgraded must be a failover cluster. To upgrade a stand-alone instance of SQL Server to a SQL Server 2005 failover cluster, install a new SQL Server 2005 failover cluster and then migrate user databases from the stand-alone instance using the Copy Database Wizard. For more information on upgrade, see [How to: Upgrade to a SQL Server 2005 Failover Cluster \(Setup\)](#). For more information on database migration, see **Using the Copy Database Wizard**.

▣ **Installing AdventureWorks Sample Databases**

AdventureWorks, AdventureWorksDW, AdventureWorksAS, and related samples are not installed by default. To install one or more of these features, select **Workstation Components, Books Online, and Development Tools** from the **Components to Install** page during SQL Server 2005 Setup, and then click **Advanced**. Expand **Documentation and Samples**, and then select the databases and samples to be installed.

▣ **See Also**

Reference

[Editions and Components of SQL Server 2005
Feature Selection](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Feature Selection



[See Also](#)



[Setup User Interface Reference](#) >

Updated: **14 April 2006**

SQL Server Express users: click [Feature Selection \(SQL Server Express\)](#) for information that is specific to SQL Server Express Setup.

Use the **Feature Selection** page of the Microsoft SQL Server Installation Wizard to select individual features for your installation, and to specify an installation directory. For an existing installation, use this page to add or remove features.

▣ Options

The **Feature Selection** page consists of two panes. The left pane shows a tree view of the features that you can install and you can expand a feature to view its subfeatures. The right pane displays feature descriptions. When you click a feature or subfeature, the **Feature description** pane displays a description of the selection and its disk space requirements.

To add or remove a feature, click the arrow next to the feature name, and then choose one of the following options from the drop-down list:

Will be installed on local hard drive

Install the selected feature in the location shown under **Installation Path**.

Entire feature will be installed on local hard drive

Install the selected feature and all subfeatures. Expand the parent feature to view its subfeatures.

Will be installed to run from network

Run the following from a network location:

- Replication Monitor

Entire feature will be unavailable

For a new installation of SQL Server, this option passes over the installation of the selected feature. For an existing installation, this option removes the feature from the installation. After you select this option, a red **X** appears on the feature tree next to the feature name, and the icon on any parent features in the tree view appears dimmed.

Note:

You cannot clear a feature if one of its subfeatures is selected. In order to deselect a feature, make sure that all of its subfeatures are also deselected.

The **Feature Selection** page also offers the following options:

Installation path

During Setup, you can change the installation path for the following features:

The installation path is displayed only for features with a user-configurable destination folder:

Component	Default path ^{1, 2}	Configurable or fixed path
Database Engine server components	\Program Files\Microsoft SQL Server\MSSQL.n\MSSQL\Binn\	Configurable
Database Engine data files	\Program Files\Microsoft SQL Server\MSSQL.n\MSSQL\Data\	Configurable
Analysis Services server	\Program Files\Microsoft SQL Server\MSSQL.n\OLAP\Bin\	Configurable
Analysis Services data files	\Program Files\Microsoft SQL Server\MSSQL.n\OLAP\Data\	Configurable
Reporting Services report server	\Program Files\Microsoft SQL Server\MSSQL.n\Reporting Services\ReportServer\Bin\	Configurable
Reporting Services report manager	\Program Files\Microsoft SQL Server\MSSQL.n\Reporting Services\ReportManager\Bin\	Fixed path
Integration Services	<Install Directory>\90\DTS\	Configurable ⁴
Notification Services	<Install Directory>\90\Notification Services\	Configurable ⁴
Client Components	<Install Directory>\90\Tools\	Configurable ⁴
Replication and server-side COM objects	Program Files\Microsoft SQL Server\90\COM ⁵	Fixed path
Integration Services component DLLs for the Data Transformation Runtime engine, the Data	Program Files\Microsoft SQL Server\90\DTS\Binn	Fixed path

Transformation Pipeline engine, and the dtexec command prompt utility		
DLLs that provide managed connection support for Integration Services	Program Files\Microsoft SQL Server\90\DTS\Connections	Fixed path
DLLs for each type of enumerator that Integration Services supports	Program Files\Microsoft SQL Server\90\DTS\ForEachEnumerators	Fixed path
SQL Server Browser Service, WMI providers	\Program Files\Microsoft SQL Server\90\Shared\	Fixed path
Components that are shared between all instances of SQL Server 2005	\Program Files\Microsoft SQL Server\90\Shared\	Fixed path

¹Make sure that the \Program Files\Microsoft SQL Server\ folder is protected with limited permissions.

² The default drive for these locations is *systemdrive*, normally drive C.

³Installation paths for child features are determined by the installation paths of their parent features.

⁴A single installation path is shared between Integration Services, Notification Services, and Client Components. Changing the installation path for one component also changes it for other components. Subsequent installations will install components to the same location as the original installation.

⁵This directory is used by all instances of SQL Server on a computer. If you apply an update, like a service pack, to any of the instances on the computer, any changes to files in this folder will affect all instances on the computer.

 **Note:**

For clustered configurations, you must select a local drive that is available on every node of the cluster.

 **Note:**

When adding features to an existing installation, you cannot change the location of a previously installed feature, nor can you specify the location for a new feature.

When you specify an installation path during Setup for the server components or data files, the Setup program uses the instance ID in addition to the specified location for program and data files. Setup does not use the instance name for tools and other shared files. Setup also does not use any instance name for the Analysis Services program and data files, although it does use the instance name for the Analysis Services repository.

For more information about installation directories, see [File Locations for Default and Named Instances of SQL Server 2005](#).

For more information about hard disk space requirements, see [Hardware and Software Requirements for Installing SQL Server 2005](#).

If you set an installation path for the "Database Engine" feature, SQL Server Setup will use that path as the root directory for all instance-specific folders for that installation, including "SQL Data Files." In this case, if you set the root to "C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\", instance-specific directories will be tacked onto that path.

Customers choosing to use the USESYSDB upgrade functionality in the SQL Server Installation Wizard (Setup UI mode) can easily lead themselves into a situation where the product gets installed into a recursive folder structure - for example, <SQLProgramFiles>\MSSQL.1\MSSQL\MSSQL.1\MSSQL\Data\. Instead, to use the USESYSDB feature, set an installation path for the "SQL Data Files" feature instead of the "Database Engine" feature.

Note: Data files are always expected to be found in a child directory named

"Data." For example, specify "C:\Program Files\Microsoft SQL Server\MSSQL.1\" to specify the root path to the data directory of the system databases during upgrade when data files are found under "C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Data."

Note: You cannot install SQL Server Windows on Windows (WOW64) to the root directory of a physical disk. Configurable paths must be specified to a node below the root directory. For example, you cannot install Database Engine server components to c:\. You can install the Database Engine server components to c:\SQL Server\. For more information about WOW64 see Extended System Support in [Hardware and Software Requirements for Installing SQL Server 2005](#).

Installing AdventureWorks Sample Databases

AdventureWorks Sample OLTP, AdventureWorksDW Sample Data Warehouse, and AdventureWorks Sample OLAP, and related samples are not installed by default. To install one or more of these features, select **Workstation Components, Books Online, and Development Tools** from the **Components to Install** page during SQL Server 2005 Setup and then click **Advanced**. Expand **Documentation and Samples** and then select the databases and samples to be installed.

Browse...

Launches the **Change Folders** page, which allows you to change the folder where the selected feature will be installed. This button is available only for features with a user-configurable installation path. Installation paths for a child feature is determined by the installation path of its parent feature. For more information on installation paths, see [File Locations for Default and Named Instances of SQL Server 2005](#).

Disk Cost...

Displays the available space for installing to different system drives on your computer. After examining the disk space on your computer in the **Disk Cost** dialog box, click **Close** to continue installing SQL Server.

Note:

Browse... and **Disk Cost...** buttons on this page are disabled for failover

cluster installations, as failover cluster installations have a separate **Installation Folders** page.

▣ **See Also**

Reference

[Editions and Components of SQL Server 2005](#)

[Hardware and Software Requirements for Installing SQL Server 2005](#)

[Components to Install](#)

Concepts

[Failover Clustering](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Instance Name



[See Also](#)



[Setup User Interface Reference](#) >

SQL Server Express users: click [Instance Name \(SQL Server Express\)](#) for information that is specific to SQL Server Express Setup.

Use the **Instance Name** page of the SQL Server Installation Wizard to specify whether to create a default instance or a named instance of SQL Server. A default instance name is used unless you choose a named instance.

Each SQL Server instance is made up of a distinct set of services with specific settings for collations and other options. The directory structure, registry structure, and service names all reflect the specific instance ID of the SQL Server instance created during SQL Server Setup.

An instance is either the default instance or a named instance. The default instance name is MSSQLSERVER; it does not require a client to specify the name of the instance to make a connection. A named instance is determined by the user during Setup.

A named instance can be installed at any time. By default, the first time you install SQL Server 2005 on a computer, Setup specifies the installation as the default instance; however, you can choose to install SQL Server as a named instance without installing the default instance first. The default instance could be an installation of SQL Server version 6.5, 7.0, 2000, or 2005. Only one installation of SQL Server, regardless of version, can be the default instance at any one time.

▣ **Multiple Instances**

SQL Server 2005 supports multiple SQL Server instances on a single server or processor, but only one instance can be the default instance; all others must be named instances. A computer can run multiple instances of SQL Server concurrently, and each instance runs independently of other instances. All instances on a single server or processor must be the same localized version of SQL Server 2005.

The following table shows the number of instances supported for each instance-aware component in the different editions of SQL Server 2005.

SQL Server 2005 edition	Database Engine instances	Analysis Services instances	Reporting Services instances
Enterprise Edition	50	50	50
Developer Edition			
Standard Edition	16	16	16
Workgroup Edition			
Express Edition			

▣ Options

If you select **Default instance** and a default instance already exists on the computer, SQL Server Setup will upgrade the default instance. If you enter the name of a named instance that already exists on the computer, SQL Server Setup will upgrade the named instance.

All components of a given SQL Server 2005 instance are managed as a unit; all SQL Server service packs and upgrades will apply to every component of a SQL Server instance.

Consider the following information when deciding whether to install a default or named instance of SQL Server 2005:

- If you are upgrading from SQL Server version 7.0, the upgraded instance is a default instance.
- If you plan to install a single instance of SQL Server on a database server, it should be a default instance.
- If you must support client connections from SQL Server version 7.0 or earlier, it is easier to use a default instance. Although older clients can be configured to connect to a named instance, each client requires a "server name" definition and a hard-coded network address for the named instance.
- Consider a named instance for situations where you plan to have multiple instances on the same computer, as a server can host only one default instance.
- Any application that installs SQL Server Express Edition should install it as a named instance. Doing so minimizes conflict in situations where multiple applications are installed on the same computer.

Default instance

Choose this option to install a default instance of SQL Server 2005. A computer can host only one default instance; all other instances must be named. However, if you have a default instance of SQL Server installed, you can add a default instance of Analysis Services to the same computer.

To upgrade the default instance of a previous SQL Server installation, click the **Default Instance** radio button and then click **Next**.

Named instance

Choose this option to create a new named instance. Bear in mind the following when naming an instance of SQL Server:

- Instance names are not case sensitive.
- Instance names cannot contain the terms "Default," "MSSQLServer," or other reserved keywords. If a reserved keyword is used in an instance name, a Setup error will occur.
- Instance names are limited to 16 characters.
- The first character in the instance name must be a letter or an underscore (_). Acceptable letters are those defined by the Unicode Standard 2.0, including Latin characters a-z, A-Z, and letter characters from other languages.
- Subsequent characters can be letters defined by the Unicode Standard 2.0, decimal numbers from Basic Latin or other national scripts, the dollar sign (\$), or an underscore (_).
- Embedded spaces or other special characters are not allowed in instance names; nor are the backslash (\), comma (,), colon (:), semi-colon (;), single quote ('), ampersand (&), or at sign (@).

 **Note:**

Only characters that are valid in the current Microsoft Windows code page can be used in SQL Server instance names. If an unsupported Unicode character is used, a Setup error will occur.

Installed instances

View a list of installed SQL Server instances and components on the computer where SQL Server Setup is running. The installed instance details pane will display upgrade options for installed instances.

Note that all components of SQL Server that share the same instance name must meet the following criteria:

- **Same version.** For example, you cannot install an instance of SQL Server 2005 Analysis Services (SSAS) with the same instance name as SQL Server 2000 Database Engine on a single computer. In addition, when you upgrade one component with a given instance name to a new version or service

pack, you must upgrade all components of that instance.

- **Same edition.** For example, you cannot install an instance of SQL Server 2005 Integration Services (SSIS) Evaluation Edition with the same instance name as SQL Server 2005 Database Engine Enterprise Edition on a single computer.
- **Same language.** For example, you cannot install an instance of German SQL Server 2005 Database Engine with the same instance name as Portuguese (Brazil) SQL Server Management Studio on the same computer.
- **Same clustered state.** For example, you cannot install a clustered instance of SQL Server 2005 Analysis Services (SSAS) with the same instance name as a nonclustered instance of SQL Server 2005 Database Engine on the same computer.

 **Note:**

Reporting Services is not cluster-aware.

- **Same platform.** For example, you can not install an instance of Business Intelligence Development Studio (32-bit) with the same instance name as an instance of SQL Server 2005 Database Engine (64-bit).

▣ **See Also**

Other Resources

[Database Engine Installation How-to Topics](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Installed Instances



[See Also](#)



[Setup User Interface Reference](#) >

Use the **Installed Instances** page of the Microsoft SQL Server Installation Wizard to view a list of installed SQL Server instances and components on the computer where SQL Server Setup is running. The **Installed Instance Details** pane displays upgrade options for installed instances.

▣ **Options**

To select a component or instance to upgrade, click the instance name and then click **OK**.

▣ **See Also**

Other Resources

[Upgrading to SQL Server 2005](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Existing Components



[See Also](#)



[Setup User Interface Reference](#) >

SQL Server Express users: click [Existing Components \(SQL Server Express\)](#) for information that is specific to SQL Server Express Setup.

Use the **Existing Components** page of the Microsoft SQL Server Installation Wizard to view a list of SQL Server components installed on the computer where SQL Server Setup is running.

▣ **Options**

Click the **Details** button to see a report that lists the maintenance and upgrade options available for each currently installed SQL Server component.

Allowed Actions

Maintenance: Component maintenance is provided through Add or Remove Programs in Control Panel.

Upgrade: Existing components can be upgraded to SQL Server 2005.

▣ **See Also**

Other Resources

[Upgrading to SQL Server 2005](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Installation Options



[See Also](#)



[Setup User Interface Reference](#) >

Updated: **14 April 2006**

SQL Server Express users: click [Installation Options \(SQL Server Express\)](#) for information that is specific to SQL Server Express Setup.

Use the **Installation Options** page of the Microsoft SQL Server Installation Wizard to view maintenance and upgrade options for existing SQL Server instances and components. This page appears when components selected for the current installation are already part of an existing installation with the same instance name.

Options

Maintenance and upgrade options are described in the following sections.

Maintenance

SQL Server 2005 instance maintenance and feature maintenance, including failover clusters, is provided through Add or Remove Programs in Control Panel.

Upgrade

The following upgrade scenarios are supported in this release of SQL Server 2005:

Upgrade from	Supported Upgrade Paths
SQL Server 7.0 Enterprise Edition SP4	SQL Server 2005 Enterprise Edition
SQL Server 7.0 Developer Edition SP4	SQL Server 2005 Enterprise Edition SQL Server 2005 Developer Edition
SQL Server 7.0 Standard Edition SP4	SQL Server 2005 Standard Edition SQL Server 2005 Enterprise Edition
SQL Server 7.0 Desktop Edition SP4	SQL Server 2005 Standard Edition SQL Server 2005 Workgroup Edition
SQL Server 7.0 Evaluation Edition SP4	Upgrades not supported
SQL Server Desktop Engine (MSDE) 7.0 SP4	SQL Server 2005 Express Edition
SQL Server 2000 Enterprise Edition SP3 or later ¹	SQL Server 2005 Enterprise Edition
SQL Server 2000 Developer Edition SP3 or later ¹	SQL Server 2005 Developer Edition
SQL Server 2000 Standard Edition	SQL Server 2005 Enterprise Edition SQL Server 2005 Developer Edition

SP3 or later ¹	SQL Server 2005 Standard Edition
SQL Server 2000 Workgroup Edition SP3 or later ¹	SQL Server 2005 Enterprise Edition SQL Server 2005 Developer Edition SQL Server 2005 Standard Edition SQL Server 2005 Workgroup Edition
SQL Server 2000 Personal Edition SP3 or later ¹	SQL Server 2005 Standard Edition SQL Server 2005 Workgroup Edition SQL Server 2005 Express Edition SQL Server 2005 Express Edition with Advanced Services
SQL Server 2000 Evaluation Edition SP3 or later ¹	SQL Server 2005 Evaluation Edition
SQL Server Desktop Engine (MSDE) 2000 SP3 or later ¹	SQL Server 2005 Workgroup Edition SQL Server 2005 Express Edition SQL Server 2005 Express Edition with Advanced Services
SQL Server 2000 IA64 (64-bit) Enterprise Edition	SQL Server 2005 IA64 (64-bit) Enterprise Edition
SQL Server 2000 IA64 (64-bit) Developer Edition	SQL Server 2005 IA64 (64-bit) Enterprise Edition SQL Server 2005 IA64 (64-bit) Developer Edition
SQL Server 2005 Developer Edition ¹	SQL Server 2005 Enterprise Edition SQL Server 2005 Standard Edition SQL Server 2005 Workgroup Edition
SQL Server 2005 Standard Edition ¹	SQL Server 2005 Enterprise Edition

	SQL Server 2005 Developer Edition
	SQL Server 2005 Enterprise Edition
SQL Server 2005 Workgroup Edition ¹	SQL Server 2005 Developer Edition
	SQL Server 2005 Standard Edition
	SQL Server 2005 Enterprise Edition
	SQL Server 2005 Developer Edition
SQL Server 2005 Evaluation Edition ¹	SQL Server 2005 Standard Edition
	SQL Server 2005 Workgroup Edition
	SQL Server 2005 Express Edition
	SQL Server 2005 Enterprise Edition
	SQL Server 2005 Developer Edition
SQL Server 2005 Express Edition ¹	SQL Server 2005 Standard Edition
	SQL Server 2005 Workgroup Edition
	SQL Server 2005 Express Edition with Advanced Services
SQL Server 2005 IA64 (64-bit) Developer Edition	SQL Server 2005 IA64 (64-bit) Enterprise Edition
	SQL Server 2005 IA64 (64-bit) Standard Edition
SQL Server 2005 X64 (64-bit) Developer Edition	SQL Server 2005 X64 (64-bit) Enterprise Edition
	SQL Server 2005 X64 (64-bit) Standard Edition
SQL Server 2005 IA64 (64-bit) Standard Edition	SQL Server 2005 IA64 (64-bit) Enterprise Edition
	SQL Server 2005 IA64 (64-bit) Developer Edition

SQL Server 2005 X64 (64-bit) Standard Edition	SQL Server 2005 X64 (64-bit) Enterprise Edition SQL Server 2005 X64 (64-bit) Developer Edition
SQL Server 2005 IA64 (64-bit) Evaluation Edition	SQL Server 2005 IA64 (64-bit) Enterprise Edition SQL Server 2005 IA64 (64-bit) Developer Edition SQL Server 2005 IA64 (64-bit) Standard Edition
SQL Server 2005 X64 (64-bit) Evaluation Edition	SQL Server 2005 X64 (64-bit) Enterprise Edition SQL Server 2005 X64 (64-bit) Developer Edition SQL Server 2005 X64 (64-bit) Standard Edition

¹This SQL Server edition can be upgraded to SQL Server 2005 on the 32-bit subsystem (WOW64) of a 64-bit server.

Upgrade Notes

- After upgrading to SQL Server 2005 from another edition of SQL Server 2005, you must re-apply any hotfix or service pack updates to the upgraded SQL Server instance.
- Cross-version instances on SQL Server 2005 are not supported. Version numbers of the Database Engine, Analysis Services, and Reporting Services components must be the same within an instance of SQL Server 2005.
- Before upgrading to SQL Server 2005, enable Windows Authentication for SQL Server Agent and verify that the SQL Server Agent service account is a member of the SQL Server **sysadmin** group.
- Before upgrading from one edition of SQL Server 2005 to another, verify that the functionality you are currently using is supported in the edition to which you are upgrading. For more information, see the section for your

components in [Features Supported by the Editions of SQL Server 2005](#).

- To upgrade a SQL Server instance to a SQL Server 2005 failover cluster, the instance being upgraded must be a failover cluster. To upgrade a stand-alone instance of SQL Server to a SQL Server 2005 failover cluster, install a new SQL Server 2005 failover cluster and then migrate user databases from the stand-alone instance using the Copy Database Wizard. For more information on upgrade, see [How to: Upgrade to a SQL Server 2005 Failover Cluster \(Setup\)](#). For more information on database migration, see **Using the Copy Database Wizard**.

▣ **Adding Components to an Instance of SQL Server 2005**

You can add components to an existing installation of SQL Server 2005 from **Add or Remove Programs** in Control Panel and then select the instance of SQL Server to which you want to add or remove features. For more information, see [How to: Change Components in an Existing Instance of SQL Server 2005 \(Setup\)](#).

▣ **Cross-Language Support**

For upgrades:

- English-language versions of SQL Server can be upgraded to any localized version of SQL Server 2005.
- Localized versions of SQL Server can be upgraded to localized versions of SQL Server 2005 of the same language.
- Localized version of SQL Server cannot be upgraded to the English-language version of SQL Server 2005.
- Localized versions of SQL Server cannot be upgraded to localized SQL Server 2005 versions of a different localized language.

Additional cross-language support:

- The English-language version of SQL Server 2005 is supported on all localized versions of supported operating systems.
- Localized versions of SQL Server 2005 are supported on localized operating systems that are the same language as the localized SQL Server version.
- Localized versions of SQL Server 2005 are also supported on English-language versions of supported operating systems through the use of Windows Multilingual User Interface Pack (MUI) settings. However, you must verify certain operating system settings before installing a localized version of SQL Server 2005 on a server that is running an English-language operating system with a non-English MUI setting. You must verify that the following operating system settings match the language of the localized SQL Server to be installed:
 - The operating system user interface setting
 - The operating system user locale setting
 - The system locale setting

If these operating system settings do not match the language of the localized SQL Server, then you must correctly set these operating system settings as described in [How to: Change Operating System Settings to Support Localized Versions](#).

▣ **See Also**

Other Resources

[Upgrading to SQL Server 2005](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Service Account



[See Also](#)



[Setup User Interface Reference](#) >

Updated: 5 December 2005

SQL Server Express users: click [Service Account \(SQL Server Express\)](#) for information that is specific to SQL Server Express Setup.

Use the **Service Account** page of the Microsoft SQL Server Installation Wizard to assign a login account to the SQL Server services. The actual services configured on this page depend on the features you have selected to install.

Options

You can assign the same login account to all SQL Server services, or you can configure each service account individually. You can also specify whether services start automatically.

Security Note:

Setting strong passwords is essential to the security of your system. Always use strong passwords.

Customize the logon for each service account

Select the **Customize for each service account** check box to customize settings for individual services.

This option assigns specific logon accounts to individual services. Click this check box to implement the principle of least privileges, where SQL Server services are granted the minimum permissions they need to complete their tasks. For more information, see [Setting Up Windows Service Accounts](#).

If this check box is not selected, the same account and settings are used for all SQL Server services.

Select any of the following services to customize its settings.

Select this service	To configure authentication settings for
SQL Server	The SQL Server Database Engine
SQL Server Agent	The service that executes jobs, monitors, SQL Server, and allows automation of administrative tasks.
Analysis Services	Analysis Services

Reporting Services	Reporting Services. Service accounts are used to configure a report server database connection. Choose a domain user account if you want to connect to a report server database on a remote SQL Server instance. If you are using a local report server database, you can use a domain user account or Local System to run the service.
SQL Server Browser	SQL Server Browser is the name resolution service that provides SQL Server connection information to client computers. This service is shared across multiple SQL Server and Integration Services instances.

Use the built-in System account

You can assign Local System, Network Service, or Local Service to the logon for the configurable SQL Server services.

The Local System option specifies a local system account that does not require a password to connect to SQL Server on the same computer. However, the local system account may restrict the SQL Server installation from interacting with other servers, depending on the privileges granted to the account.

 **Important:**

Local System is a powerful account; it may not be appropriate for all Service settings. For more information, see [Security Considerations for a SQL Server Installation](#).

The Network Service account is a special, built-in account that is similar to an authenticated user account. The Network Service account has the same level of access to resources and objects as members of the Users group. Services that run as the Network Service account access network resources using the credentials of the computer account.

 **Important:**

We recommend that you do not use the Network Service account for the SQL Server or the SQL Server Agent services. Local User or Domain User accounts are more appropriate for these SQL Server services.

The Local Service account is a special, built-in account that is similar to an authenticated user account. The Local Service account has the same level of access to resources and objects as members of the Users group. This limited access helps safeguard your system if individual services or processes are compromised. Services that run as the Local Service account access network resources as a null session with no credentials.

For more information on service accounts, see [Setting Up Windows Service Accounts](#).

Use a domain user account

Specifies a domain user account that uses Windows Authentication to set up and connect to SQL Server. Microsoft recommends using a domain user account with minimal rights for the SQL Server service, as the SQL Server service does not require administrator account privileges.

The SQL Server Agent account must have administrator privileges if you create CmdExec and ActiveScript jobs that belong to someone other than a SQL Server administrator, or if you use the AutoRestart feature. If the above features are requirements in your environment, consider using separate service accounts for the SQL Server and SQL Server Agent services.

For strong password guidelines, see [Authentication Mode](#).

 **Note:**

The domain name cannot be a full DNS name. For example, if your DNS name is *my-domain-name.com*, use *my-domain-name* in the domain field. SQL Server Setup will not accept *my-domain-name.com* in the domain field.

Start services at the end of SQL Server Setup

Automatically starts the services when your operating system is started. For SQL Server 2005, the services **Auto-start** option will be selected by default for SQL Server, Analysis Services, and Reporting Services.

 **Note:**

The SQL Server Agent service depends on the SQL Server service. If you select the **Auto-start** check box for SQL Server Agent, **Auto-start** is automatically selected for the SQL Server service and it cannot be unchecked unless you uncheck the option for SQL Server Agent.

 **Important:**

Off by Default - To enhance security in SQL Server 2005, some services and features are not activated by default. They have to be configured and enabled after Setup is complete. For more information, see [SQL Server Surface Area Configuration](#) and [Security Considerations for a SQL Server Installation](#).

▣ **See Also**

Reference

[Setting Up Windows Service Accounts](#)

Concepts

[Security Considerations for a SQL Server Installation](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Authentication Mode



[See Also](#)



[Setup User Interface Reference](#) >

SQL Server Express users: click [Authentication Mode \(SQL Server Express\)](#) for information that is specific to SQL Server Express Setup.

Use the **Authentication Mode** page of the Microsoft SQL Server Installation Wizard to choose the security mode used to authenticate client and server connections to this installation. If you select **Mixed Mode**, you must enter and confirm the SQL Server system administrator (**sa**) password. After a device establishes a successful connection to SQL Server, the security mechanism is the same for both Windows Authentication and Mixed Mode.

▣ Options

Windows Authentication Mode

When a user connects through a Microsoft Windows user account, SQL Server validates the account name and password using information in the Windows operating system. This is the default authentication mode, and is much more secure than Mixed Mode. Windows Authentication utilizes Kerberos security protocol, provides password policy enforcement in terms of complexity validation for strong passwords, provides support for account lockout, and supports password expiration.

Security Note:

When possible, use Windows Authentication.

If you select Windows Authentication, Setup creates an **sa** account that is disabled by default. To utilize Mixed Mode Authentication and activate the **sa** account after Setup completes, see the topic for "How to: Change Server Authentication Mode" in SQL Server 2005 Books Online.

Important:

Setting strong passwords is essential to the security of your system. Never set a blank or weak **sa** password.

Mixed Mode (Windows Authentication or SQL Server Authentication)

Allows users to connect using Windows Authentication or SQL Server Authentication. Users who connect through a Windows user account can make use of trusted connections that are validated by Windows.

If you must choose Mixed Mode Authentication and you have a requirement for using SQL logins to accommodate legacy applications, then you must set strong passwords for all SQL accounts. This is of particular importance for accounts that are members of the sysadmin role, especially the **sa** account.

 **Note:**

SQL Server Authentication is provided for backward compatibility only. When possible, use Windows Authentication.

Enter Password

Enter and confirm the system administrator (**sa**) login when you choose Mixed Mode Authentication. Passwords are the first line of defense against intruders, so setting strong passwords is essential to the security of your system. Never set a blank or weak sa password.

 **Note:**

SQL Server passwords can contain from 1 to 128 characters, including any combination of letters, symbols, and numbers. If you choose Mixed Mode authentication, you must enter a strong **sa** password before you can continue to the next page of the Installation Wizard.

Strong Password Guidelines

Strong passwords are not readily guessed by a person, and are not easily hacked using a computer program. Strong passwords cannot use prohibited conditions or terms, including:

- A blank or NULL condition
- "Password"
- "Admin"
- "Administrator"
- "sa"
- "sysadmin"

A strong password cannot be the following terms associated with the installation computer:

- The name of the user currently logged onto the machine.
- The machine name.

A strong password must be more than 8 characters in length and satisfy at least three of the following four criteria:

- It must contain uppercase letters.
- It must contain lowercase letters.
- It must contain numbers.
- It must contain non-alphanumeric characters; for example, #, %, or ^.

Passwords entered on the **Authentication Mode** page must meet strong password policy requirements. If you have any automation that uses SQL Authentication, make sure that the password meets strong password policy requirements.

 **Security Note:**

For this release, Setup may not enforce the strong password requirement on some default configurations of Windows Server 2003 where the machine is not a member of a domain. Setting strong passwords is essential to the security of your system. Never set a blank or weak **sa** password.

▣ Change History

Release	History
5 December 2005	Changed content: <ul style="list-style-type: none">• Updated strong password guidelines: Strong passwords require more than 8 characters.

▣ **See Also**

Concepts

[Security Considerations for a SQL Server Installation](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Collation Settings



[See Also](#)



[Setup User Interface Reference](#) >

Updated: 5 December 2005

SQL Server Express users: click [Collation Settings \(SQL Server Express\)](#) for information that is specific to SQL Server Express Setup.

Use the **Collation Settings** page of the Microsoft SQL Server Installation Wizard to modify default collation settings used by the Database Engine and Analysis Services for language and sorting purposes. Choose the **Collation Designator** option to match collation settings of different installations of SQL Server or of another computer. Use the **SQL Collations** option to match settings that are compatible with the sort orders in earlier versions of SQL Server.

▣ Options

Customize for each service account

You can specify separate collation settings for the Database Engine and Analysis Services, or you can specify a single collation for all of the services. If you select both the Database Engine and Analysis Services for your installation, you can select the check box next to **Customize for each service account** to activate the list of services in the drop-down list. Select a service from the drop-down list and then make your collation and sort order selections for that service. After you have made your selections, click **Next** to continue.

Collation designator and sort order

Designate the collation to be used by this instance of SQL Server 2005. A SQL collation is selected by default for English-language system locales. The default collation for non-English locales is the Microsoft Windows system locale setting for your computer - the **Language for non-Unicode programs** setting, or the closest equivalent from the Regional and Language Options in Control Panel.

Change the default settings only if the collation setting for this installation of SQL Server must match the collation settings used by another instance of SQL Server, or if it must match the Windows system locale of another computer.

Note SQL collations cannot be used for Analysis Services. If you select SQL collations for use with your Database Engine installation, SQL Server Setup will select a best match Windows collation for Analysis Services, based on the SQL collation you select. If the Database Engine and Analysis Services collations do not match, you may receive inconsistent results. To ensure consistent results between the Database Engine and Analysis Services, use Windows collations.

For more information, see [Collation Settings in Setup](#).

Sort order

Specify the sort order to use with the **Collation designator** you have selected. Binary is the fastest sorting order and is case-sensitive. If **Binary** is selected, the **Case-sensitive**, **Accent-sensitive**, **Kana-sensitive**, and **Width-sensitive** options are not available. For more information, see [Windows Collation Sorting Styles](#).

SQL collations

Match settings compatible with SQL Server versions 8.0, 7.0, or earlier. The **SQL collations** option is used for compatibility with earlier versions of SQL Server. For more information, see [Using SQL Collations](#).

 **Note:**

SQL collations cannot be used for Analysis Services. If you select SQL collations for use with your Database Engine installation, SQL Server Setup will select a best match Windows collation for Analysis Services, based on the SQL collation you select. If the Database Engine and Analysis Services collations do not match, you may receive inconsistent results. To ensure consistent results between the Database Engine and Analysis Services, use Windows collations.

▣ Typical User Scenarios

For a table of Windows System Locales and the corresponding default collations used by SQL Server Setup, see [Collation Settings in Setup](#).

Best practices include use of a standardized collation for your organization, if possible. Using a standard collation setting across all systems in your organization will help to eliminate the need to explicitly specify the collation in every character or Unicode expression. If you must work with objects that have different collation and code page settings, you must code your queries to consider the rules of collation precedence.

When selecting a collation for your SQL Server installation, consider the following recommendations, presented in priority order:

- Select a binary collation if binary ordering is acceptable.
- If your applications use both **nchar/nvarchar** and **char/varchar** data types and may mix them in comparison, select a Windows collation for consistent comparison across data types.
- If your applications use only **nchar/nvarchar** data types, there is no difference in behavior between Windows collations and SQL collations.
- If your applications use only **char/varchar** data types, and you utilize Analysis Services, select a Windows collation.
- If your applications use only **char/varchar** data types, and you are concerned about string comparisons/LIKE performance, and you are not concerned about the better linguistic comparison semantics offered by Windows collations, and you want the dictionary sort order offered by SQL collations, select a SQL collation.

▣ **See Also**

Reference

[Collation Options and International Support](#)
[Collation Settings in Setup](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Report Server Installation Options



[See Also](#)



[Setup User Interface Reference](#) >

Use the **Report Server Installation Options** page of the Microsoft SQL Server Installation Wizard to specify options that determine how a report server instance is installed. The availability of an installation option depends on whether you are also installing a local instance of SQL Server Database Engine at the same time you are installing the report server, and whether other default values are available to use.

If Setup cannot perform a default report server configuration because one or more requirements are not met, the Installation Wizard allows only the minimal installation option; copying the files you need, but requiring you to use the Reporting Services Configuration tool to configure the report server after setup is finished.

If a Secure Sockets Layer (SSL) certificate is installed on the computer and associated with the default Web site, Setup detects the certificate and configures the report server to use SSL when connecting to other systems. If you do not have SSL installed or if Setup cannot use it, you can configure a report server manually to use SSL through the Reporting Services Configuration tool or by setting **SecureConnectionLevel** in the RSReportServer configuration file. For more information on how to change the **SecureConnectionLevel** in the configuration file, see **Using Secure Web Service Methods** in SQL Server Books Online.

▣ Options

Install the default configuration

Installs a report server instance using the default values for the report server databases, service accounts, and virtual directories. When you choose this option, the report server instance is ready to use when Setup is finished. Setup creates the report server database using a local Database Engine instance, and configures a report server to use default values.

This option is available only if the default values used in a report server installation are valid for your system. This option is recommended for developers who want to install all components locally, and for users who are evaluating the software.

To view information about the default Settings that Setup uses, or to find out why the default configuration cannot be installed, click **Details**. To learn more about the default configuration, see [Default Configuration for a Report Server Installation](#).

Install but do not configure the server

Installs the report server program files, creates the report server Windows service account, and registers the report server Windows Management Instrumentation (WMI) provider. This installation option is referred to as a "Files-only" installation. You can choose this option if you do not want to use the default configuration. If the default configuration cannot be installed, this is the only option available.

After Setup completes, you must create the report server database and configure the report server before it can be used. To configure a report server and create the database, use the Reporting Services Configuration tool. For more information, see [Creating a Report Server Database](#) and [Configuring a Report Server Database Connection](#).

▣ **See Also**

Tasks

[How to: Start Reporting Services Configuration](#)

[How to: Create a Report Server Database \(Reporting Services Configuration\)](#)

Concepts

[Installing Reporting Services Using Setup](#)

[Default Configuration for a Report Server Installation](#)

[Selecting Reporting Services Components to Install](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Report Server Installation Details



[See Also](#)



[Setup User Interface Reference](#) >

Use the **Report Server Installation Details** page of the Microsoft SQL Server Installation Wizard to view the configuration details of a report server installation before it is created. If Setup cannot install a report server in the default configuration, this page explains which default values are missing. Otherwise, this page describes the settings, names, and paths that will be configured during the setup process. Depending on your computer configuration and installation options, you can expect to see the following information in this page:

- Report server information displays the user account used to run the report server Windows service and Web service. The Windows service account is specified in the Installation Wizard in the **Service Accounts** page. The Web service account is the default ASP.NET account for this computer. These accounts will be used by the report server to access the report server database.
- Report server database information identifies the local SQL Server instance that hosts the report server databases. Setup determines which instance of the Database Engine is used. It is always a local instance, and it is always the instance that is installed concurrently with the report server. For more information about the Database Engine instance used, see [Instance Name](#).
- Virtual directory information identifies the names of the virtual directories used to access report server and Report Manager. The virtual directories are created under the default Web site. They are named **reportserver** and **reports**, respectively. You cannot specify different names during setup, or choose a different Web site for the virtual directories.
After Setup completes, you can use the URLs noted on this page to access the server components. The URL specifies the computer name and instance name if multiple report server instances are installed locally.
- Secure Sockets Layer (SSL) information tells you whether an SSL

certificate is installed on the computer and whether it will be used for report server connections.

 **Note:**

You can modify the report server installation using the Reporting Services Configuration tool after Setup is complete.

▣ **See Also**

Reference

[Service Account](#)

[Report Server Installation Options](#)

Concepts

[Installing Reporting Services Using Setup](#)

[Default Configuration for a Report Server Installation](#)

[Selecting Reporting Services Components to Install](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Error and Usage Report Settings



[See Also](#)



[Setup User Interface Reference](#) >

Updated: 5 December 2005

SQL Server Express users: click [Error and Usage Report Settings \(SQL Server Express\)](#) for information that is specific to SQL Server Express Setup.

Use the **Error and Usage Report Settings** page of the Microsoft SQL Server Installation Wizard to enable feature error and usage reporting functionality for SQL Server 2005.

▣ Options

The Feature Usage data collection and Error Reporting features are enabled by default for SQL Server 2005 and its components in this release. However, you can disable one or both options by clearing the checkboxes.

Error Reporting,

If you enable the Error Reporting feature, SQL Server will be configured to send a report to Microsoft automatically if a fatal error occurs in any of the following SQL Server components:

- The Database Engine
- SQL Server Agent
- Analysis Services
- Reporting Services
- Notification Services
- Integration Services
- Replication

Microsoft uses error reports to improve SQL Server functionality and treats all information as confidential.

Information about errors is sent over a secure (https) connection to Microsoft, where it is stored with limited access. Alternatively, error reports can be sent to your own Corporate Error Reporting server. See this [Microsoft Web Site](#) for more information about setting up a Corporate Error Reporting server.

Error reports contain the following information:

- The condition of SQL Server when the problem occurred.
- The operating system version and computer hardware information.
- Your Digital Product ID, which is not used to identify your license.
- The network IP address of your computer or proxy server.
- Information from memory or file(s) of the process that caused the error.

Microsoft does not intentionally collect your files, name, address, e-mail address, or any other form of personal information. The error report can, however, contain personal information from the memory or file(s) of the process that caused the error. Although this information can potentially be used to determine your identity, Microsoft does not use this information for that purpose.

For the Microsoft SQL Server privacy and data collection policy, see this [Microsoft Web site](#).

If you enable Error Reporting and a fatal error occurs, you may see a response from Microsoft in the Windows Event log that points to a Microsoft Knowledge Base article on a particular error.

To disable Error or Feature Usage reporting for all instances of SQL Server and its components after Setup completes, go to the **Error and Usage Report Settings** dialog and clear the check boxes for **Feature Usage**. If **Error Reporting** is enabled for multiple components of SQL Server (the SQL Server Database Engine, Analysis Server, Reporting Server, and shared components) you can disable Error Reporting for each instance of an individual component as well as shared components, listed as **Others**.

Feature Usage

If you enable Feature Usage reporting, SQL Server will be configured to periodically send a report to Microsoft. Reports will include information about your hardware configuration and how you use SQL Server 2005 software and services. Microsoft will use feature usage data to improve SQL Server. SQL Server 2005 components monitored by this feature include:

- The Database Engine
- Analysis Services
- Reporting Services
- Notification Services
- Integration Services
- Replication
- Business Intelligence Development Studio

Information about feature use will be sent to Microsoft, where it will be

stored with limited access.

To disable feature usage reporting after Setup completes, use the **SQL Server Error and Usage Reporting** tool on the SQL Server 2005 **Configuration Tools** menu.

Information is collected once per day for all enabled SQL Server instances. The time of collection defaults to midnight to minimize the load on the server. If you want to change the time of collection, you can manually edit the registry key that controls the collection time. Each SQL Server instance has its own registry key; the following example is for the first SQL Server instance installed on the server (MSSQL.1):

```
HKLM\Software\Microsoft\Microsoft SQL  
Server\MSSQL.1\CPE\TimeofReporting
```

The value of this registry key contains the time for the collection as the number of minutes from 00:00 (midnight) to run. For example, a value of 60 would run the collection at 1:00 a.m., a value of 1200 would run the collection at 8:00 p.m., and so on.

▣ **See Also**

Concepts

[Configuring a SQL Server Installation](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Ready to Install



[See Also](#)



[Setup User Interface Reference](#) >

SQL Server Express users: click [Ready to Install \(SQL Server Express\)](#) for information that is specific to SQL Server Express Setup.

Before installing or updating Microsoft SQL Server 2005, the **Ready to Install** page provides a summary of pre-requisites and features to be installed or changed by SQL Server Setup.

▣ Options

You have the following options on this page:

- To continue with Setup, click **Install**.
- To change your installation settings, click **Back**.
- To exit Setup, click **Cancel**.

▣ **See Also**

Tasks

[How to: Verify a Successful Installation of SQL Server 2005 Services](#)

[How to: Read a SQL Server 2005 Setup Log File](#)

Concepts

[How to: View SQL Server 2005 Setup Log Files](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Upgrade Logon Information

 [See Also](#)



[Setup User Interface Reference](#) >

Use the **Upgrade Logon Information** page of the Microsoft SQL Server Installation Wizard to provide logon credentials for the SQL Server instance to be upgraded.

During the upgrade process, SQL Server 2005 Setup must access the instance to be upgraded. Credentials provided on this page allow Setup to login to the previous SQL Server instance.

Windows Authentication

Windows Authentication mode allows a user to connect through a Microsoft Windows user account. Microsoft recommends use of Windows Authentication.

SQL Server Authentication

When a user connects with a specified login name and password from a non-trusted connection, SQL Server performs the authentication itself by checking to see if a SQL Server login account has been set up and if the specified password matches the one previously recorded. If SQL Server does not have a login account set, authentication fails, and the user receives an error message.

Security Note:

When possible, use Windows Authentication.

User name

Enter the user name to connect with. This option is only available if you have selected to connect using Windows Authentication.

Password

Enter the password for the login. This option is only editable if you have

selected to connect using SQL Server Authentication.

▣ **See Also**

Other Resources

[Upgrading to SQL Server 2005](#)

How to: Upgrade to SQL Server 2005 with the Copy Database Wizard

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Upgrade Blocked



[See Also](#)



[Setup User Interface Reference](#) >

The Microsoft SQL Server component you have selected to upgrade is blocked. The **Upgrade Blocked** dialog displays the cause of the block, and the user action necessary to remove the blocking issue.

To continue with the upgrade process, remove blocking issues, and then run Microsoft SQL Server 2005 Setup again.

▣ **See Also**

Concepts

[Using Upgrade Advisor to Prepare for Upgrades](#)

Other Resources

[Upgrading to SQL Server 2005
Backward Compatibility](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Virtual Server Name



[See Also](#)



[Setup User Interface Reference](#) >

Use the **Virtual Server Name** page of the Microsoft SQL Server Installation Wizard to specify the name assigned to your virtual server.

▣ Options

Virtual server name

Enter a unique instance name for your installation. This will be the name used to identify your failover cluster on the network. The virtual SQL Server name must be unique on your network, and must have a name that is different than the host cluster and the cluster nodes.

▣ **See Also**

Reference

[Instance Name](#)

Concepts

[Failover Clustering](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Virtual Server Configuration



[_See Also](#)



[Setup User Interface Reference](#) >

Use the **Virtual Server Configuration** page of the Microsoft SQL Server Installation Wizard to define the IP address configuration for the virtual server.

▣ Options

Virtual Server Name

Displays the name of the virtual server at the defined IP address.

Network to use

Specifies the network for which you want to enter an IP address.

IP Address

Specifies the IP address for each network you want to configure for client access. Enter one IP address for each network on which the virtual server will be available to clients on a public (or mixed) network.

To avoid IP address conflicts, make sure that the IP addresses assigned to the heartbeat and to SQL Server 2005 are not shared by any other application.

Network Address

Displays the network address supplied by the Microsoft Cluster Service.

Network Subnet

Displays the network subnet supplied by the Microsoft Cluster Service.

Add

Adds a network and IP address to the **Selected Networks and IP Addresses** dialog box.

Remove

Removes the selected network and IP address from the **Selected Networks and IP Addresses** dialog box.

Selected Networks and IP Addresses

Displays any selected networks and IP addresses configured for the virtual server.

▣ **See Also**

Concepts

[Failover Clustering](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Cluster Group Selection



[_See Also](#)



[Setup User Interface Reference](#) >

Updated: 5 December 2005

Use the **Cluster Group Selection** page of the Microsoft SQL Server Installation Wizard to select the cluster group. The selected cluster group is where the SQL Server virtual server resources are placed.

▣ Options

Available Cluster Groups

Select the cluster group where the SQL Server virtual server resources are placed. Only one virtual server can be installed per cluster group.

If you select the group containing the cluster quorum resource, a warning appears. It is recommended that you do not install to the cluster quorum resource.

Unavailable Cluster Groups

Lists any unavailable cluster groups, and the reason each group is unavailable. For example, a cluster group will be listed as unavailable if it already contains a virtual server.

Data files

Shows the default location for Database Engine data files and Analysis Services data files.

If you have multiple, shared drives in your cluster group, use the drop-down list to select the drive for shared data files.

▣ **See Also**

Concepts

[Installing a Failover Cluster](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Service Accounts (Clusters)



[_See Also](#)



[Setup User Interface Reference](#) >

Updated: 5 December 2005

Use the **Service Accounts (Clusters)** page of the Microsoft SQL Server Installation Wizard to assign a login account to the SQL Server services. The actual services configured on this page depend on the features that you have selected to install.

Options

You can assign the same login account to all SQL Server services, or you can configure each service account individually. You can also specify whether services start automatically.

Security Note:

Setting strong passwords is essential to the security of your system. Always use strong passwords.

Customize for each service account

Select the **Customize for each service account** check box to customize settings for individual services.

This option assigns specific logon accounts to individual services. Click this check box to implement the principle of least privileges, where SQL Server services are granted the minimum permissions they need to complete their tasks. For more information, see [Setting Up Windows Service Accounts](#).

If this check box is not selected, the same account and settings are used for all SQL Server services.

Select any of the following services to customize its settings.

Select this service	To configure authentication settings for
SQL Server	The SQL Server Database Engine
SQL Server Agent	The service that executes jobs, monitors, SQL Server, and allows automation of administrative tasks.
Analysis Services	Analysis Services

Reporting Services	Reporting Services. Service accounts are used to configure a report server database connection. Choose a domain user account if you want to connect to a report server database on a remote SQL Server instance. If you are using a local report server database, you can use a domain user account or Local System to run the service.
SQL Server Browser	SQL Server Browser is the name resolution service that provides SQL Server connection information to client computers. This service is shared across multiple SQL Server and Integration Services instances.

Use the built-in System account

This option is disabled because you cannot use a built-in System account on a SQL Server Cluster.

Use a domain user account

Specifies a domain user account that uses Windows Authentication to set up and connect to SQL Server. We recommend using a domain user account with minimal rights for the SQL Server service, as the SQL Server service does not require administrator account privileges.

The SQL Server Agent account must have administrator privileges if you create CmdExec and ActiveScript jobs that belong to someone other than a SQL Server administrator, or if you use the AutoRestart feature. If the above features are requirements in your environment, consider using separate service accounts for the SQL Server and SQL Server Agent services.

For strong password guidelines, see [Authentication Mode](#).

 **Note:**

The domain name cannot be a full DNS name. For example, if your DNS name is *my-domain-name.com*, use *my-domain-name* in the domain field. SQL Server Setup will not accept *my-domain-name.com* in the domain field.

Start services at the end of SQL Server Setup

Automatically starts the services when your operating system is started. For SQL Server 2005, the services **Auto-start** option will be selected by default for SQL Server, Analysis Services, and Reporting Services.

 **Note:**

The SQL Server Agent service depends on the SQL Server service. If you select the **Auto-start** check box for SQL Server Agent, **Auto-start** is automatically selected for the SQL Server service and it cannot be unchecked unless you uncheck the option for SQL Server Agent.

 **Important:**

Off by Default - To enhance security in SQL Server 2005, some services and features are not activated by default. They have to be configured and enabled after Setup is complete. For more information, see [SQL Server Surface Area Configuration](#) and [Security Considerations for a SQL Server Installation](#).

▣ **See Also**

Reference

[Setting Up Windows Service Accounts](#)

Concepts

[Security Considerations for a SQL Server Installation](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Domain Groups for Clustered Services



[See Also](#)



[Setup User Interface Reference](#) >

Updated: 12 December 2006

Use the **Domain Groups for Clustered Services** page of the Microsoft SQL Server Installation Wizard to specify global or local domain groups for the clustered services that will be installed as part of your SQL Server 2005 failover cluster.

To ensure a secure operating environment, access to resources on failover cluster nodes must be restricted. On a SQL Server 2005 failover cluster, domain groups that are common to all cluster nodes are used to control access to registry keys, files, SQL Server objects, and other cluster resources. All resource permissions are controlled by domain-level groups that include SQL Server service accounts as members.

Adding domain groups is one of the principal responsibilities of a domain administrator. Ensure that all domain groups for your SQL Server 2005 failover cluster are created or modified with consideration for the security policies of your organization.

▣ Options

SQL Server service, SQL Server Agent service, Analysis Services service, and Full-Text Search service must run as domain accounts. The domain groups for these services must exist at the time SQL Server Setup is run. If necessary, ask your domain administrator for the names of existing global or local domain groups, or to create domain groups for your failover cluster.

For each clustered service in the instance of SQL Server that you are installing, enter the domain and group name in the format `<DomainName>\<GroupName>`, subject to the following guidelines:

- Domain groups can be global domain groups or local domain groups.
- Domain groups must be within the same domain as the machine accounts. For example, if the machine where SQL Server will be installed is in the SQLSVR domain which is a child of MYDOMAIN, you must specify a group in the SQLSVR domain. The SQLSVR domain may contain user accounts from MYDOMAIN.
- The user must be a direct member of the domain group, not a member via sub-groups. Setup will not follow sub-group memberships to validate that the user account is in the domain group.
- The domain and domain group names must already exist at the time Setup is run. If necessary, ask your domain administrator for the names of existing domain groups, or to create domain groups for your failover cluster. If domain groups are created immediately prior to running Setup, allow time for changes to propagate through a corporate network.
- The domain groups should have the appropriate service accounts added to them. If the service accounts are not members of the appropriate domain groups at the time Setup is run, Setup will attempt to add them. In this case, the account under which Setup is running must have adequate privileges to add accounts to the domain groups. If SQL Server Setup is run under an account that does not have permission to add users to the domain groups, the users must already be members of the appropriate group.
- Each service should use a different domain group. However, you can use the same domain group and the same account for all SQL Server services, you can use the same domain group and different accounts for each SQL Server service, or you can use a different domain group and a different

account for each SQL Server service. To maintain the most granular control over permissions, use a different account and different domain group for each SQL Server service and for each virtual server in your domain.

- The SQL Server domain groups should not be shared with any other application.

Note that in order to troubleshoot domain group issues, you must have access to the domain controller.

SQL Server accounts will not be removed from the domain groups if SQL Server is uninstalled or if the accounts are changed. A domain administrator must ensure that all unwanted accounts are removed following removal of SQL Server.

▣ Change History

Release	History
12 December 2006	Changed content: <ul style="list-style-type: none">• Specified that domain groups for clustered services can be global or local domain groups..

▣ **See Also**

Concepts

[Before Installing Failover Clustering](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Cluster Node Configuration



[See Also](#)



[Setup User Interface Reference](#) >

Use the **Cluster Node Configuration** page of the Microsoft SQL Server Installation Wizard to select the cluster nodes to include in your virtual server.

▣ Options

Available Nodes

Displays the nodes that can be included in the virtual server definition. Click **Add** to move an available node to the **Selected Nodes** list.

Selected Nodes

Displays the nodes selected to be included in the virtual server definition.

Add

Move the selected node from the **Available Nodes** list to the **Selected Nodes** list, thereby adding the node to the virtual server definition.

Remove

Remove a selected node from the **Selected Nodes** list. This in turn removes the node from the virtual server.

Required Node

Displays the required node. The required node owns the cluster group where the SQL Server data files are installed.

Unavailable Nodes

Displays a list of unavailable nodes, and the reason each node is unavailable.

▣ **See Also**

Concepts

[Failover Clustering](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Remote Account Information



[_See Also](#)



[Setup User Interface Reference](#) >

Use the **Remote Account Information** page of the Microsoft SQL Server Installation Wizard to enter remote logon information for your virtual server.

▣ Options

Username

If the logon field is not pre-populated, specify a username that is a valid administrator account for all nodes of the cluster.

Password

Specify a password for the selected administrator account.

Security Note:

Setting strong passwords is essential to the security of your system.
Always use strong passwords.

▣ **See Also**

Concepts

[Failover Clustering](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Change or Remove Instance



[_See Also](#)



[Setup User Interface Reference](#) >

Use the **Change or Remove Instance** page of the Microsoft SQL Server Installation Wizard to choose whether to maintain or remove an existing instance of SQL Server. Maintenance options are provided through **Add or Remove Programs** in Control Panel.

▣ Options

Change Installed Features

Use this option to add features to or remove features from the selected instance of SQL Server.

Remove Microsoft SQL Server

Select this option to remove the SQL Server installation from your computer.

Note:

In this release, the maintenance of failover clusters is available only by using **Add or Remove Programs**.

▣ **See Also**

Tasks

[How to: Change Components in an Existing Instance of SQL Server 2005 \(Setup\)](#)

[How to: Uninstall an Existing Instance of SQL Server 2005 \(Setup\)](#)

Reference

[Editions and Components of SQL Server 2005](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Cluster Maintenance



[See Also](#)



[Setup User Interface Reference](#) >

Use the **Change or Remove** page of the Microsoft SQL Server Installation Wizard to choose whether to change or remove a virtual server installation.

Maintenance options are provide through **Add or Remove Programs** in Control Panel. Each component of SQL Server 2005 is listed individually in **Add or Remove Programs**.

▣ **Options**

Maintain the Virtual Server

Use this option to add or remove nodes.

Remove Microsoft SQL Server

Select this option to remove the virtual server from your cluster.

▣ **See Also**

Tasks

[How to: Create a New SQL Server 2005 Failover Cluster \(Setup\)](#)

[How to: Remove a SQL Server 2005 Failover Clustered Instance \(Setup\)](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Setup Progress



[See Also](#)



[Setup User Interface Reference](#) >

Use the **Setup Progress** page of the Microsoft SQL Server Installation Wizard to monitor the status of SQL Server Setup.

▣ Options

SQL Server 2005 Setup runs separately on each node of a failover cluster. To monitor Setup progress in individual nodes during a failover cluster installation, use the **Node** drop-down menu to toggle between cluster nodes.

Maintenance options are provide through **Add or Remove Programs** in Control Panel.

▣ **See Also**

Concepts

[How to: View SQL Server 2005 Setup Log Files](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Reserved Keywords (Setup)

 [See Also](#)



[Setup User Interface Reference](#) >

Microsoft SQL Server 2005 uses reserved keywords for defining, manipulating, and accessing databases. Reserved keywords are part of the grammar of Transact-SQL that is used by SQL Server to parse and understand Transact-SQL statements and batches.

SQL Server instance names cannot match a reserved keyword. Doing so will cause a setup error. Use the following table to ensure that reserved keywords are not used in SQL Server 2005 instance names.

ADD	EXCEPT	PERCENT
ALL	EXEC	PLAN
ALTER	EXECUTE	PRECISION
AND	EXISTS	PRIMARY
ANY	EXIT	PRINT
AS	FETCH	PROC
ASC	FILE	PROCEDURE
AUTHORIZATION	FILLFACTOR	PUBLIC

BACKUP	FOR	RAISERROR
BEGIN	FOREIGN	READ
BETWEEN	FREETEXT	READTEXT
BREAK	FREETEXTTABLE	RECONFIGURE
BROWSE	FROM	REFERENCES
BULK	FULL	REPLICATION
BY	FUNCTION	RESTORE
CASCADE	GOTO	RESTRICT
CASE	GRANT	RETURN
CHECK	GROUP	REVOKE
CHECKPOINT	HAVING	RIGHT
CLOSE	HOLDLOCK	ROLLBACK
CLUSTERED	IDENTITY	ROWCOUNT
COALESCE	IDENTITY_INSERT	ROWGUIDCOL

COLLATE	IDENTITYCOL	RULE
COLUMN	IF	SAVE
COMMIT	IN	SCHEMA
COMPUTE	INDEX	SELECT
CONSTRAINT	INNER	SESSION_USER
CONTAINS	INSERT	SET
CONTAINSTABLE	INTERSECT	SETUSER
CONTINUE	INTO	SHUTDOWN
CONVERT	IS	SOME
CREATE	JOIN	STATISTICS
CROSS	KEY	SYSTEM_USER
CURRENT	KILL	TABLE
CURRENT_DATE	LEFT	TEXTSIZE
CURRENT_TIME	LIKE	THEN

CURRENT_TIMESTAMP	LINENO	TO
CURRENT_USER	LOAD	TOP
CURSOR	NATIONAL	TRAN
DATABASE	NOCHECK	TRANSACTION
DBCC	NONCLUSTERED	TRIGGER
DEALLOCATE	NOT	TRUNCATE
DECLARE	NULL	TSEQUAL
DEFAULT	NULLIF	UNION
DELETE	OF	UNIQUE
DENY	OFF	UPDATE
DESC	OFFSETS	UPDATETEXT
DISK	ON	USE
DISTINCT	OPEN	USER
DISTRIBUTED	OPENDATASOURCE	VALUES

DOUBLE	OPENQUERY	VARYING
DROP	OPENROWSET	VIEW
DUMMY	OPENXML	WAITFOR
DUMP	OPTION	WHEN
ELSE	OR	WHERE
END	ORDER	WHILE
ERRLVL	OUTER	WITH
ESCAPE	OVER	WRITETEXT

Additionally, the SQL-2003 standard defines a list of reserved keywords. Avoid using SQL-2003 reserved keywords for object names and identifiers. The ODBC reserved keyword list, shown in the following table, is the same as the SQL-2003 reserved keyword list.

 **Note:**

The SQL-2003 reserved keywords list sometimes can be more restrictive than SQL Server and at other times less restrictive. For example, the SQL-2003 reserved keywords list contains **INT**. SQL Server does not treat **INT** as a reserved keyword.

▣ ODBC Reserved Keywords

The following words are reserved for use in ODBC function calls. These words do not constrain the minimum SQL grammar; however, to ensure compatibility with drivers that support the core SQL grammar, avoid using these keywords in SQL Server instance names.

ABSOLUTE	EXEC	OVERLAPS
ACTION	EXECUTE	PAD
ADA	EXISTS	PARTIAL
ADD	EXTERNAL	PASCAL
ALL	EXTRACT	POSITION
ALLOCATE	FALSE	PRECISION
ALTER	FETCH	PREPARE
AND	FIRST	PRESERVE
ANY	FLOAT	PRIMARY
ARE	FOR	PRIOR
AS	FOREIGN	PRIVILEGES
ASC	FORTRAN	PROCEDURE
ASSERTION	FOUND	PUBLIC
AT	FROM	READ
AUTHORIZATION	FULL	REAL
AVG	GET	REFERENCES
BEGIN	GLOBAL	RELATIVE
BETWEEN	GO	RESTRICT
BIT	GOTO	REVOKE
BIT_LENGTH	GRANT	RIGHT
BOTH	GROUP	ROLLBACK
BY	HAVING	ROWS
CASCADE	HOURL	SCHEMA
CASCADE	IDENTITY	SCROLL
CASE	IMMEDIATE	SECOND
CAST	IN	SECTION
CATALOG	INCLUDE	SELECT

CHAR	INDEX	SESSION
CHAR_LENGTH	INDICATOR	SESSION_USER
CHARACTER	INITIALLY	SET
CHARACTER_LENGTH	INNER	SIZE
CHECK	INPUT	SMALLINT
CLOSE	INSENSITIVE	SOME
COALESCE	INSERT	SPACE
COLLATE	INT	SQL
COLLATION	INTEGER	SQLCA
COLUMN	INTERSECT	SQLCODE
COMMIT	INTERVAL	SQLERROR
CONNECT	INTO	SQLSTATE
CONNECTION	IS	SQLWARNING
CONSTRAINT	ISOLATION	SUBSTRING
CONSTRAINTS	JOIN	SUM
CONTINUE	KEY	SYSTEM_USER
CONVERT	LANGUAGE	TABLE
CORRESPONDING	LAST	TEMPORARY
COUNT	LEADING	THEN
CREATE	LEFT	TIME
CROSS	LEVEL	TIMESTAMP
CURRENT	LIKE	TIMEZONE_HOUR
CURRENT_DATE	LOCAL	TIMEZONE_MINUTE
CURRENT_TIME	LOWER	TO
CURRENT_TIMESTAMP	MATCH	TRAILING
CURRENT_USER	MAX	TRANSACTION
CURSOR	MIN	TRANSLATE
DATE	MINUTE	TRANSLATION
DAY	MODULE	TRIM
DEALLOCATE	MONTH	TRUE
DEC	NAMES	UNION
DECIMAL	NATIONAL	UNIQUE
DECLARE	NATURAL	UNKNOWN
DEFAULT	NCHAR	UPDATE
DEFERRABLE	NEXT	UPPER
DEFERRED	NO	USAGE

DELETE	NONE	USER
DESC	NOT	USING
DESCRIBE	NULL	VALUE
DESCRIPTOR	NULLIF	VALUES
DIAGNOSTICS	NUMERIC	VARCHAR
DISCONNECT	OCTET_LENGTH	VARYING
DISTINCT	OF	VIEW
DOMAIN	ON	WHEN
DOUBLE	ONLY	WHENEVER
DROP	OPEN	WHERE
ELSE	OPTION	WITH
END	OR	WORK
END-EXEC	ORDER	WRITE
ESCAPE	OUTER	YEAR
EXCEPT	OUTPUT	ZONE
EXCEPTION		

▣ Future Keywords

The following keywords could be reserved in future releases of SQL Server as new features are implemented. Consider avoiding the use of these words in SQL Server instance names.

ABSOLUTE	FOUND	PREPARE
ACTION	FREE	PRESERVE
ADMIN	FULLTEXTTABLE	PRIOR
AFTER	GENERAL	PRIVILEGES
AGGREGATE	GET	READS
ALIAS	GLOBAL	REAL
ALLOCATE	GO	RECURSIVE
ARE	GROUPING	REF
ARRAY	HOST	REFERENCING
ASSERTION	HOUR	RELATIVE
AT	IGNORE	RESULT
BEFORE	IMMEDIATE	RETURNS
BINARY	INDICATOR	ROLE
BIT	INITIALIZE	ROLLUP
BLOB	INITIALLY	ROUTINE
BOOLEAN	INOUT	ROW
BOTH	INPUT	ROWS
BREADTH	INT	SAVEPOINT
CALL	INTEGER	SCROLL
CASCADED	INTERVAL	SCOPE
CAST	ISOLATION	SEARCH
CATALOG	ITERATE	SECOND
CHAR	LANGUAGE	SECTION
CHARACTER	LARGE	SEQUENCE
CLASS	LAST	SESSION
CLOB	LATERAL	SETS
COLLATION	LEADING	SIZE
COMPLETION	LESS	SMALLINT

CONNECT	LEVEL	SPACE
CONNECTION	LIMIT	SPECIFIC
CONSTRAINTS	LOCAL	SPECIFICTYPE
CONSTRUCTOR	LOCALTIME	SQL
CORRESPONDING	LOCALTIMESTAMP	SQLEXCEPTION
CUBE	LOCATOR	SQLSTATE
CURRENT_PATH	MAP	SQLWARNING
CURRENT_ROLE	MATCH	START
CYCLE	MINUTE	STATE
DATA	MODIFIES	STATEMENT
DATE	MODIFY	STATIC
DAY	MODULE	STRUCTURE
DEC	MONTH	TEMPORARY
DECIMAL	NAMES	TERMINATE
DEFERRABLE	NATURAL	THAN
DEFERRED	NCHAR	TIME
DEPTH	NCLOB	TIMESTAMP
DEREF	NEW	TIMEZONE_HOUR
DESCRIBE	NEXT	TIMEZONE_MINUTE
DESCRIPTOR	NO	TRAILING
DESTROY	NONE	TRANSLATION
DESTRUCTOR	NUMERIC	TREAT
DETERMINISTIC	OBJECT	TRUE
DICTIONARY	OLD	UNDER
DIAGNOSTICS	ONLY	UNKNOWN
DISCONNECT	OPERATION	UNNEST
DOMAIN	ORDINALITY	USAGE
DYNAMIC	OUT	USING
EACH	OUTPUT	VALUE
END-EXEC	PAD	VARCHAR
EQUALS	PARAMETER	VARIABLE
EVERY	PARAMETERS	WHENEVER
EXCEPTION	PARTIAL	WITHOUT
EXTERNAL	PATH	WORK
FALSE	POSTFIX	WRITE
FIRST	PREFIX	YEAR

FLOAT

PREORDER

ZONE

▣ **See Also**

Tasks

[How to: Install SQL Server 2005 \(Setup\)](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Component Selection - Maintenance



[See Also](#)



[Setup User Interface Reference](#) >

Use the **Component Selection - Maintenance** page of the Microsoft SQL Server Installation Wizard to select an instance of SQL Server or shared components to update.

▣ **Options**

Only a single selection is allowed on this page. You can select an instance or a common component to update.

Report

View the list of SQL Server 2005 components and features installed on your computer. The report includes version, edition, update level, and language information for each installed component and feature.

▣ **See Also**

Reference

[Components to Install](#)

[Features Supported by the Editions of SQL Server 2005](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Sample Databases Setup



[_See Also](#)



[Setup User Interface Reference](#) >

Use the **Sample Databases Setup** page of the Microsoft SQL Server Installation Wizard to select a method to install the sample databases to an existing instance of SQL Server.

Important When the **AdventureWorks** and **AdventureWorksDW** sample databases are installed, the collation used is SQL_Latin1_General_CP1_CI_AS and not the collation selected for the instance of SQL Server. To change the collation of a sample database to match the server collation, drop the database and recreate it from the installation script. For more information, see [Reinstalling Sample Databases from Script](#).

▣ Options

Install sample databases without attaching

The **AdventureWorks** database files are created in the folder <drive>:\Program Files\Microsoft SQL Server\90\Tools\Samples\AdventureWorks OLTP. **AdventureWorksDW** files are created in the <drive>:\Program Files\Microsoft SQL Server\90\Tools\Samples\AdventureWorks Data Warehouse folder. You must attach the database before you can use it. For more information, see **Attaching and Detaching a Database**.

Install and attach sample databases

The database files are created and stored in the folder <drive>:\Program Files\Microsoft SQL Server\MSSQL.n\MSSQL\Data. The database is attached to the selected instance of SQL Server and ready for use.

To attached the databases, you must have permission to create databases on the specified instance.

SQL Server instance

Select the instance of SQL Server on which to install the sample databases and samples.

To Complete the Samples Installation

To complete the installation of the samples, after Setup, perform one of the following steps:

1. From the **Start** menu, click **All Programs**, click **Microsoft SQL Server 2005**, click **Documentation and Tutorials**, click **Samples**, and then click **Microsoft SQL Server 2005 Samples**.
2. Alternatively, using Windows Explorer, navigate to <drive>:\Program Files\Microsoft SQL Server\90\Tools\Samples\ and double-click **SqlServerSamples.msi** to launch the installer.

 **Important:**

The samples are installed to <drive>:\Program Files\Microsoft SQL Server\90\Samples. Be sure to read and follow the installation instructions in the Readme files that accompany the samples you want to use. For a list of samples, see the 'Samples' topic in SQL Server 2005 Books Online.

▣ **To Deploy the Adventure Works DW Analysis Services Project**

1. Make sure that the **AdventureWorksDW** and **Adventure Works DW** databases have been installed.
2. Make sure Analysis Services has been installed.
3. From the **SQL Server Business Intelligence Development Studio** toolbar, click **File**, point to **Open**, and then click **Project/Solution**.
4. Browse to <drive>:\Program Files\Microsoft SQL Server\90\Tools\Samples\AdventureWorks Analysis Services Project, select the file **Adventure Works.sln**, and then click **Open**.
5. From **Solution Explorer**, right-click **Adventure Works DW** and select **Deploy** or **Process**.

▣ Change History

Release	History
5 December 2005	Changed content: <ul style="list-style-type: none"><li data-bbox="662 499 1247 541">• This is new content for this release.

▣ **See Also**

Reference

[Components to Install](#)

[Features Supported by the Editions of SQL Server 2005](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Component Selection - Uninstall



[_See Also](#)



[Setup User Interface Reference](#) >

Use the **Component Selection - Uninstall** page of the Microsoft SQL Server Installation Wizard to select a SQL Server instance and any shared components to remove.

▣ **Options**

You can select an instance and any common components to uninstall.

Report

View the list of SQL Server 2005 components and features installed on your computer. The report includes version, edition, update level, and language information for each installed component and feature.

▣ **See Also**

Reference

[Components to Install](#)

[Features Supported by the Editions of SQL Server 2005](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Confirmation - Uninstall



[Setup User Interface Reference](#) >

Use the **Confirmation - Uninstall** page of the Microsoft SQL Server Installation Wizard to confirm the features and components to remove from your SQL Server 2005 installation.

▣ **Options**

The dialog lists features and components you have selected to remove. Confirm that the list is complete, and then click **Next** to continue.

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Instance Type - Uninstall



[_See Also](#)



[Setup User Interface Reference](#) >

Use the **Instance Type- Uninstall** page of the Microsoft SQL Server Installation Wizard to select component to uninstall.

▣ **Options**

Select the component to remove, and then click **Next**.

▣ **See Also**

Reference

[Features Supported by the Editions of SQL Server 2005](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Orphan Component Selection



[_See Also](#)



[Setup User Interface Reference](#) >

Use the **Orphan Component Selection** page of the Microsoft SQL Server Installation Wizard to select a partially uninstalled component. You may also remove any remaining components from a previous installation after an upgrade.

▣ **Options**

Select the component to remove, and then click **Next**.

▣ **See Also**

Reference

[Features Supported by the Editions of SQL Server 2005](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Setup User Interface Reference (SQL Server Express)



[Setup User Interface Reference](#) >

This section helps guide you through the Microsoft SQL Server Installation Wizard so you can install SQL Server 2005 Express Edition.

The topics in the following table are available from the user interface by pressing the F1 key or by clicking **Help** in dialog boxes and on wizard pages.

F1 topics
System Configuration Check (SQL Server Express)
Registration Information (SQL Server Express)
Feature Selection (SQL Server Express)
Instance Name (SQL Server Express)
Existing Components (SQL Server Express)
Installation Options (SQL Server Express)
Service Account (SQL Server Express)
Authentication Mode (SQL Server Express)
Collation Settings (SQL Server Express)

[Configuration Options \(SQL Server Express\)](#)

[Error and Usage Report Settings \(SQL Server Express\)](#)

[Ready to Install \(SQL Server Express\)](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

System Configuration Check (SQL Server Express)



[Setup User Interface Reference](#) > [Setup User Interface Reference \(SQL Server Express\)](#) >

The System Configuration Check (SCC) page of the Microsoft SQL Server Installation Wizard shows progress of the system preparation tool. SCC routines scan the destination computer for conditions that can block Setup. For clusters, SCC runs on all nodes of the operating system cluster.

Click **Stop** at any time to cancel the SCC and exit the Installation Wizard.

Progress symbols are as follows:

- Arrow symbol = Check in progress.
- Check symbol = Check item successful.
- × symbol = Check item failed.

After the SCC check is complete, the **Report** button becomes available. **Next** is available only if all check results are successful, or if failed checks are non-fatal. For failed check items, resolution to blocking issues is included with results in the SCC report.

The **Report** button at the bottom of the SCC page lets you view the report, save it as a file, copy it to the Clipboard, or send it as e-mail. Information presented in the report includes:

- A list of configuration items that were checked on the destination computer: requirements for the current operating system, the operating system service pack level, SQL Server edition compatibility with the current operating system, and Internet Information Services (IIS).
- The result of each item check.
- The recommended remedy for failed configuration checks.

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Registration Information (SQL Server Express)



[Setup User Interface Reference](#) > [Setup User Interface Reference \(SQL Server Express\)](#) >

The Registration Information page of the Microsoft SQL Server Installation Wizard requests identification information to register your copy of SQL Server.

▣ Options

Option	Description
Name	Enter the name of a user responsible for administering the server that is running SQL Server 2005 Express Edition (SQL Server Express). A user name is required to continue setup.
Company	Enter the name of your company. For a personal copy of the product, you can provide your name again. Information for this field is optional.

 **Note:**

Because SQL Server Express is a free download, there is no Product Key text box on the Registration Information page.

▣ **Hide Advanced Configuration**

At the bottom of the Registration Information page, there is a **Hide Advanced Configuration Options** check box. If this check box is selected, the following setup pages do not appear:

- Instance Name
- Service Account
- Collation Settings
- User Instances

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Feature Selection (SQL Server Express)



[Setup User Interface Reference](#) > [Setup User Interface Reference \(SQL Server Express\)](#) >

The Feature Selection page consists of two panes. The left pane shows a tree view of the features that you can install. Expand a feature to view its subfeatures by clicking the + symbol. When you click a feature or subfeature, the Feature Description pane displays a description of the selection and its disk space requirements.

To add or remove a feature, click the feature's icon, and then choose one of the following options from the drop-down list:

Option	Description
Will be installed on local hard drive	Installs the selected feature in the location shown under Installation Path.
Entire feature will be installed on local hard drive	Installs the selected feature and all subfeatures. Expands the parent feature to view its subfeatures
Entire feature will be unavailable	For a new installation of SQL Server, this option skips installation of the selected feature. For an existing installation, this option removes the feature from the installation. After you select this option, a red X appears on the feature tree next to the feature name, and the icon on the parent feature in the tree view appears gray.  Note:

	<p>You cannot clear a feature if one of its subfeatures is selected. To clear a feature, make sure that all of its subfeatures are also cleared.</p>
Browse	<p>Launches the Change Folders page, which lets you change the folder where the selected feature will be installed. This button is available only for features with a user-configurable installation path. For more information on installation paths, see File Locations for Default and Named Instances of SQL Server 2005 in SQL Server 2005 Books Online.</p>
Disk Cost	<p>Displays the available space for installing to different system drives on your computer. After examining the disk space on your computer in the Disk Cost dialog box, click Close to continue installing SQL Server.</p>

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Instance Name (SQL Server Express)



[Setup User Interface Reference](#) > [Setup User Interface Reference \(SQL Server Express\)](#) >

Use the **Instance Name** page of the Microsoft SQL Server Installation Wizard to specify whether to create a default instance or a named instance of SQL Server Express. SQL Server Express always installs a named instance (SQLEXPRESS) unless you select a default instance. This behavior differs from SQL Server 2005, which installs a default instance unless a named instance is selected.

Options

Option	Description
Default instance	<p>If you select Default instance and a default instance already exists on the machine, SQL Server Setup will upgrade the default instance. Choose this option to install a default instance of SQL Server Express. A computer can host only one default instance.</p> <p> Note: The default instance name is MSSQLSERVER; it does not require a client to specify the name of the instance to make a connection.</p> <p>To upgrade the default instance of a previous SQL Server installation, click Default Instance and then click Next.</p>
Named instance	<p>When you run SQL Server Express, by default, the named instance option is selected with SQLExpress as the instance name.</p> <p>Choose this option to create a new named instance. Consider the following when you name an instance of SQL Server:</p> <ul style="list-style-type: none">• Instance names are not case sensitive.• Instance names cannot contain the terms Default, MSSQLServer, or other reserved keywords. If a reserved keyword is used in an instance name, an error will occur.• Instance names are limited to 16 characters.• The first character in the instance name must be a letter or an underscore (_). Acceptable letters are those defined by the Unicode Standard 2.0, including Latin characters a-z and A-Z, and letter characters from other languages.• Subsequent characters can be letters defined by the Unicode Standard 2.0, decimal numbers from Basic Latin or other national scripts, the dollar sign (\$), or an underscore (_).• Embedded spaces or other special characters are not allowed in instance names, nor are the backslash (\), comma (,), colon (:), semi-colon (;), single quotation mark ('), ampersand (&), number sign (#), or at sign (@).

	<p> Note: Only characters that are valid in the current Microsoft Windows code page can be used in SQL Server instance names. If an unsupported Unicode character is used, an error will occur.</p>
Installed Instances	Use the Installed Instances page of the SQL Server Installation Wizard to view a list of installed SQL Server Express instances and components on the computer where SQL Server Setup is running. The Installed Instance Details pane displays upgrade options for installed instances. To select a component or instance to upgrade, click the instance name and then click OK .

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Existing Components (SQL Server Express)



[Setup User Interface Reference](#) > [Setup User Interface Reference \(SQL Server Express\)](#) >

Use the Existing Components page of the Microsoft SQL Server Installation Wizard to view a list of SQL Server components that are already installed on the computer that is running SQL Server Setup.

▣ Options

Option	Description
Details	Click Details to see a report that lists the maintenance and upgrade options available for each currently installed SQL Server component. For more information about options that display after clicking Details , see Installation Options (SQL Server Express) .

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Installation Options (SQL Server Express)



[Setup User Interface Reference](#) > [Setup User Interface Reference \(SQL Server Express\)](#) >

Use the Installation Options page of the Microsoft SQL Server Installation Wizard to view maintenance and upgrade options for existing SQL Server Express instances and components. This page appears when components selected for the current installation are already part of an existing installation with the same instance name.

▣ Options

Maintenance and upgrade options are described in the following sections.

Maintenance

SQL Server Express instance maintenance and feature maintenance is provided through **Add or Remove Programs** in **Control Panel**. Select the instance of SQL Server to which you want to add or remove features. For more information, see [How to: Change Components in an Existing Instance of SQL Server 2005 \(Setup\)](#).

Upgrade

Supported upgrade paths are as follows:

Current SQL Server Instances	Upgrade to SQL Server Express (32-Bit)	Upgrade to SQL Server Express with Advanced Services (32-Bit)
SQL Server Desktop Engine (MSDE) 7.0 (SP4)	Yes	Yes
SQL Server 7.0 Desktop Edition SP4	No	No
SQL Server 7.0 Standard Edition SP4	No	No
SQL Server Developer Edition SP4	No	No
SQL Server Enterprise Edition SP4	No	No
SQL Server 7.0 Evaluation Edition SP4	Upgrades not supported	Upgrades not supported
SQL Server Desktop Engine (MSDE) 2000 SP3 or later	Yes	Yes
SQL Server 2000		

Personal Edition SP3 or later	Yes	Yes
SQL Server 2000 Workgroup Edition SP3 or later	No	No
SQL Server 2000 Standard Edition SP3 or later	No	No
SQL Server 2000 Developer Edition SP3 or later	No	No
SQL Server 2000 Enterprise Edition SP3 or later	No	No
SQL Server 2000 Evaluation Edition or later	No	No
SQL Server 2005 Express Edition	No	Yes
SQL Server 2005 Enterprise Edition	No	No
SQL Server 2005 Workgroup Edition	No	No
SQL Server 2005 Standard Edition	No	No
SQL Server 2005 Developer Edition	No	No
SQL Server 2005 Evaluation Edition	No	No

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Service Account (SQL Server Express)



[Setup User Interface Reference](#) > [Setup User Interface Reference \(SQL Server Express\)](#) >

Use the Service Account page of the Microsoft SQL Server Installation Wizard to assign a logon account to the SQL Server service.

You can assign the same login account to all SQL Server services, or you can configure each service account individually. You can also specify whether services start automatically.

Security Note:

Setting strong passwords is essential to the security of your system. Always use strong passwords.

Options

Option	Description
Service	SQL Server is selected as a service.
Use the built-in System account	You can assign Local System, Network Service, or Local Service to the logon for the configurable SQL Server services.
Local System account	<p>The Local System option specifies a local system account that does not require a password to connect to SQL Server on the same computer. However, the local system account might restrict the SQL Server installation from interacting with other servers, depending on the privileges granted to the account.</p> <p> Important: Local System is a powerful account. It might not be appropriate for all service settings. For more information, see "Security Considerations for a SQL Server Installation." in SQL Server 2005 Books Online.</p>
Network Service account	<p>The Network Service account is a special, built-in account that is similar to an authenticated user account. The Network Service account has the same level of access to resources and objects as members of the Users group. Services that run as the Network Service account access network resources using the credentials of the computer account.</p> <p> Important: We recommend that you do not use the Network Service account for the SQL Server. Local User or Domain User accounts are more appropriate for these SQL Server services.</p>
	The Local Service account is a special, built-in account that is similar to an authenticated user account. The Local Service account has the same level of access to resources and objects as members of the

Local Service account	Users group. This limited access helps safeguard the system if individual services or processes are compromised. Services that run as the Local Service account access network resources as a null session without credentials. For more information on service accounts, see Setting Up Windows Service Accounts in SQL Server 2005 Books Online.
Use a Domain User account	Specifies a domain user account that uses Windows Authentication to set up and connect to SQL Server. We recommend that you use a domain user account that has minimal rights for the SQL Server service, because the SQL Server service does not require administrator account privileges.
Username	Accepts or changes the domain user name.
Password	Setting strong passwords is essential to the security of the system. Always use strong passwords. For strong password guidelines, see Authentication Mode (SQL Server Express) .
Domain	Accepts or changes the domain name that is assigned to the account.  Note: The domain name cannot be a full DNS name. For example, if your DNS name is <i>my-domain-name.com</i> , use <i>my-domain-name</i> in the domain field.
Start services at the end of SQL Server Setup	Automatically starts the following services when the operating system is started. <ul style="list-style-type: none"> • SQL Server By default, the SQL Server option is selected. • SQL Server Browser SQL Server Browser is the name resolution service that provides SQL Server connection information to client computers. This service is shared across multiple instances of SQL Server.  Important: To enhance security in SQL Server 2005 Express Edition, some services and features are not enabled by default. They have to be configured and enabled after Setup is finished. For more information, see SQL Server Surface Area Configuration and

[Security Considerations for a SQL Server Installation.](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Authentication Mode (SQL Server Express)



[See Also](#)



[Setup User Interface Reference](#) > [Setup User Interface Reference \(SQL Server Express\)](#) >

Use the Authentication Mode page of the Microsoft SQL Server Installation Wizard to choose the form of authentication used to validate connections to this instance of SQL Server Express. If you select **Mixed Mode**, you must enter and confirm the SQL Server system administrator (**sa**) password. After a device establishes a successful connection to SQL Server, the security mechanism is the same for both Windows Authentication and Mixed Mode.

Options

Option	Description
Windows Authentication Mode	<p>When a user connects through a Microsoft Windows user account, SQL Server validates the account name and password using information in the Windows operating system. This is the default authentication mode, and is much more secure than Mixed Mode. Windows Authentication uses Kerberos security protocol, provides password policy enforcement in terms of complexity validation for strong passwords, provides support for account lockout, and supports password expiration.</p> <p> Security Note: When possible, use Windows Authentication.</p> <p>If you select Windows Authentication, Setup creates an sa account that is disabled by default. To use Mixed Mode Authentication and activate the sa account after Setup completes, see the How to: Change Server Authentication Mode and ALTER LOGIN (Transact-SQL) topics in SQL Server 2005 Books Online.</p> <p>Important Do not use a blank password. Use a strong password.</p>
Mixed Mode (Windows Authentication and SQL Server Authentication)	<p>Allows users to connect by using Windows Authentication and SQL Server Authentication. Users who connect through a Windows user account can use trusted connections that are validated by Windows.</p> <p>If you choose Mixed Mode Authentication you must set strong passwords for all SQL Server logins. This is of particular importance for sa, and for logins that are members of the sysadmin fixed server role.</p> <p>Note SQL Server Authentication is provided for backward compatibility only. When possible, use Windows</p>

	Authentication.
Specify the sa logon password below	<p>Enter and confirm the system administrator (sa) password when you choose Mixed Mode authentication. Setting strong passwords is essential to the security of your system. Never set a blank or weak sa password.</p> <p>Note SQL Server passwords can contain from 1 to 128 characters, including any combination of letters, symbols, and numbers. If you choose Mixed Mode authentication, you must enter a strong sa password before you can continue to the next page of the Installation Wizard</p>

▣ **Strong Password Guidelines**

Strong passwords are not readily guessed by a person, and are not easily hacked by using a computer program. Follow these guidelines when you set passwords:

1. Strong passwords cannot use prohibited conditions or terms, including:
 1. A blank or NULL condition
 2. "Password"
 3. "Admin"
 4. "Administrator"
 5. "sa"
 6. "sysadmin"
2. A strong password cannot use the following terms associated with the installation computer:
 1. The name of the user currently logged onto the computer.
 2. The computer name.
3. A strong password must be at least six characters in length and satisfy at least three of the following four criteria:
 1. It must contain uppercase letters.
 2. It must contain lowercase letters.
 3. It must contain numbers.
 4. It must contain non-alphanumeric characters – e.g., #, %, or ^.

Passwords entered on the Authentication page must meet strong password policy requirements. If you have any automation that uses SQL Server Authentication, make sure that the password meets strong password policy requirements.

▣ **See Also**

Other Resources

[SQL Server 2005 Books Online Download Center](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Collation Settings (SQL Server Express)



[See Also](#)



[Setup User Interface Reference](#) > [Setup User Interface Reference \(SQL Server Express\)](#) >

Use the Collation Settings page of the Microsoft SQL Server Installation Wizard to modify default collation settings used by SQL Server for language and sorting purposes. Choose the **Collation designator** option to match collation settings of different installations of SQL Server or of another computer. Use the **SQL Collations** option to match settings that are compatible with the sort orders in earlier versions of SQL Server.

□ Options

Option	Description
Collation designator and sort order	<p>Designates the collation to be used by this instance of SQL Server Express. An SQL collation is selected by default for English-language locales. The default collation for non-English locales is the Microsoft Windows locale setting for your computer — the Language for non-Unicode programs setting, or the closest equivalent from Regional and Language Options in Control Panel.</p> <p>The check boxes under Collation designator and sort order let you specify the sort order to use with the Collation designator you have selected. Binary is the fastest sorting order and is case-sensitive. If Binary is selected, the Case-sensitive, Accent-sensitive, Kana-sensitive, and Width-sensitive options are not available. For more information, see Windows Collation Sorting Styles in SQL Server 2005 Books Online.</p> <p>Note Change the default settings for the sort order only if your installation of SQL Server must match the collation settings used by another instance of SQL Server, or if it must match the Windows locale of another computer.</p>
SQL collations	<p>The SQL collations option is used for compatibility with earlier versions of SQL Server. Select this option to match settings compatible with SQL Server 8.0, 7.0, or earlier. For more information, see Using SQL Collations in SQL Server 2005 Books Online.</p>

▣ **See Also**

Other Resources

[SQL Server 2005 Books Online Download Center](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Configuration Options (SQL Server Express)



[See Also](#)



[Setup User Interface Reference](#) > [Setup User Interface Reference \(SQL Server Express\)](#) >

Use the Configuration Options page to generate a separate instance of the Database Engine for users without administrator permissions, and to add users to the administrator role.

▣ Option

Enable User Instances

Default is on. To disable the functionality of enabling user instances, clear the check box.

The user instance, also known as a child or client instance, is an instance of SQL Server that is generated by the parent instance (the primary instance that runs as a service, such as SQLExpress) on behalf of a user. The user instance runs as a user process under the security context of that user. The user instance is isolated from the parent instance and any other user instances running on the computer. The user instance feature is also referred to as “Run As Normal User” (RANU).

Add User to the SQL Server Administrator role

Default is not on. To add the current setup user to the SQL Server Administrator role, select the check box.

Windows Vista users that are members of BUILTIN\Administrators are not automatically added to the sysadmin fixed server role when they connect to SQL Server Express. Only Windows Vista users that have been explicitly added to a server-level administrator role can administer SQL Server Express. Any member of the Built-In\Users group can connect to the SQL Server Express instance, but they will have limited permissions to perform database tasks. For this reason, users whose SQL Server Express privileges are inherited from BUILTIN\Administrators and Built-In\Users in previous releases of Windows must be explicitly granted administrative privileges in instances of SQL Server Express running on Windows Vista.

To make any changes to the user roles after this installation program ends, use the SQL Server 2005 Surface Area Configuration Tool (SQLSAC.exe). To update the list of users in the SQL Server System Administrator role, click the **Add New Administrator** link.

Ensure that the **User to provision** field lists the DomainName\UserName of the user whose permissions should be updated. Select the role to be updated

from the list of SQL Server instances in the **Available privileges** pane, and then click the right arrow. To add the user to all available roles for all available instances of SQL Server instances and all available roles, click the double right arrow.

To implement the changes when your selections are complete, click **OK**. To end the tool without making changes, click **Cancel**.

▣ **Command Line Options**

The **ENABLERANU** switch is used to configure user instances.

To turn user instances off:

Set ENABLERANU = 0

To turn user instances on (default):

Set ENABLERANU = 1

▣ **Note:**

User instances apply only to SQL Server 2005 Express Edition editions.

The **ADDUSERASADMIN** switch is used to add the current setup user to the SQL Server Administrator role.

To add the current setup user to the Administrator role:

Set ADDUSERASADMIN=1

To not add the current setup user to the Administrator role:

Set ADDUSERASADMIN=0 (Default)

▣ **Note:**

This command line option only exists for SQL Server 2005 Express Edition editions.

▣ **See Also**

Other Resources

User Instances for Non-administrators

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Error and Usage Report Settings (SQL Server Express)



[Setup User Interface Reference](#) > [Setup User Interface Reference \(SQL Server Express\)](#) >

Use the Error Reporting page of the Microsoft SQL Server Installation Wizard to disable error reporting functionality for SQL Server 2005 Express Edition (SQL Server Express). Error reporting is enabled by default for SQL Server Express.

□ Option

Option	Description
Error reporting	<p>By default, the error reporting check box is selected. You can disable error reporting by clearing the check box.</p> <p>If you keep this feature enabled, SQL Server is configured to send a report to Microsoft automatically if a fatal error occurs in the SQL Server Database Engine. Microsoft uses error reports to improve SQL Server functionality and treats all information as confidential.</p> <p>Information about errors is sent over an encrypted channel to Microsoft, where it is stored with limited access. Alternatively, this information can be sent to your own Corporate Error Reporting server. See this Microsoft Web site for more information about how to set up a Corporate Error Reporting server. Error reports contain the following information:</p> <ul style="list-style-type: none">• The condition of SQL Server when the problem occurred.• The operating system version and computer hardware information.• Your Digital Product ID, which identifies your license.• The network IP address of the computer.• Information from memory or files of the process that caused the error. <p>Microsoft does not intentionally collect your files, name, address, e-mail address, or any other form of personal information. However, the error report can contain customer-specific information from the memory or files of the process that caused the error. Although this information can potentially be used to determine your identity, Microsoft does not use this information for that purpose. For the Microsoft error reporting data collection policy, see this Microsoft Web site.</p> <p>If you enable error reporting and a fatal error occurs, you might see a response from Microsoft in the Windows event log that points to a</p>

	<p>Microsoft Knowledge Base article for a particular error.</p> <p>To disable error reporting after Setup completes, click Start, click All Programs, click Microsoft SQL Server 2005, click Configuration Tools, and then click SQL Server Error and Usage Reporting.</p>
Feature Usage	<p>By default, the feature usage check box is selected. You can disable feature usage by clearing the check box. If you enable Feature Usage reporting, SQL Server will be configured to periodically send a report to Microsoft. Reports will include information about your hardware configuration and how you use SQL Server Express software and services. Microsoft will use feature usage data to improve SQL Server.</p> <p>Information is collected once per day for all enabled SQL Server instances. The time of collection defaults to midnight to minimize the load on the server. If you want to change the time of collection, you can manually edit the registry key that controls the collection time. Each SQL Server instance has its own registry key; the following example is for the first SQL Server instance installed on the server (MSSQL.1):</p> <p>HKLM\Software\Microsoft\Microsoft SQL Server\MSSQL.1\CPE\TimeofReporting</p> <p>The value of this registry key contains the time for the collection as the number of minutes from 00:00 (midnight) to run. For example, a value of 60 would run the collection at 1:00 a.m., a value of 1200 would run the collection at 8:00 p.m., and so on.</p> <p>To disable error reporting after Setup completes, click Start, click All Programs, click Microsoft SQL Server 2005, click Configuration Tools, and then click SQL Server Error and Usage Reporting.</p>

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Ready to Install (SQL Server Express)



[Setup User Interface Reference](#) > [Setup User Interface Reference \(SQL Server Express\)](#) >

The Ready to Install page appears when the SQL Server Installation Wizard has enough information to begin copying files to update your system.

▣ Options

Option	Description
Install	Select to install the components that are listed.
Back	Select to change any of the installation settings.
Cancel	Select to exit Setup.

[© 2006 Microsoft Corporation. All rights reserved.](#)

SQL Server Setup Help

Maintenance Feature Selection



[See Also](#)



[Setup User Interface Reference](#) > [Setup User Interface Reference \(SQL Server Express\)](#) >

Use the **Maintenance Feature Selection** page of the Microsoft SQL Server Installation Wizard to select a server component to maintain.

▣ **Options**

You can select any single server component to maintain.

Report

View the list of SQL Server 2005 components and features installed on your computer. The report includes version, edition, update level, and language information for each installed component and feature.

▣ **See Also**

Reference

[Components to Install](#)

[Features Supported by the Editions of SQL Server 2005](#)

[© 2006 Microsoft Corporation. All rights reserved.](#)